12-2014

Mathematical modeling for partial object detection.

Ahmed Reda Amin EL-Barkouky
University of Louisville

MATHEMATICAL MODELING FOR PARTIAL OBJECT DETECTION

By

Ahmed Reda Amin EL-Barkouky
B.S., Ain Shams University, 2002
M.S., Ain Shams University, 2009

A Dissertation
Submitted to the Faculty of the
J. B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering
University of Louisville
Louisville, KY, USA

December 2014

MATHEMATICAL MODELING FOR PARTIAL OBJECT DETECTION

By

Ahmed Reda Amin EL-Barkouky
B.S., Ain Shams University, 2002
M.S., Ain Shams University, 2009

A Dissertation Approved on

November 21, 2014

by the following Dissertation Committee:

Aly Farag, Ph.D.

Bruce Alphenaar, Ph.D.

Prasanna Sahoo, Ph.D.

Tamer Inanc, Ph.D.

Roman Yampolskiy, Ph.D.

DEDICATION

This dissertation is dedicated to my country

EGYPT

A great country in a temporary hardship.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Aly Farag, for his guidance and patience over the past five years. I would also like to thank my committee members, Dr. Bruce Alphenaar, Dr. Prasanna Sahoo, Dr. Tamer Inanc and Dr. Roman Yampolskiy, for their valuable comments and great support. I would also like to thank all the CVIP lab members for their help and support. I would also like to thank the graduate school of the University of Louisville for all the encouragement and support they provided for me through all my Ph.D. years and especially in the last year. I would also like to thank the Speed school of Engineering and the ECE department for their continuous support.

This work would have been impossible without the love, patience and support of my family. Many thanks are due to my beloved wife, Menna, for her patience, support and understanding during the Ph.D. hardworking years and my kids Malak and Omar for their lovely smiles that pushed me forward. I would also like to thank my parents, Dr. Reda EL-Barkouky and Dr. Ragaa Bahaa, for being my role models and for supporting my research journey before and through the Ph.D. degree. I would also like to thank my mother-in-law Eng. Howida Eissa for her support and continuous care.

ABSTRACT

MATHEMATICAL MODELING FOR PARTIAL OBJECT DETECTION

Ahmed R. EL-Barkouky

December 18, 2014

From a computer vision point of view, the image is a scene consisting of objects of interest and a background represented by everything else in the image. The relations and interactions among these objects are the key factors for scene understanding. In this dissertation, a mathematical model is designed for the detection of partially occluded faces captured in unconstrained real life conditions. The proposed model novelty comes from explicitly considering certain objects that are common to occlude faces and embedding them in the face model. This enables the detection of faces in difficult settings and provides more information to subsequent analysis in addition to the bounding box of the face.

In the proposed Selective Part Models (SPM), the face is modelled as a collection of parts that can be selected from the visible regular facial parts and some of the occluding objects which commonly interact with faces such as sunglasses, caps, hands, shoulders, and other faces. With the face detection being the first step in the face recognition pipeline, the proposed model does not only detect partially occluded faces efficiently but it also suggests the occluded parts to be excluded from the subsequent recognition step. The model was tested on several recent face detection databases and benchmarks and achieved state of the art performance. In addition, detailed analysis for the performance with respect

to different types of occlusion were provided. Moreover, a new database was collected for evaluating face detectors focusing on the partial occlusion problem.

This dissertation highlights the importance of explicitly handling the partial occlusion problem in face detection and shows its efficiency in enhancing both the face detection performance and the subsequent recognition performance of partially occluded faces. The broader impact of the proposed detector exceeds the common security applications by using it for human robot interaction. The humanoid robot Nao is used to help in teaching children with autism and the proposed detector is used to achieve natural interaction between the robot and the children by detecting their faces which can be used for recognition or more interestingly for adaptive interaction by analyzing their expressions.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

From the computer vision point of view, the image is a scene consisting of objects of interest and a background which is everything else in the image. Object detection can be defined as the automatic process of isolating these objects of interest from the background. The input for an object detector is a digital image or several frames from a video which may have multiple objects of interest, a single one or even no objects of interest at all. The output of the detector is the location and extent of each object of interest in the image if any. Object detection is crucial for any further processing or analysis starting from object tracking and recognition to scene understanding. Hence, failing to detect the object will eliminate the whole process.

The primary goal of this dissertation is focused on detecting partially occluded faces captured in unconstrained conditions. The face is modelled as a collection of parts that can be selected from the visible regular facial parts and some of the other objects that can possibly occlude faces such as sunglasses, caps, hands, shoulders and other faces. The proposed model can be seen from a scene understanding point of view in the sense that it is explicitly considering the relations and interactions between the face and its facial parts with other objects including facial accessories, other faces, and body parts that can occlude faces such as hands and shoulders which can be of great advantage to any further analysis of these faces.

1.1 Motivation

Faces in the wild have recently captured the focus of researchers for all facial analysis problems in different applications. Partial occlusion is a major problem for analyzing faces captured in unconstrained real life conditions. Even detecting the faces in such conditions is a challenging problem that needs to be solved before any further analysis of such faces can be done. The resulting problem is called Partial Face Detection where one or more of the main facial features of the face namely the two eyes, the nose and the mouth might be occluded. To illustrate the importance of the problem, Figure 1.1 shows two celebrities: Daniel Radcliffe (Harry Potter) and Jake Gyllenhaal (from the "Source Code" movie). Both Google Picasa and facebook auto-tagging face detectors failed to detect these faces. Although the faces in both images are partially occluded in a way that makes it very difficult even for state of the art commercial face detectors to detect, most people can still not just detect the faces but maybe easily recognize the celebrities in the images.

Figure 1.1: Examples of hard to detect partially occluded faces that can be detected by considering common face occlusions such as sunglasses, caps and hands.

The Partial occlusion is always considered as a problem that lowers the probability of any face candidate to be classified correctly as a face. The proposed argument here is that there are some types of occlusion that are common to hide parts of the face in real life images. For example, hands are common to hide the mouth if someone is smoking, eating or yawning and can similarly hide also other parts of the face with different other activities. Beside hands, there are also caps and sunglasses which are common for people to wear and they may result in hiding part of the face as well. This dissertation proposes that detecting these objects while detecting faces, can transform these types of partial occlusion from a problem that lowers the score of a face candidate to an advantage that actually raises it.

For the research of face recognition in the wild, most researchers either depend only on the output of typical face detectors as the starting point for face recognition which means that if the detector fails with such difficult faces then these face will not pass for recognition and hence the results are biased away from occlusion settings; or they crop these faces manually which results in systems that are not fully automated. Even if the face is detected, comparing the occluded part of the face with the un-occluded same part in the gallery will lead to problems in recognition. The proposed detector aims to supply the recognition module not only with a cropped face but also with information about the visible parts of this face which can be used in recognition. For example, if the eyes are hidden by sunglasses then the recognition module should not use signatures extracted from the eyes in recognition.

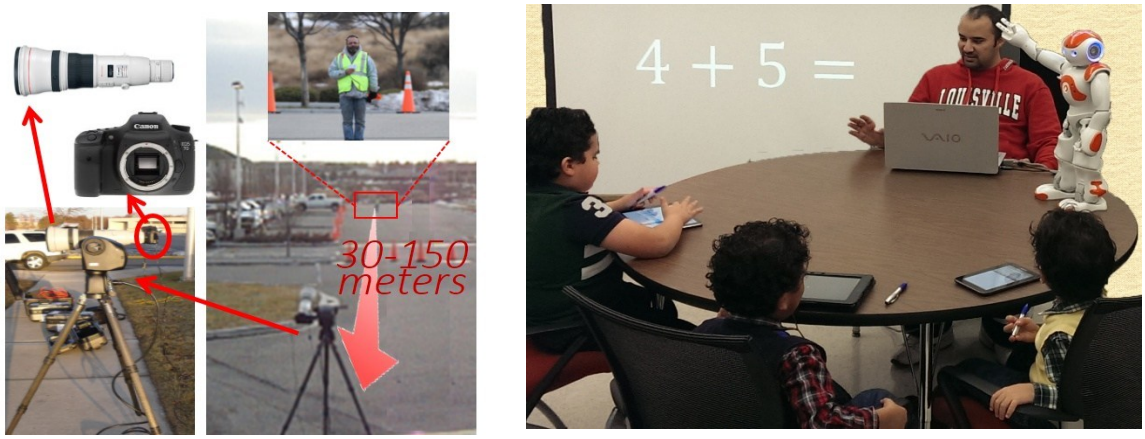
To complete the discussion, Figure 1.2 shows a categorization of different types of partial occlusion that can affect face detection. They can be grouped into two main types according to the reason of partial occlusion:

1. **Self-occlusion** which can result from:
    - Pose: where part of the face is occluded due to profile poses.
    - Facial accessories: part of the face is occluded by sunglasses, caps and scarfs.
    - Other objects that belong to the same subject such as his hands.
    - Facial hair: including moustaches and beards.
2. **External occlusion** which can result from:
    - Other objects between the face and the camera including other faces.
    - Limited field of view: part of the face is outside the camera's field of view.
    - Extreme illumination which includes sensor saturation, darkness, or shadows.

Figure 1.2: Categorization of partially occluded faces captured in the wild from FDDB and POF databases.

The proposed detector is designed with three applications in mind:

1. **Face Recognition at A Distance (FRAD) for security purposes:** In this application, the proposed detector is used as the front end of a system used in security for recognizing people at large distances. At this large distances the subjects are not aware of the camera shooting them and hence all types of occlusion can happen in such non-cooperative situations as shown in Figure 1.3.
2. **Human robot interaction:** In this application, a humanoid robot will be used to help in teaching children with autism. To interact properly with different children, it needs to recognize them interactively while they are doing activities. During the different teaching activities, parts of the face might get occluded and the robot cameras will be moving so faces will get out of field of view and then back in which complicates the problem. The proposed detector will be the first module in the pipeline for successful recognition of the children to allow natural interaction as shown in Figure 1.3.
3. **Auto-tagging:** A popular feature in social networking sites like Facebook and personal photo organizers like Picasa which enables users to add metadata about an image that include the names of the people in the image. To automatically do that, the first step is



Face Recognition at a distance **Human Robot Interaction**

Figure 1.3: Different applications for the proposed detector.

to detect the faces in an image. The proposed detector can enhance this automatic process in the cases of complicated images with occluded faces that current auto tagging systems may fail to detect as in Figure 1.1.

1.2 Problem statement

The problem of Partial Face Detection required for the aforementioned applications is illustrated in Fig 1.4 and can be defined as follows:

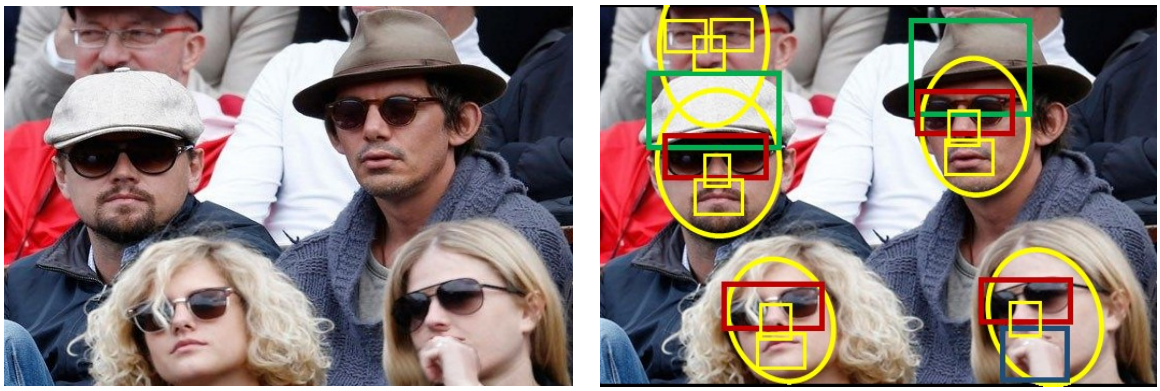
Input: - A still image, several frames from a video or live stream from the robot camera.

Output: - The location and extent of each face in the image.

- The location and extent of each facial part if visible (the two eyes the nose and the mouth).

- The location and extent of some occluding objects such as caps, hands and sunglasses.

Assumptions: - The input may have no faces at all, one face or multiple faces.



Input **Output**

Figure 1.4: Input/Output examples for partial face detection where in the output: faces and the visible facial parts are in yellow, sunglasses are in red, caps are in green and hands are in blue.

- Each face might not be completely visible in the input due to partial occlusion, but at least one of the four main facial features (the two eyes, the nose and the mouse) must be completely visible.

1.3 Evaluation methods

To evaluate the proposed face detector a database with ground truth annotations and a measure to compare the proposed method with other related work are needed. There are two types of error rates that define the performance of any object detector, false negative and false positive rates. The False Negative (FN) rate counts the number of objects in the image that was not detected and the False Positive (FP) rate counts the number of false detections in the image where the detector labeled a wrong region in the image as an object of interest. The number of correctly detected objects is called the True Positive (TP) rate. Usually False negative rates and true positive rates are normalized with respect to the total number of objects in all the images in the experiment and as a percentage they must add up to a 100% so false negative rate is implicitly included in the true positive rate. There is always a tradeoff between true positive and false positive rates; the score of each candidate is used with a threshold to change those two numbers resulting in what is called the Receiver Operating Characteristic (ROC) curve. The operating point of the detector can be selected from the ROC curve depending on the requirements of the system. Another curve that is also commonly used for evaluation is the precision recall curve, where precision is defined as Precision=TP/(TP+FP) and recall is defined as Recall=TP/(TP+FN) which is equivalent to TP rate definition. These terminologies are illustrated in Figure 1.5 and will be used to evaluate the efficiency of the detector.

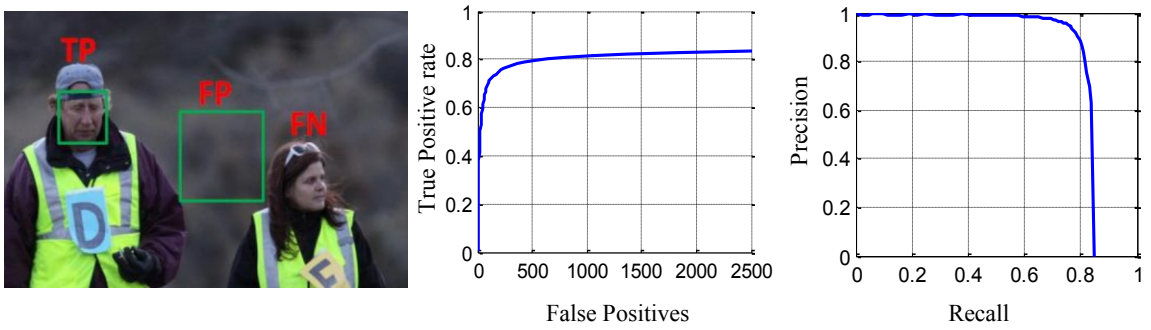


Figure 1.5: Evaluation methods: True Positive (TP), False Positive (FP), and False Negative (FN) definitions along with examples of ROC and Precision Recall curves.

1.4 Document organization

The rest of this document is organized as follows: Chapter 2 gives a brief description of modeling faces for the task of face detection from an image, where different feature extraction methods and machine learning algorithms will be visited to explain how they are used to model the face and distinguish it from the background. Chapter 3 discusses the problem of partial face detection and the proposed method for solving it. Chapter 4 shows how the proposed detector can be accelerated which is important for any input but of particular interest in videos. Chapter 5 completes the discussion with experimental results and applications for the proposed methods. Finally Chapter 6 summarizes the conclusion of the work and discusses possible future directions. Intuitively, Chapter 2 is the related work of the problem while Chapter 3 and 4 are the proposed solution, and Chapter 5 is its justifications and applications.

CHAPTER 2

MODELING FACES FOR DETECTION

Face detection is one of the most studied topics in computer vision because of its wide range of applications. It is also the first step for all facial analysis algorithms including face tracking, face alignment and face recognition, hence failing to detect the face will eliminate any following facial analysis step. The problem of face detection can be considered well solved for the constrained environments. Currently, any digital camera or cell phone can easily detect frontal faces and adjust its capture settings real time according to that. But despite of this maturity, the problem of face detection is still challenging in unconstrained environments which are currently tagged as “Faces in The Wild”. In such environments, factors like pose, illumination, expression and partial occlusion can combine to produce difficult settings that can make the state of the art face detectors fail to attain their task.

This chapter introduce the details of feature extraction and learning algorithms required to model faces for the detection task. First a general categorization of the existing face detectors is provided from different perspectives. Then two major general frameworks are introduced from a machine learning point of view explaining how each framework represents faces for detection through feature extraction and learning algorithm. Then, a brief discussion for partial occlusion handling in face detection is provided. Finally, a post processing generic method is proposed for reducing the false positives of any face detector using complimentary features like skin color.

2.1 Categorization of existing face detectors

Face detection has been extensively studied over the last two decades. A comprehensive survey of early approaches for face detection has been presented in Yang et al. [1] where single image face detection methods have been classified into four categories: **knowledge based methods** which use roles derived from human perception of a face; **feature invariant methods** which searches for properties of faces that are robust to pose and lighting variations; **template matching methods** which correlates the image with a standard face pattern; **appearance based methods** which learn face models from a set of training images. Due to the rapid advances in computational power and data storage, appearance based methods showed superior performance and have recently dominated the other categories in face detection as mentioned in the recent survey of Zhang et al. [2]. The general practice is to use a large set of positive and negative examples and then two key issues need to be chosen: the type of features extracted from the images and the learning algorithm used for training.

From another perspective, Gopalan et al. [3] classified the face detection methods into two categories: **sliding window** based methods and **local interest point** based methods. For sliding windows, a model is trained using a set of positive and negative windows corresponding to faces and non-faces; then this model is used across windows at all locations in different scales of a test image. On the other hand, instead of directly analyzing all regions of an image, local interest points with useful invariant properties can be detected and then descriptors only around these points are used for locating the faces. Although detecting local interest points might look better computationally and for handling object

deformations, but it does not guarantee repeatability of interest points. In particular for faces, most current detection methods are based on sliding windows.

Many face detection algorithms evolved as a special case of general object detection algorithms. From a machine learning point of view, Adaboost and Support Vector Machines (and their variants) are the most used learning algorithms for recent object detectors. The following two object detectors frameworks stand out as representatives for those two learning techniques because of the many recent detectors based on them for detecting faces in unconstrained environments:

- Boosting based approaches leaded by the seminal work of Viola and Jones (VJ) [4] which is a milestone in object detection in general and in face detection in particular that inspired many of the recent advances in face detection to use similar boosting cascade framework [5].
- Part based approaches have also received considerable attention in the last decade because of their flexibility for object detection. The Deformable Part Models (DPM) proposed by Falzenswalb et al. [6] can be considered the baseline framework for many of the recent face detectors using discriminatively trained maximum margin classifiers [7], [8], [9].

The following sections explain these two main frameworks, starting by a brief description of the learning framework in general then elaborate on the recent face detectors based on them and their corresponding features. The proposed face detector in this work belongs to the part based approaches but it will be evaluated relative to recent face detectors from both frameworks.

2.2 Boosting based approaches

In this section, a quick explanation of the AdaBoost algorithm is first provided, followed by a review of the Viola Jones framework which can be considered the baseline for this direction. Finally, some of the recent boosting algorithms are highlighted.

2.2.1 AdaBoost algorithm

Boosting is a method for combining (boosting) the performance of many weak classifiers to produce one highly accurate strong classifier (committee). Adaboost (Adaptive Boosting) trains weak classifiers successively on weighted versions of the training data giving higher weights to cases that are currently misclassified then combines these weak classifiers to produce a powerful strong classifier.

As explained in Freund et al. [10], the input to the Adaboost algorithm is a training set $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in \mathcal{X}$ the space of feature vectors extracted from an image window, $y_i \in \{-1, +1\}$ the binary label set.

Adaboost calls a given weak learning algorithm repeatedly in a series of rounds $t = 1, \dots, T$ over a weighted version of the training set. The weight of the training element i on round t is denoted as $D_t(i)$. Initially, all weights are set equally as $D_1(i) = 1/m$. Then for each iteration t, the weak learner is trained over the weighted training set to produce a weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$ appropriate for the distribution (weights) D_t. The goodness of a weak hypothesis is measured by its error $\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$ which sums all the weights of misclassified elements of the training set. Using this error value, the algorithm choses for this hypothesis h_t a coefficient $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ which is inversely

proportional with ϵ_t, where $\alpha_t \geq 0$ if $\epsilon_t \leq \frac{1}{2}$ (which should always be the case for a weak hypothesis to be better than random guessing). Finally, the weights are updated using $D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ such that the weights of correctly classified elements are decreased while the weights of incorrectly classified elements are increased in a way that forces the next weak hypothesis to focus on the misclassified elements. A normalization factor Z_t is chosen such that D_{t+1} will remain a distribution that sums up to 1.

After T rounds, the final hypothesis $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$ is a majority vote of the T weak hypothesis with coefficients α_t. The algorithm is shown in Figure 2.1.

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.

Initialize $D_1(i) = 1/m$.

For $t = 1, \ldots, T$:

1. Train weak learner using distribution D_t.
2. Get a weak hypothesis $h_t : \mathcal{X} \rightarrow \{-1, +1\}$ with error $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$
3. Choose $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
4. updated weights using $D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{\sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i))}$

The final hypothesis $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Figure 2.1: The Adaboost algorithm of Freund et al. [10]

2.2.2 The Viola Jones framework

The work of Viola and Jones [11] can be considered one of the first robust real time face detectors that is still being used in practice. Three main ideas are behind the success of this detector: the integral image, the Adaboost and the attentional cascade structure.

The integral image is an algorithm for a quick and efficient calculation for the sum of intensity values in a rectangular subset of an image. The integral image is defined as:

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

where $i(x,y)$ is the intensity of the gray scale image at pixel (x,y). Using the integral image as illustrated in Figure2.2, the sum of the intensity pixels of any rectangular area ABCD can be calculated with only four array references as:

$$\sum_{(x,y)\in ABCD} i(x,y) = ii(D) + ii(A) - ii(B) - ii(C) \tag{2.2}$$

Viola and Jones used this concept for rapid computation of a huge number of Haar like features which are simply defined as the difference between the sum of the intensities in the dark and light shaded regions of simple rectangular patterns as shown in Fig 2.2 (e).

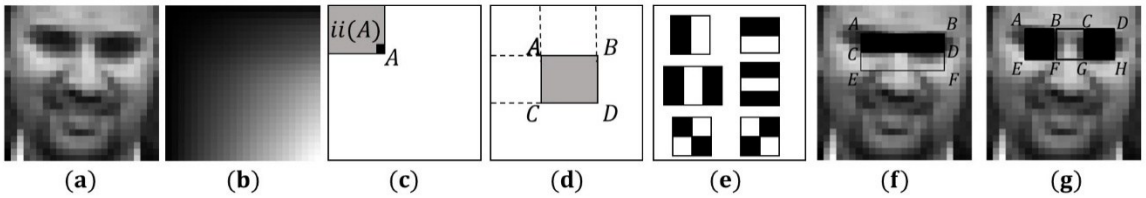


Figure 2.2: Efficient calculation of Haar like features using integral image: (a) Original image (b) Integral image (c) illustration of $ii(A)$ (d) Illustration of equation 2.2 (e) Examples of Haar like features (f) Example of a Haar feature overlapped with original image and illustration of equation 2.3 (g) Example of another Haar feature overlapped with original image.

For example, the feature shown in Fig 2.2 (f) can be calculated as:

$$f = [ii(D) + ii(A) - ii(B) - ii(C)] - [ii(F) + ii(C) - ii(D) - ii(E)] \quad (2.3)$$

This computational advantage enabled scaling the features for multi-scale detection at no additional cost because it requires the same number of operations despite of size. In contrast to the conventional image pyramid used in most face detectors to detect over multi-scales by scanning a fixed scale detector over different scales of the image, Viola and Jones scaled the detector itself and saved the time of building the image pyramid.

The Adaboost is used both to select features and to train the classifier. The weak learner is designed here to select the feature which best separates the weighted positive and negative training examples. A weak classifier $h(x, f, p, \theta)$ is defined as:

$$h(x, f, p, \theta) = \begin{cases} 1 & if\ pf(x) < p\theta \\ 0 & otherwise \end{cases} \quad (2.4)$$

Where f is a feature from the huge set spanning different sizes of the Haar like features shown in Figure 2.2 (e), p is a polarity indicating the direction of the inequality, θ is a threshold and x is a training sub window of size 24x24 pixel. Note that in training, all the training examples are 24x24 pixels. This weak classifiers that threshold single features can be viewed as single node decision trees which are usually called decision stups in the machine learning literature.

The Adaboost algorithm of Viola and Jones is described in Fig 2.3 which has been modified to match the notation used in Fig 2.1. The convention used by Viola and Jones was to use 0 for labels of negative examples which is changed to -1 to match the terminology used here. One difference from the Adaboost algorithm explained earlier is

Given: $(x_1, y_1), \ldots, (x_m, y_m), x_i \in \mathcal{X}, y_i \in \{-1,1\}$

Initialize $D_1(i) = \frac{1}{2n}, \frac{1}{2l}$ for $y_i = -1,1$ respectively.

For $t = 1, \ldots, T$:

1. Normalize the weights $D_t(i) = \frac{D_t(i)}{\sum_{i=1}^{m} D_t(i)}$
2. Select the best weak classifier w.r.t weighted error:
$$\epsilon_t = \min_{f,p,\theta} \sum_{i:h(x_i,f,p,\theta)\neq y_i} D_t(i)$$
3. Define $h_t(x) = h(x, f_t, p_t, \theta_t)$; $f_t, p_t, \theta_t = \operatorname{argmin}_{f,p,\theta} \sum_{i:h(x_i,f,p,\theta)\neq y_i} D_t(i)$
4. Choose $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
5. Updated weights using $D_{t+1}(i) = D_t(i)\left(\frac{\epsilon_t}{1-\epsilon_t}\right)^{1-e_i}$
where $e_i = 0$ if example x_i is classified correct and $e_i = 1$ if misclassified.

The final hypothesis $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Figure 2.3: The Adaboost algorithm used in Viola Jones detector [4]

that the weights of correctly classified examples are not reduced while the weights of misclassified examples are still increased. Also the weights of the positive and negative examples are initialized according to their respective numbers.

The attentional cascade of classifiers is used to combine increasingly more complex classifiers successively which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. Simpler and therefore faster boosted classifiers (with low T) are used first to reject the majority of negative windows while passing almost all positive windows. Then more complex and therefore slower boosted classifiers (with high T) are used to reject the much fewer number of difficult negative windows.

2.2.3 Boosting based face detectors

During the last decade, several face detectors have been designed based on the Viola Jones framework. The differences are mainly in the version of the Adaboost used and the feature extraction method ranging from variants of the Haar like features to LBP, LAB and SURF. The following are some of the face detectors designed in this direction.

Lienhart et al. [12] generalized the Haar like features by introducing 45 degree rotation and center surround features that can still be computed efficiently using integral image but provided more flexibility leading to better results. The OpenCV implementation of the VJ detector adopted this modification as an option for the Haar like features they offer. The OpenCV implementation also had an option for using Local Binary Pattern (LBP) [13]. LBP has been very successful in face recognition tasks due to its robustness to illumination variations which encouraged Zhang et al. to adopt it under the boosting framework for face detection using a multi branch regression tree as its weak classifier and Gentle Adaboost [14].

Recently, Li et al. [15], [5] proposed a face detector derived from the VJ boosting cascade framework but using the multi-dimensional SURF features instead of single dimensional Haar features to describe local patches which enabled them to reduce the number of used local patches from hundreds of thousands to several hundreds. They also adapted logistic regression as a weak classifier instead of decision trees and used it in building a face detector that can be trained very efficiently from billions of negative samples.

Zhang et al. [16] visited the problem of detector adaptation in which a generic classifier trained from extensive data is adapted to data coming from a different test environment to improve the performance on it. They presented a general formulation of the adaptation problem and demonstrated it on the boosting framework. Jain et al. [17] also presented an online approach for rapidly adapting a pre-trained cascade of classifiers to a new testing data set without retraining the classifier. They used the VJ framework as a base face detector on which they applied their domain adaptation to each of the classifiers in the cascade.

2.3 Part based approaches

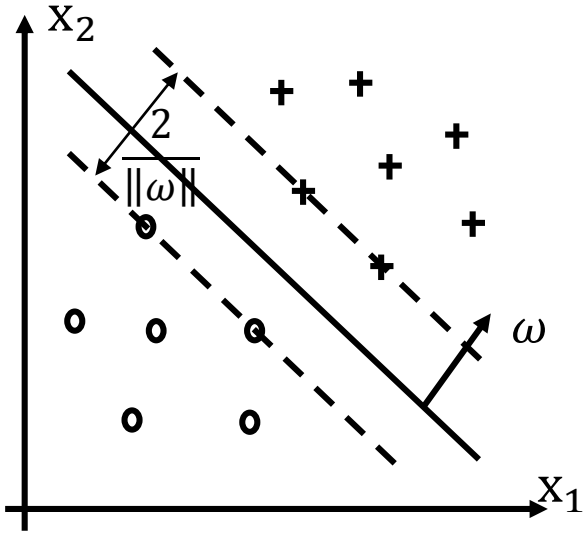
Part based approaches models the face as a root filter that captures the global appearance of the face and several part filters that capture more detailed texture of the different facial parts. These models have received considerable attention in the last decade for object detection in general, while in the last couple of years it received special attention in solving the difficult challenges of face detection in the wild. This section starts with a brief overview of the maximum margin classifiers also known as the Support Vector Machines (SVM). Followed by an explanation of the Deformable Part Models (DPM) proposed in Felzenszwalb et al. [6] which can be considered as the baseline framework for many of the recent face detectors using discriminatively trained maximum margin classifiers [7], [8], [9]. Finally, a brief overview of these face detectors will be provided.

2.3.1 Support Vector Machines

A Support Vector Machine is a discriminative classifier that constructs a hyperplane to be used for separating examples from two different classes with maximum margin to the

closest training examples from each class which are called the support vectors. The discussion here starts with linear SVM using a training data that is linearly separable to illustrate the problem then adds soft margins to allow for outliers.

A training set $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in R^n$, $y_i \in \{-1, +1\}$ is linearly separable if exists $\omega \in R^n, b \in R$, $\varepsilon > 0$ such that $(\omega . x_i) + b \geq \varepsilon$ for any training example with $y_i = 1$ and $(\omega . x_i) + b \leq -\varepsilon$ for any training example with $y_i = -1$. Figure 2.4 shows a two dimensional example with $x_i = (\mathrm{x}_{1_i}, \mathrm{x}_{2_i}) \in R^2$, and training examples with $y_i = 1$ represented by "+" while training example with $y_i = -1$ represented by "o". Consider the separating line $(\bar{\omega} . x) + \bar{b} = 0$; which is the middle line between two support lines $(\bar{\omega} . x) + \bar{b} = k, (\bar{\omega} . x) + \bar{b} = -k$ as shown in Figure 2.4 with the solid and dashed lines respectively. Using $\omega = \frac{\bar{\omega}}{k}, b = \frac{\bar{b}}{k}$, the separating line becomes $(\omega . x) + b = 0$ and the two support lines become $(\omega . x) + b = 1, (\omega . x) + b = -1$. To find the normal distance between the two support lines, any point $(\mathrm{x}_1, \mathrm{x}_2)$ on the first line is first found by using its equation $\omega_1 \mathrm{x}_1 + \omega_2 \mathrm{x}_2 + b = 1$ and for example setting $\mathrm{x}_2 = 0$ which leads to



Given $(x_1, y_1), \dots, (x_m, y_m)$
$x_i \in R^n$, $y_i \in \{-1, +1\}$

find $\omega^\star, b^\star$ that :

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2$$

s.t. $y_i\big((\omega . x_i) + b\big) \geq 1, \ i = 1, \dots, m.$

The separating hyper plane
$(\omega^\star . x) + b^\star = 0$

The decision function is
$f(x) = sign((\omega^\star . x) + b^\star)$

Figure 2.4: Linearly separable SVM problem

$x_1 = \frac{1-b}{\omega_1}$. The normal distance between the point $(\frac{1-b}{\omega_1}, 0)$ and the second line $\omega_1 x_1 + \omega_2 x_2 + b = -1$ is $\frac{\omega_1 \frac{1-b}{\omega_1} + \omega_2(0) + b + 1}{\sqrt{\omega_1^2 + \omega_2^2}} = \frac{2}{\|\omega\|}$. The idea of maximal margin is to maximize this margin (or minimize it reciprocal) while preserving the separation of all the samples in the two classes. The problem is illustrated in Figure 2.4 where $\|\omega\|$ is squared for mathematical convenience.

To allow for outliers that might result from labelling errors or from the data not being perfectly linearly separable, a slack variable ξ is introduced to measure the degree to which each constraint is violated and a cost is associated in the minimization for this violation. This results in the soft margin problem described in Figure 2.5.

Given a training set: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in R^n,\ y_i \in \{-1, +1\}$

Find $\omega^\star, b^\star$ that:

$$\min_{\omega, b, \xi} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m} \xi_i$$

s.t. $\quad y_i\big((\omega . x_i) + b\big) \geq 1 - \xi_i, \quad i = 1, \dots, m$

$$\xi_i \geq 0, \quad i = 1, \dots, m$$

Where $\omega \in R^n, b \in R,\ \xi = (\xi_1, \dots, \xi_m), C > 0$ (penalty parameter)

The separating hyper plane is $(\omega^\star . x) + b^\star = 0$

The decision function is $f(x) = sign((\omega^\star . x) + b^\star)$

Figure 2.5: The Support Vector Machine linear classification problem.

The aforementioned problem is called the primal problem of linear SVM and to solve it, the general practice is to construct a dual problem that is convex quadratic programming using Lagrange multipliers as explained in Deng et al. [18]. Several libraries are available

for solving the SVM such as LibSVM [19] and SVMlight [20]. Nonlinear SVM is achieved through the use of kernels to map the nonlinearly separable training data to another space that is linearly separable [18]. The problem in Figure 2.5 remains the same with just replacing x_i with the map $\varphi(x_i)$. This minimization problem can also be formulated as $\min_{\omega,b,\xi} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m} \max\big(0,1 - y_i(\omega.\varphi(x_i))\big)$ where the hinge loss $\sum_{i=1}^{m} \max\big(0,1 - y_i(\omega.\varphi(x_i))\big)$ is minimized when the scoring function fits the examples well with a safety margin, while $\frac{1}{2}\|\omega\|^2$ is interpreted as a regularization term that ensures a small variation of $\varphi(x_i)$ does not change the score $\omega.\varphi(x_i)$ too much.

2.3.2 Deformable Part Models

Deformable part models provide an elegant framework for modeling different object categories. The Deformable Part Models proposed in Felzenszwalb et al. [6] will be briefly described here as it can be considered the baseline framework for many of the recent face detectors. This star model is defined by a root filter that can capture coarse details such as the face boundary, combined with part filters that can capture smaller and more detailed parts of the face such as the eyes, nose and mouth at twice the spatial resolution.

They defined the model for an object with n parts as $(F_0, P_1, \dots, P_n, b)$ where F_0 is the root filter, P_i is the model for the i^{th} part and b is a bias term. Each part model is defined as (F_i, v_i, d_i) where F_i is the i^{th} part filter, v_i is the anchor position representing the default part position with respect to the root filter, and d_i defines the deformation cost of the part relative to anchor position. An object candidate is determined by the location $z = (p_0, p_1, \dots, p_n)$ of each filter in the model in a feature pyramid H where $p_i = (x_i, y_i, l_i)$ with l_0 being the level of the root filter in H while the rest of l_i being the level of the part

filters in H this is referred to as $l_i = l_0 - \lambda$ where λ is the number of levels separating two levels in the pyramid at twice the resolution. The score of this candidate is given by:

$$score(p_0, \dots, p_n) = F_0 \,.\, \varphi_a(H, p_0) + \sum_{i=1}^{n} [F_i \,.\, \varphi_a(H, p_i) - d_i . \varphi_d(dx_i, dy_i)] + b \quad (2.5)$$

where φ_a is the appearance feature vector in a sub-window of H defined by p_i, $\varphi_d(dx_i, dy_i)$ is the deformation feature defined as $(dx_i, dy_i, dx_i^2, dy_i^2)$ where $(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i)$. This score can be expressed as a dot product between a vector of model parameters $\omega = (F_0, F_1, \dots, F_n, d_1, \dots, d_n, b)$ and a feature vector $\psi(H, z) = (\varphi_a(H, p_0), \varphi_a(H, p_1), \dots, \varphi_a(H, p_n), -\varphi_d(H, p_1), \dots, -\varphi_d(H, p_n), 1)$:

$$score(z) = \omega . \psi(H, z) \quad (2.6)$$

For training, they used latent SVM which considers a classifier that scores each example x with a function $f_\omega(x) = \max_{z \epsilon Z(x)} \omega \,.\, \psi(H, z)$ where the set $Z(x)$ defines all possible latent values for an example x. A binary label for this example x can be obtained by thresholding the score. The model parameters ω can be trained from labeled examples $(x_1, y_1), \dots, (x_m, y_m)$ by minimizing the objective function:

$$L(\omega) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{m} \max\big(0, 1 - y_i f_\omega(x_i)\big) \quad (2.7)$$

The training data contains only bounding box of the object and hence the unknown part filter locations are considered latent variables. Note that if there is a single possible latent value for each example then f_ω is linear in ω and the latent SVM is simplified to the linear SVM explained in the previous subsection.

For detection, the response of the i^{th} filter F_i in the l^{th} level of H is calculated by $R_{i,l}(x,y) = F_i \,.\, \varphi_a(H,(x,y,l))$ in a sub-window of H with top left corner at (x,y) and its size is defined by the size of the root filter. To find the optimum locations for the parts with respect to the root location they used the generalized distance transform proposed by Felzenszwalb et al. [21]. The updated response of the i^{th} part filter in this optimum location $D_{i,l}$ and the corresponding optimal displacement $P_{i,l}$ are given respectively by:

$$D_{i,l}(x,y) = \max_{dx,dy} \left[R_{i,l}(x+dx, y+dy) - d_i \,.\, \varphi_d(dx,dy)\right] \tag{2.8}$$

$$P_{i,l}(x,y) = \operatorname*{argmax}_{dx,dy} \left[R_{i,l}(x+dx, y+dy) - d_i \,.\, \varphi_d(dx,dy)\right] \tag{2.9}$$

This transformation spreads high part scores to nearby locations according to a deformation cost. The overall score of this window is then calculated as:

$$score(x,y,l) = R_{0,l}(x,y) \; + \; \sum_{i=1}^{n} D_{i,l-\lambda}(2(x,y) + v_i) \; + \; b \tag{2.10}$$

where part filters are taken at twice the resolution of root filter $(l-\lambda)$ using the shifted and subsampled transformed part filter responses $D_{i,l-\lambda}(2(x,y)+v_i)$. The bias term is introduced to make the scores of multiple models comparable if combined in a mixture model for example to have separate models for frontal and profile views. After finding a window with high score that will be considered a detection, the corresponding part locations can be found as $P_{i,l-\lambda}(2(x,y)+v_i)$.

2.3.3 Part based face detectors

Many of the state of the art face detectors in the last couple of years have turned to the use of part based models for the task of detecting faces in unconstrained conditions. Zhu et

al. [7] presented a unified model for face detection, pose estimation and landmark localization based on mixtures of trees with a shared pool of parts by modeling every facial landmark as a part and allowing different view mixtures to share part templates. Their tree structured model contains 13 different viewpoints and 5 expressions limited to frontal view point. They used up to 68 part filter per pose corresponding to the facial landmarks with a total of 99 parts across all viewpoints. Each part is represented as a 5x5 HOG cells of size 4. They explored 4 levels of sharing parts between mixtures starting from share-99 which is fully shared that shares a single template for each of the 99 parts across all mixtures, then share-146 which shares templates only across similar topology viewpoints, then share-622 that shares only across neighboring viewpoints and finally share-1050 which does not share any part templates.

However in [8], Orozco et al. argued that these models presented in [7] were aimed to landmark localization and pose estimation but they are suboptimal when only face detection is required because it requires full landmark labelling which reduces the amount of data that can be used in training and it is slow for practical face detection applications. They presented an empirical analysis comparing the approach in [7] and a cascade DPM [6] for the task of face detection. They treated part locations as latent variables during training and used latent SVM to train a 4 pose model. They also used a cascade DPM which speeded up the performance without sacrificing the detection accuracy.

Yan et al. [9] proposed using a hierarchical part based structure face model based on DPM but with allowing part subtypes to describe appearance variations in parts for example differentiating between closed eye and open eye. Similar to [7], they defined their parts around facial landmarks. For part subtypes they used K-means to cluster each part to

a fixed number of subtypes. They also used body context to enhance their results when faces are difficult to detect. In [22], Yan et al. also proposed a real time face detector based again on DPM by using pre-calculated lookup tables to efficiently calculate the Histogram of Oriented Gradient (HOG). Finally, Mathias et al. in the latest work of face detection [23] agrees with the previous categorization of recent best face detection methods as the children of two classic detection approaches Viola Jones with Adaboost and DPM with HOG and SVM. They trained two detectors based on each of these methods and achieved comparable performance concluding that the DPM is the method of choice if only few training samples are available and at the same time high recall is of essence.

2.4 Occlusion handling

The problem of partial occlusion on face detection is not usually addressed explicitly in the literature, but instead it is considered implicitly. For example, in part based models the partial occlusion affects the occluded parts score but the face can still be weakly detected by the score of other parts depending on how severe is the occlusion. Goldmann et al. [24] used a component detection stage to detect facial parts individually then a topology verification stage that uses the spatial configuration of these parts to find possible faces even in the absence of some parts but with weak scores. To help in such situations, Yan et al. [9] used body context to raise the score of a partial face using its accompanied upper body. Another direction used by Lin et al. [25] was to train a separate detector similar to the VJ detector for faces with different parts occluded, they artificially simulated eight different types of occlusions by removing either one third of the face vertically or horizontally or by removing one fourth of the face diagonally. The disadvantage in artificially removing part of the face is it cannot model the real occlusions happening in

unconstrained environments rather it only models the remaining part of the face. The problem becomes difficult when more than one part of the face is occluded and most face detectors fail to detect these faces like the examples in Figure 1.1. In my work, Selective Part Models (SPM) is proposed which can be used to select between regular parts of the face (e.g. eyes, nose and mouth) and common types of occlusion (e.g. hands, sunglasses and caps). This enables explicitly modeling some occlusions and hence detecting faces in very difficult settings that cannot be detected even by state of the art commercial face detectors. The model also use overlap maps to raise the scores of faces occluded by other faces and shoulders. Although many of the recent work based on part models [7], [9] used large number of parts defined around the common face recognition landmarks; I used only four facial parts in SPM to allow more in-class variability. I believe that this large number of parts is more suitable when classification between different faces is needed as in face recognition but not when different faces need to be considered as one class as in face detection. In addition, the execution time is also affected by the large number of parts.

2.5 Reducing the False Positives

There are two types of errors that define the performance of any face detector: False negatives which are the faces in the image that were not detected by the detector and false positives which are non-faces regions in the images that were mistakenly detected as faces by the detector. The false negatives usually result from challenges like pose, illumination, expression and partial occlusion that could not be captured by the face model, while false positives usually results from the complex environment around the faces which may deceive the face model and be detected as faces because of their resemblance to the face shape and texture or because of the difficulty of selecting negative training examples to

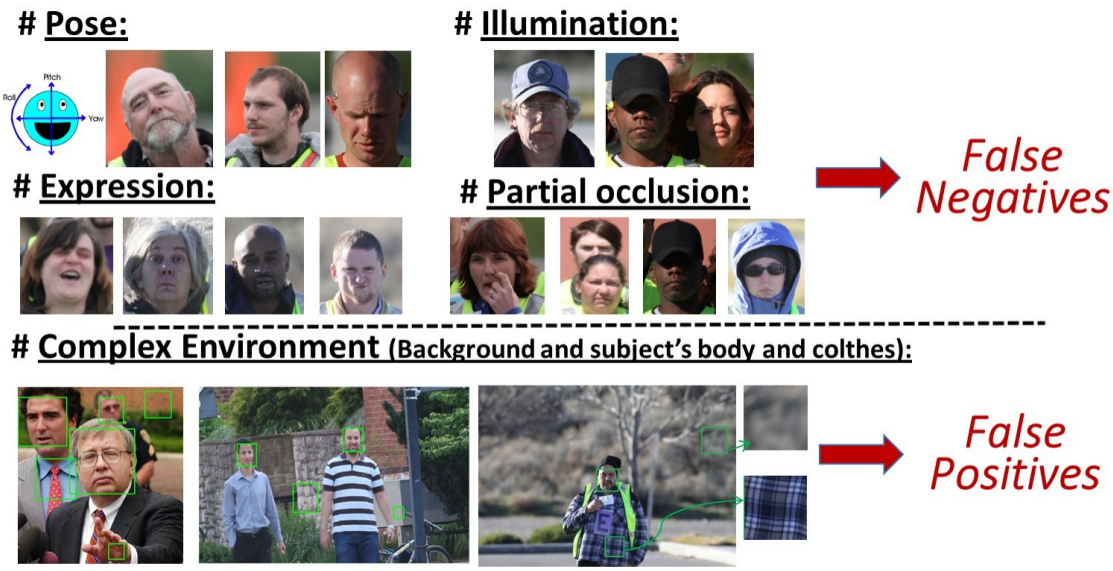


Figure 2.6: The different challenges of face detection and the two types of errors resulting from them.

train the model that can cover all the possible background environments. Examples of these challenges that can cause both types of errors for a face detector are shown in Figure 2.6.

The performance of the face detector can be measured by the ROC curve which reflects the tradeoff between these two types of errors: the false positive rate and the false negative rate implicitly represented in the true positive rate. The detector output can be seen as face candidates represented by the locations and extent in the image, and corresponding scores representing how confident the detector is about each face candidate. By thresholding these scores with different threshold values the ROC curve can be constructed. Typically, a high threshold value results in detecting only face candidates with high scores which can reduce the false positives significantly but with the price of also reducing the true positive rate (i.e. increasing the false negative rate). On the other hand, reducing the threshold value will recover more faces resulting in a high true positive rate (i.e. low false negative rate), but in the meanwhile it will also detect more false positives increasing their rates.

The proposed method in this section can be used with any generic face detector to reduce false positives by pushing the whole ROC curve to the left using other features that are different from the features used in the base detector so that it can attack the false positives from a different perspective with a minor effect on the true positives. For example, with a detector that depends on features extracted from the texture of the face (e.g. HOG or Haar) to detect face candidates, skin color can be used in a post processing step to reject false positives that resembled the texture of the face for the detector but hopefully not the skin color used in the post processing stage. The ROC curves in Figure 2.7 explains this idea.

From a different perspective, the method can also be seen as if it increases the true positive rate since if a maximum fixed false positive rate is required for the system (for example 100) then by looking to Fig 2.7 we can see that at this false positive rate there is an increase from 0.6 to 0.75 in the true positive rate. Although the method does not increase the true positive rate but it allows the detector to operate at another operating point with a higher true positive rate while maintaining the same false positive rate.

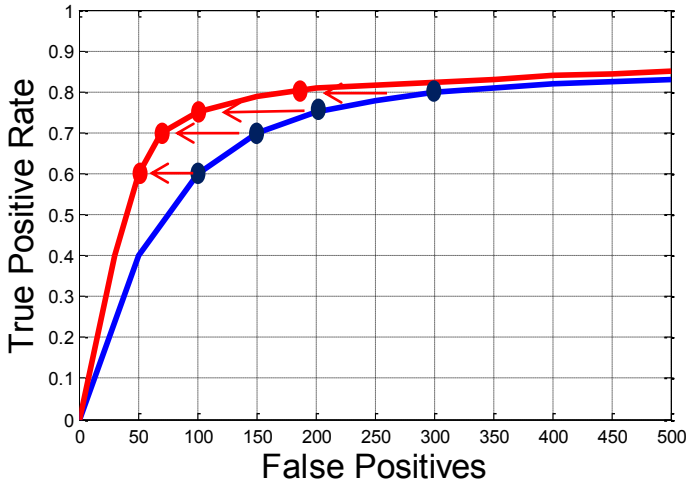


Figure 2.7: Reducing false positives result in pushing the whole ROC curve to the left and allow the use of a better operating point that can be seen at a fixed false positive rate as an increase in the true positive rate.

This work implements this idea with two different approaches that were both initially tested using the VJ face detector as the base detector which is used to find initial face candidates that are then augmented with different other information to reduce false positive or more intuitively to push the ROC curve to the left as shown in Figure 2.7 and then select the suitable operating point on the curve. The Robust Score Fusion based Face Detection (RSFFD) was proposed in the face detection competition described in Parris et al. [26] which was tested on a data set of low-light and long-distance images that possess most of the problems encountered by face detectors in a real-world setting. The performance of face detection algorithms from the academia and the industry were analyzed on this hard dataset and the initial version of the RSFFD was one of the best performers. Then in EL-Barkouky et al. [27], I proposed a modified version of the RSFFD which used saliency and skin information and will be explained in the following section. Another solution along the same line was presented in our work [28] based on utilizing a probabilistic framework for facial feature extraction combined with a skin model. The advantage is that false positive reduction is achieved using a facial feature extraction process (which is essential for most facial analysis algorithms). It simply combines the first two blocks in any facial analysis pipeline and make use of that in rejecting false positives.

2.6 Robust Score Fusion based Face Detection

The RSFFD module explained in this section uses saliency maps which will be first explained. Although the method of extracting saliency maps illustrated here is adopted from Li et al. [29], the contribution in this part is in using it into the face detection framework. Especially for FRAD applications where the Narrow Field Of View (NFOV) lenses used for large distances had some blurring effect on the background that enables

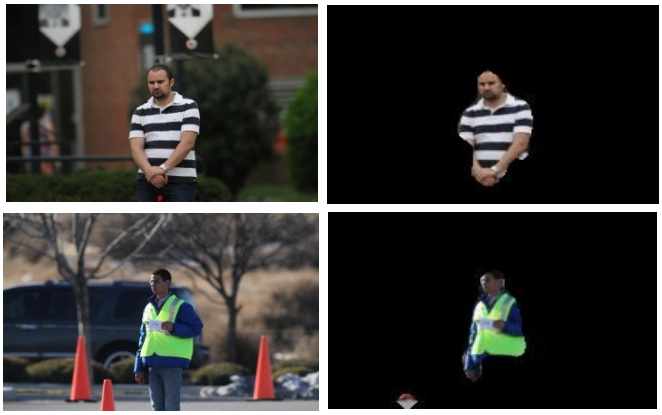
saliency maps to perform very well in detecting the subject of interest (including his face) who is usually in focus from the complex background that can be slightly blurred because it is out of focus.

2.6.1 Saliency Detection

In psychology, one of the most severe problems of perception is information overload. Therefore, the human nervous system makes an effort to determine which part of the available information is to be selected for further processing and which component needs to be discarded [30]. Koch et al. [31] proposed that different visual features that contribute to attentive selection of stimulus (e.g., color, movement, orientation) are combined into a single topographically oriented map, also known as the saliency map, which measures the relevance of information in a scene.

From a computational point of view, computer vision researchers are interested in visual saliency since it reduces computational cost in real-time object detection problems [32], [33]. Recently, Xingbao et al. [34] used saliency detection to segment pedestrians from an aerial video. In this work, it is used to help in rejecting false positives through the RSFFD framework.

For this part, the work will be based on the saliency detection framework of Li et al. [29], which uses both frequency and spatial domain analysis. In the frequency domain analysis, instead of modeling salient regions, they model the non-salient regions and suppress them later. For the spatial domain analysis, they enhanced more informative regions using a center-surround mechanism similar to that of the visual cortex. The outputs



Input Output

Figure 2.8: The input and output of saliency detection which uses saliency maps that measure the relevance of information in a scene which is in this case the subject being captured.

from these two analyses are combined to produce the final saliency map. The input and output of the process are illustrated in Fig 2.8.

Given an image $f(x, y)$, it can be transformed to the frequency domain via the Fourier transform: $f(x, y) \rightarrow F(u, v)$. The amplitude, $A(u, v)$ and phase spectra $P(u, v)$, can be computed as: $A(u, v) = |F(u, v)|$ and $P(u, v) = angle\big(F(u, v)\big)$.

To suppress repeated patterns, which correspond to non-salient regions, using spectrum smoothing, a Gaussian kernel h (with scale σ) is employed to suppress spikes in the amplitude spectrum, i.e.

$$A_s(u, v) = |F(u, v)| * h \tag{2.11}$$

The resulting smoothed spectrum $A_s(u, v)$ and the original phase spectrum are combined via the inverse Fourier transform to produce the saliency map L as:

$$L = F^{-1}\big(A_s(u, v)e^{iP(u,v)}\big) \tag{2.12}$$

The next step is to model salient pixels and regions locally, through spatial domain analysis. Li et al. [29] used Independent Component Analysis (ICA) filters to simulate the receptive fields of the visual cortex. Given the two saliency maps from the frequency and spatial domain analysis, Li et al. discussed in [29] an elaborate way to combine the two sources of information into a final saliency map which is beyond the focus of this dissertation. At the end, the saliency map is processed by several morphological operations to produce a mask that isolates the salient part in the image as shown in Figure 2.8.

2.6.2 Skin Detection

In this work, the skin detector is adopted from Conaire et al. [35] where skin pixels are detected using a Naïve Bayes classifier. They trained non-parametric histogram-based models using manually annotated skin and non-skin pixels. A total of 14,985,845 skin pixels and 304,844,751 non-skin pixels were used. Separate RGB histograms for "skin pixels" and "non-skin pixels" are created with bin size 8. Since each of the RGB channels takes values from 0 to 255, an RGB histogram that uses bins of size 8 results in an array of size 32x32x32 where the first dimension represents the 32 bins of the R channel, the second dimension represents the 32 bins of the G channel, and the third dimension represents the 32 bins of the B channel.

To construct the skin histogram SH(R,G,B), each entry of the 32x32x32 array represents a bin in the histogram and is simply filled with the normalized count of skin pixels having RGB values belonging to this bin. Similarly, the non-skin histogram NH(R,G,B) is calculated by counting the non-skin pixels. The skin model is a 32x32x32 array that contains for each bin the log likelihood of it being skin represented by:

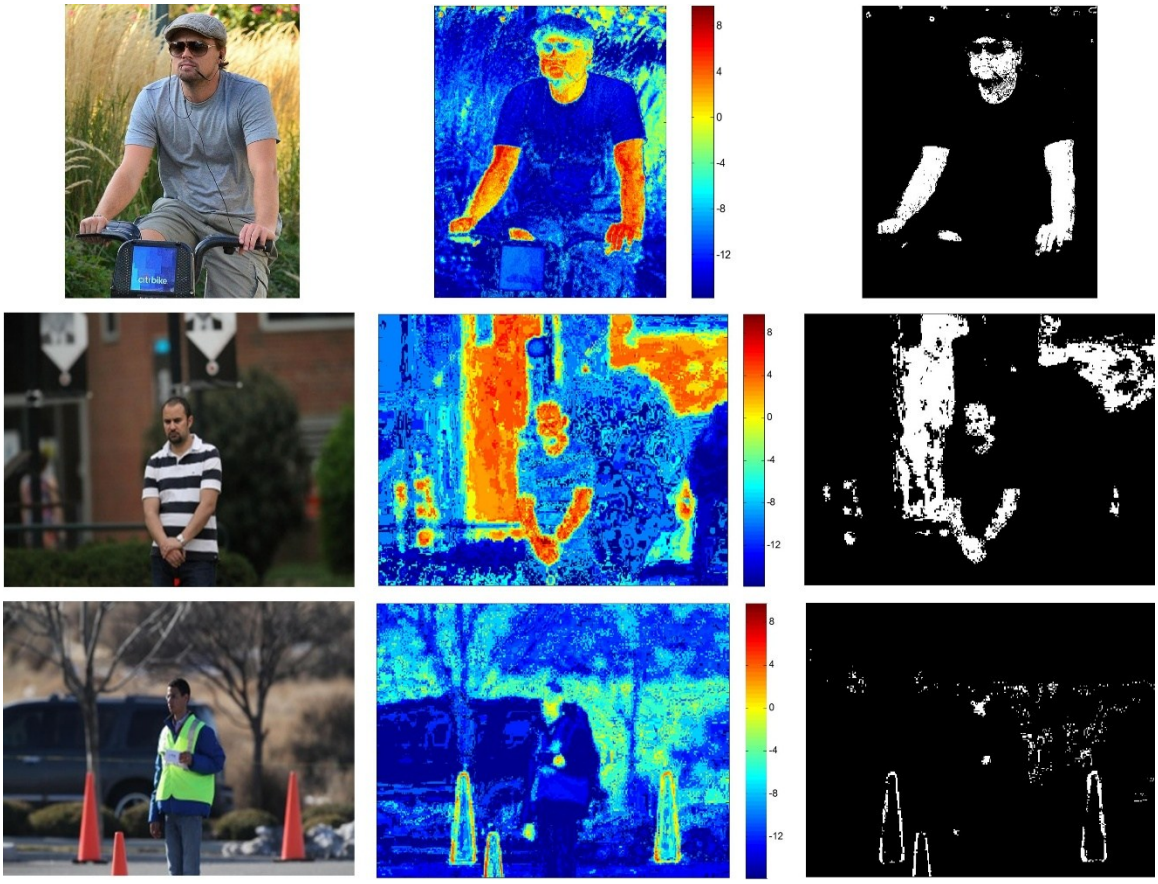


Figure 2.9: Examples of input images with their skin likelihood displayed as a color map image and skin mask displayed as a binary image. The first row shows a sample of good results and the second and third row show challenging images.

$$S = \log\left(\frac{SH(R,G,B)}{NH(R,G,B)}\right)$$

this results in a skin likelihood of 0 if SH=NH for certain RGB pixel values, positive value if SH>NH, and negative value if SH<NH. The values in the used model ranged from -15.7895 to 7.4944. For a new image, the log likelihood of each pixel in the image is computed and then a threshold of zero is applied on the result to decide skin or non-skin. The process is illustrated in Figure 2.9 using three different images, the skin log likelihood is shown using a color map image just for illustration and the skin output after applying a zero threshold is shown as a binary image.

The skin detection in general produces good results as shown in the first row of Figure 2.9 but not good enough to be used separately for face detection especially in unconstrained environments where illumination variations and complex background can be very challenging for the skin model as illustrated in the second and third rows of the same figure. In this work, a simple skin model was adopted to be very efficient computationally and it is only used for false positive reduction. It adds the valuable color information to the detector which compliments the other features used and enhances the performance.

2.6.3 Method description

The RSFFD method starts by identifying possible face candidates in the input image using Haar-like features over multiple scales using the OpenCV implementation of the Viola Jones [11] face detector (the same idea can be applied to any other detector). The overlapped rectangles from different scales are combined into a single rectangle. A first score that represents the number of combined rectangles is generated and assigned to each candidate. After detecting the facial region, the next step locates facial parts (two eyes and mouth) using the same VJ object detection approach but with a different cascade training for each facial part. The geometric structure of the face (i.e., expected facial feature locations) is taken into consideration to constrain the search space. Each candidate rectangle is given a second score that corresponds to the number of facial features detected inside it. Nose was not detected because of its low performance with the VJ framework.

In addition to detecting facial features inside the facial region, the percentage of skin pixels is also taken into account. Skin pixels are detected using the skin detctor from Conaire et al. [35], which uses non-parametric histogram-based models trained on

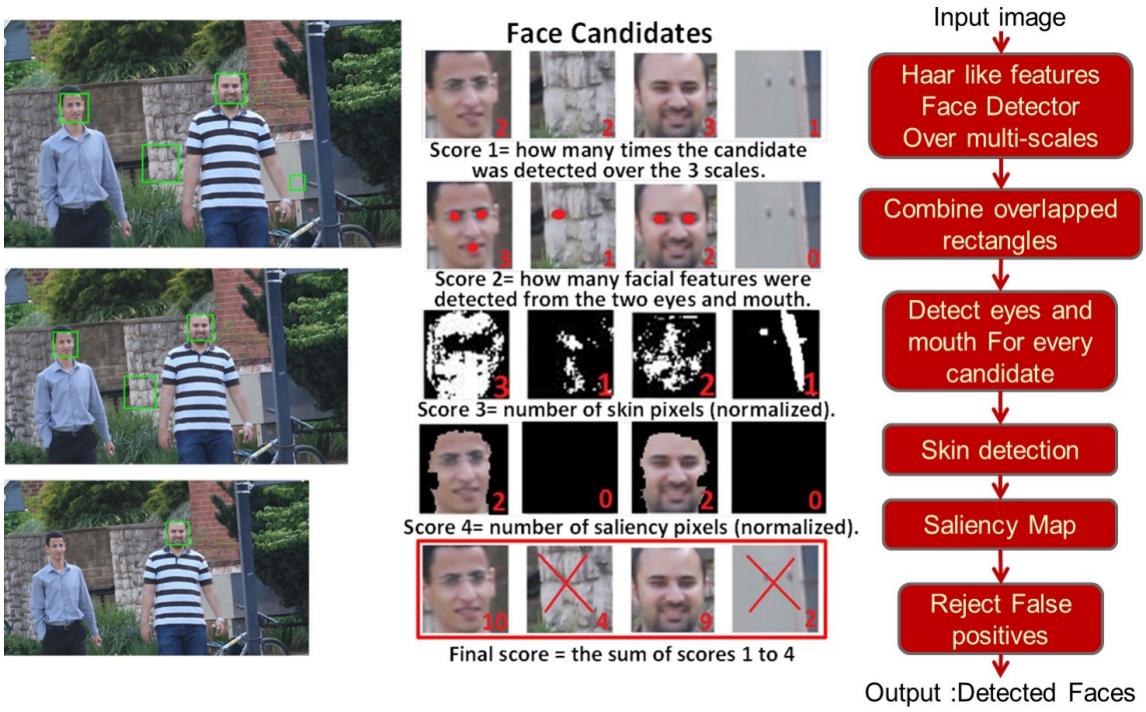


Figure 2.10: The input and output of salience detection which uses saliency maps that measure the relevance of information in a scene which is in this case the subject being captured.

manually annotated skin and non-skin pixels as illustrated in the previous subsection. The percentage of skin pixels in each candidate is transformed into the third score. In the same manner the saliency map discussed earlier is generated and again the percentage of saliency map pixels in each candidate is transformed into the fourth score as shown in Figure 2.10.

The skin was combined with the saliency because the information they provide is usually complimentary in the complex background attached with the unconstrained face detection problem. For example, sometimes a false candidate is detected in the background and it might have color that is close to skin, so at this time the saliency is used to reject. On the other hand, sometimes a false candidate is detected on the body of the subject which is part of the saliency map, so here the skin can be used to reject it. Figure 2.6 shows such examples for false positive rejection.

Every candidate face detected initially by the base face detector is assigned four scores, representing the sum of the number of overlapped rectangles, the number of facial features detected, the skin pixels percentage and the saliency map pixels percentage which are combined into a single score. Candidates with scores above a certain threshold are considered as the final detected faces.

2.7 Conclusion

This chapter provided a detailed overview of the face detection problem in general. It started with a categorization of the existing face detectors from different perspectives. Then, it focused on the details of two major general frameworks explaining how each framework models the faces for the detection task and what are the recent methods based on the same framework. It also discussed briefly the few related works that visited the partial occlusion problem in face detection. Finally, it proposed a post processing generic method for reducing the false positives of any face detector.

In this dissertation, the part based approach is adopted to build a model that explicitly handle occlusion which will result in a problem called Partial Face Detection [36] in analogy to the Partial Face Recognition problem recently defined in [37]. The partial face detection problem and its proposed solution "The Selective Part Models" are explained in the next chapter.

CHAPTER 3

PARTIAL FACE DETECTION

Faces in the wild can be considered the current target of research done in the face detection field. Partial occlusion is a major problem for analyzing these faces captured in unconstrained non-cooperative real life conditions. In this chapter, the problem of partial face detection is tackled with a novel detector that explicitly focus on partial occlusion. The model depends on modeling the face as a collection of parts that can be selected from the visible regular facial parts and possible other objects that are known to usually occlude faces such as sunglasses, caps and hands. The proposed algorithm is called Selective Part Models and it can be seen from a scene understanding point of view in the sense that it is not only detecting faces but it also suggests the visible parts of these faces and even some of the occluding objects which can help in any further analysis.

3.1 Selective Part Models

To represent faces in unconstrained settings and under partial occlusion, SPM is proposed which builds over the DPM proposed in [6]. The model consists of a root filter F_0 to capture the global appearance of the face and several part filters F_i to capture more detailed texture at twice the spatial resolution. The SPM detector selects its parts from a pool of facial part subtypes and common occluding objects to allow for variability that can capture the reach possibilities of faces under unconstrained conditions. The selection

procedure is done over two levels, the first level uses different part filters for each facial part to model different subtypes for the appearance of this part for example the eyes can be opened, closed or with eyeglasses. The second level of selection uses overlap maps to select between facial parts and possible occluding objects such as sunglasses, caps, hands and other faces. To fully describe the model, the following sections discuss the feature extraction, the detection and the training used with SPM.

3.2 Feature Extraction

In the last couple of years, almost all of the part based face detectors have adopted the HOG features or one of its modifications to describe the appearance of faces in an image. For this work the same line is followed and the SPM is also illustrated using the HOG features although the model is generic and can be applied with other types of features as well. In this section, a brief description is given for the HOG features proposed by Dalal and Triggs in [38] and its modification that was proposed in [6] which will be used here. Then a discussion is provided for the different parts used in the proposed model illustrating examples of their HOG features.

A feature map is an array with d-dimensional feature vectors as its entries. Each feature vector describes a local image patch called a cell. The image is first divided into non-overlapping cells. Then for each cell, a one dimensional histogram of gradient orientations is calculated over its pixels. The gradient magnitude and orientation are calculated at each pixel using finite difference filters [-1,0,1] and its transpose. For color images, the color channel with the largest gradient magnitude is used. The gradient orientation at each pixel is quantized into one of nine undirected orientation bins with a voting strength that depends

on its gradient magnitude to collectively build up the histogram of oriented gradients in this cell as a vector of length 9. Finally, the histogram of each cell is normalized with respect to the four 2x2 blocks that contains the cell leading to four different vectors each of length 9 that when concatenated leads to a feature vector of dimension 36 per cell. The process is illustrated in Figure 3.1.

These HOG features were modified in [6] to reduce the dimensionality from 36 to 13 while maintaining the same performance by using the 9 orientation vector together with a 4 dimensional vector that reflects the overall gradient energy in different cells around the cell. In practice, they combined the 9 undirected (contrast insensitive) gradient orientations with another 18 directed (contrast sensitive) gradient orientations ending up with the final feature map having 31 dimensional vectors. They found empirically that this combination enhances the performance for different objects.

For training data, the positive and negative examples are resized to the desired minimum face size to be detected by the model for the root filter and also resized to twice of that size for part filters. In this illustration, a root window size of 40x40 is described,

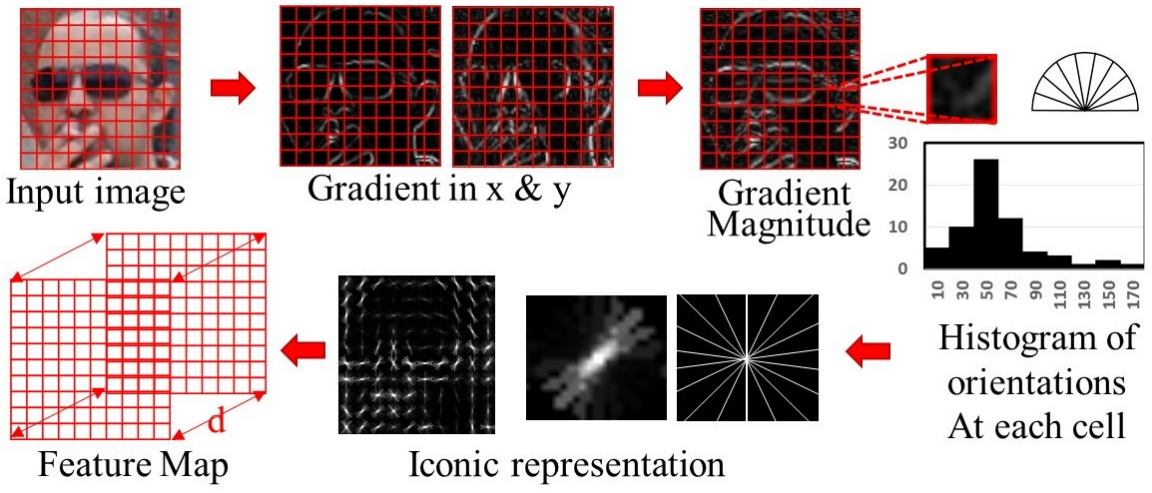


Figure 3.1: Extracting the HOG feature map from an image.

with varying part window sizes relative to a root window of 80x80 although the same concept can be used with any other sizes. For the feature map corresponding to root filters, each training example is resized to 40x40 pixels and then divided into non-overlapping cells of size 4x4 pixels resulting into 10x10 cells which are then used to calculate the 31 dimensional modified HOG features. For the feature map corresponding to part filters, each training example is resized to 80x80 pixels which is also divided into 4x4 cells yielding to 20x20 cells. The eyes part filters are 8x10 cells each anchored at the top left and right corners of the window corresponding to the root filter. The nose part filter is 8x8 cells anchored in the center of the window. The mouth part filter is 6x12 cells centered at the bottom of the window. Figure 3.2 shows an iconic representation of these different HOG feature maps.

Each facial part is classified further into three subtypes to account for variability within this part different appearances. The eyes used in training are classified manually into opened, closed, or with eyeglasses. The nose is classified according to nostrils as nose with both, only one or none of the nostrils appearing in the image where one or both of the nose openings can be hidden due to the pose of the face. The mouth is classified into closed, opened slightly with only teeth appearing or opened extremely with more than teeth appearing from the inside of the mouth. The reason behind these classifications is twofold, first the different appearance of the parts in these categories is difficult to be captured with one filter; and second the meaning of each subtype can help in further analysis of the face such as in recognition. The particular selection of the nature of these different subtypes is a design issue that can be changed as desired. Examples of different subtypes for each part are shown in Figure 3.2 with their corresponding HOG features.

The proposed model does not only detect faces and their parts but it also detects some of the occluding objects that are common to occlude faces such as sunglasses, caps and hands. The model considers these objects as possible alternatives to occluded facial parts. To comply with the dimensions of the previous illustration, these objects are considered as parts which are related to the same window of 20x20 cells similar to the facial parts. The sunglasses are of size 6x16, the caps are 16x20 and the hands are 10x10. Examples for these objects and their corresponding features are also illustrated in Figure 3.2.

The aspect ratio of each part was selected using statistics from the training data, while the exact dimensions can be adjusted based on the minimum face size required to be detected for different applications. For faster processing, the cell size can be increased from

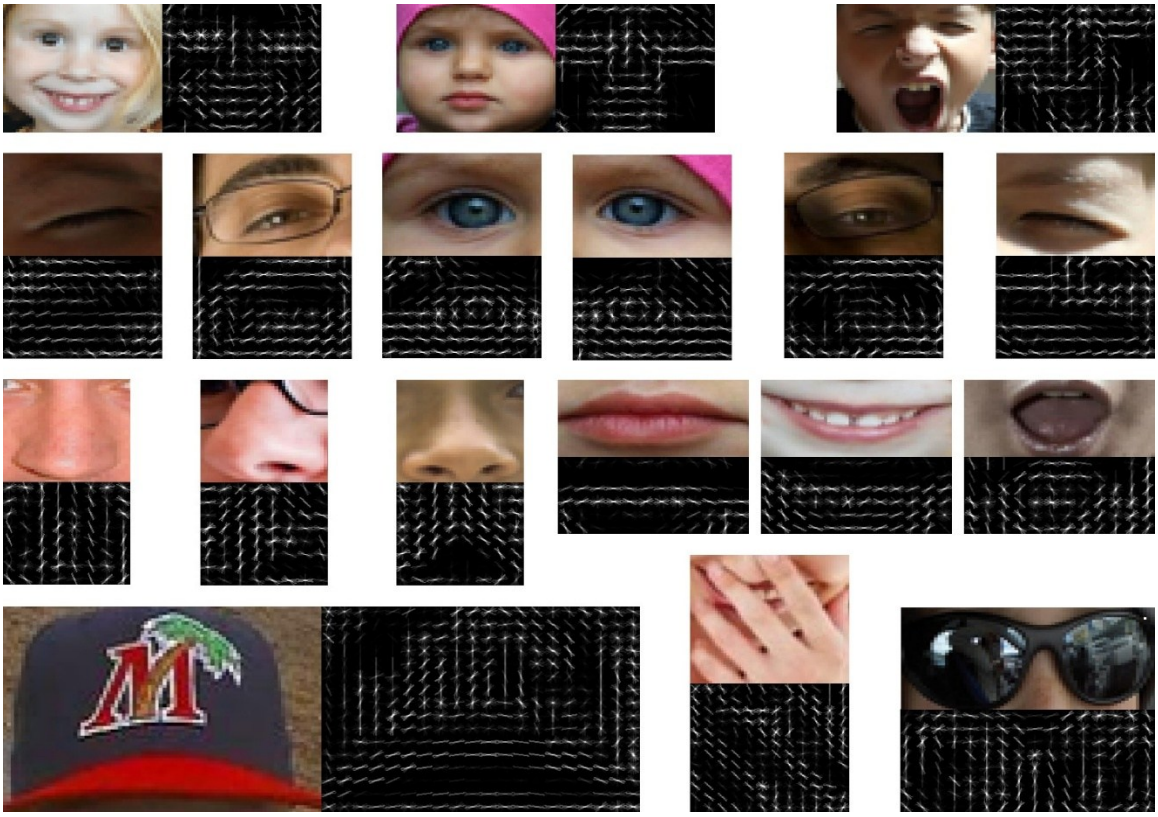


Figure 3.2: Images and feature maps for examples of the whole face in the first row and the remaining rows are the facial parts with their different subtypes along with some of the common occluding objects. The parts are at twice the resolution of the whole face and relative sizes are maintained in the figure.

4x4 to 8x8 which will reduce the number of cells processed per root or part by one fourth resulting in faster overall processing with a slight degradation in the performance.

3.3 Detection

For the detection, the sliding window approach is used to test image windows over different locations and scales. The process starts by building the feature pyramid H from the input image which enables the detection of faces over multi-scales using filters of fixed size. The response of the i^{th} filter F_i in the l^{th} level of the feature pyramid is calculated by

$$R_{i,l}(x, y) = F_i \,.\, \varphi_a\big(H, (x, y, l)\big) \tag{3.1}$$

where φ_a is the appearance HOG feature vector in a window of H with top left corner at (x, y) and its size is defined by the size of the filter. To capture the deformable variations in the parts of different faces or different occluding objects, we use the generalized distance transform proposed in Felzenszwalb et al. [21] to find the optimum locations for the parts with respect to the root location. The updated response of the i^{th} part filter in this optimum location is given by

$$D_{i,l}(x, y) = \max_{dx,dy} \left[R_{i,l}(x + dx, y + dy) - d_i \,.\, \varphi_d(dx, dy)\right] \tag{3.2}$$

where $\varphi_d = (dx, dy, dx^2, dy^2)$ is the deformation feature and d_i is a four dimensional vector specifying its coefficients. This transformation spreads high part scores to nearby locations according to a deformation cost.

The main contribution in the SPM is to allow selecting the model parts from a pool of parts that includes different subtypes of each of the main facial parts namely the two eyes,

nose and mouth and common occluding objects that can hide these parts like caps, sunglasses and hands. The overall score of any window is calculated by adding the root score on this window to the sum of the selected part scores using

$$S_t(x, y, l) = S_r(x, y, l) + \sum_{P_i \in S} S_{P_i}(x, y, l) \tag{3.3}$$

$$S_r(x, y, l) = R_{0,l}(x, y) \tag{3.4}$$

$$S_{P_i}(x, y, l) = D_{i,l-\lambda}(2(x, y) + v_i) + b_i \tag{3.5}$$

The root filter is at level l of the pyramid and the part filters are at level $l - \lambda$ which is twice the resolution of l, v_i is the anchor position for part i relative to the root position, b_i is a bias term to make the scores of different parts comparable for selection and using the same threshold. The summation is over the parts belonging to a set S containing the selected parts for this window.

Figure 3.3 explains how the set S in equation (3.3) is found from the possible pool of facial parts P_1 to P_4 including their subtypes and the occluding objects P_5 to P_7. The upper part of the figure shows the overlap maps of these parts and their anchor points with respect to the root window, while the second row shows some possible scenarios clarifying how the selection is made using the overlap maps and their corresponding scores S_{P_i}. The algorithm selects the parts in S as follows:

1. Initialize S with P_1 to P_4 despite of their scores by comparing the different subtypes of each part and selecting the largest score.

2. Add to S each of the objects P_5 to P_7 only if their respective scores are higher than a threshold.
3. Check the overlap of each added occluding object with the corresponding facial parts according to the overlap maps and compare the scores of overlapped parts to keep from them only the parts with the highest score and remove the others from S.

For example, in the right lower image of Figure 3.3 the set S starts as $\{P_1, P_2, P_3, P_4\}$ then by checking the scores of P_5 to P_7 for this window the set S

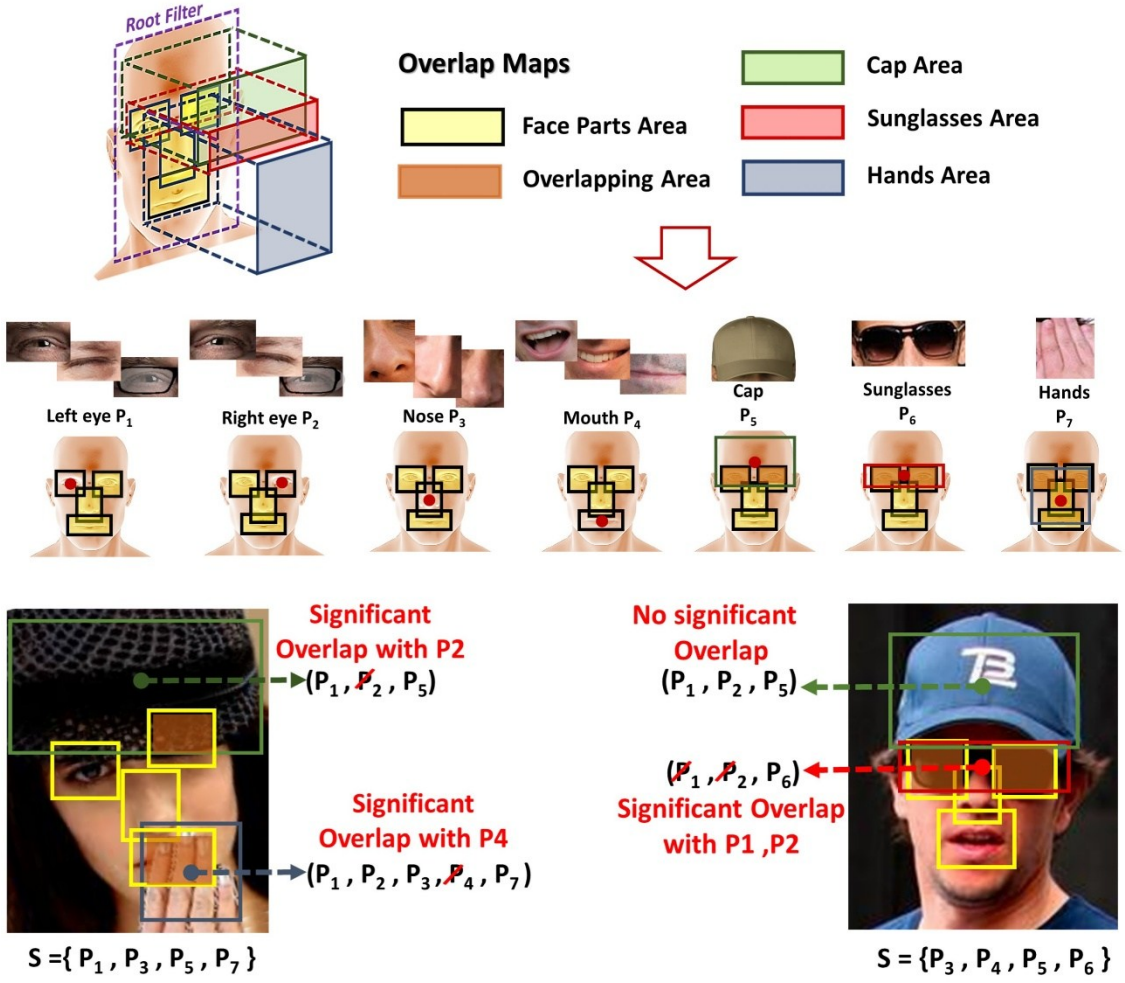


Figure 3.3: Selective Part Models: The upper part shows the overlap maps of different parts and their anchor points with respect to the root window. The lower part shows two examples of the selection process using overlap maps and the final parts in the set S (below the face).

becomes $\{P_1, P_2, P_3, P_4, P_5, P_6\}$. From the overlap maps, a possible overlap of P_5 (the cap) should be checked with P_1 and P_2. In this example, they do not overlap significantly so the set S is kept as it is and the algorithm moves to P_6 (the sunglasses) which should be checked for possible overlap also with P_1 and P_2. Since in this example they do overlap significantly and the score of P_6 is higher than P_1 and P_2, then P_1 and P_2 are removed from S and the final set S becomes $\{P_3, P_4, P_5, P_6\}$.

Another example is in the left lower image of Figure 3.3 where the set S starts as $\{P_1, P_2, P_3, P_4\}$ then by checking the scores of P_5 to P_7 for this window the set S becomes $\{P_1, P_2, P_3, P_4, P_5, P_7\}$. From the overlap maps, a possible overlap of P_5 (the cap) should be checked with P_1 and P_2. In this example, P_5 does not overlap significantly with P_1 but it does overlap with P_2 so the algorithm keeps P_1 as it was in S and compare the score of P_2 and P_5 to conclude removing P_2 from S since it has a smaller score and the set S becomes $\{P_1, P_3, P_4, P_5, P_7\}$. Then the algorithm moves on to P_7 (the hand) which should be checked for possible overlap with all parts. Since in this example P_7 only overlaps significantly with P_4 and the score of P_7 is higher than that of P_4, then P_4 is removed from S and the final set S becomes $\{P_1, P_3, P_5, P_7\}$.

Equation (3.3) can then be used to calculate the total score of this window from these selected parts. If regular DPM was used with this face it could have resulted in either an undetected face or a weakly detected face with a low score because of the low scores of the occluded parts but the advantage of our model is that it detects these faces with high scores and hence allows the use of a higher threshold that can reduce false positives. Besides that, it provides more information to the following facial analysis steps about the visible parts

of the face. For example, knowing that the eyes are covered by sunglasses suggests excluding them from the recognition signature.

To solve the problem of faces occluded by other faces, two thresholds are used for the score of each window; an initial low threshold T_l that passes all the possible face candidates including false positives and possible weakly scored faces because part of it is occluded by other faces. These faces have two properties that distinguish them from false positives: first the scores of the visible parts are high while the scores of the occluded parts are low causing the total score to be low, and second these occluded parts overlap with other strong faces. We use these two properties to verify that this is a face partially occluded by another face and increase its score above the final high threshold T_h that is used to reject other false positives that do not share these two properties. The process is illustrated in Figure 3.4 where four face candidates have scores higher than T_l, with only two of them higher than T_h shown in green while the other two lower than T_h shown in red (dashed). The occluded face satisfy the overlap condition with one of the strong faces and two of its part scores are visible reflecting high part scores while the other two are occluded causing its final low

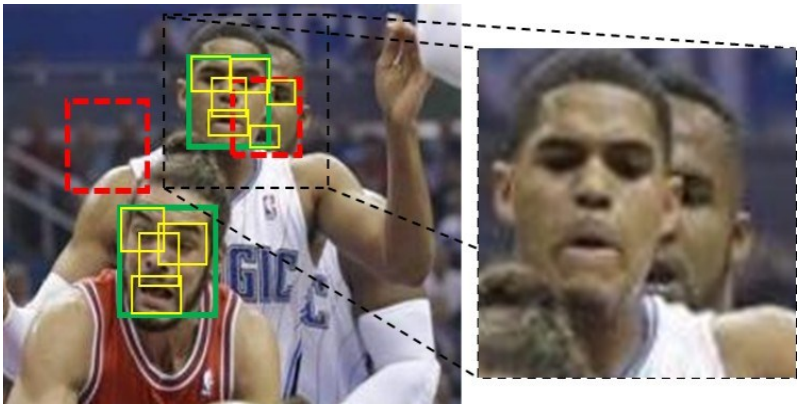


Figure 3.4: Increasing the low score of faces occluded by other faces if it overlaps with a high score face and at least two of its part scores are high. Face candidates with high scores are shown in green, face candidates with low scores are shown in red (dashed) and only parts with high scores are displayed for illustration.

score. In this case, the score of this weak face is increased to pass T_h while the false positive remains low to be rejected by T_h.

Finally, the execution time of face detectors is a very important factor because face detection is usually the first step in other facial analysis applications. The advantage in this system is that it does not only detect the faces but it also provides the subsequent analysis with more information about the part locations, their subtypes and some occluding objects. The system also can be used with three levels of accuracy where it can only use root and facial parts for the fastest execution time, or it can add to that checking for occluding objects as additional optional parts, or it can also add subtypes for each facial part. This means that the number of part filters evaluated ranges from 4 for only facial parts, or 7 for adding occluding objects up-till 15 for adding also subtypes. The appropriate method should be selected depending on the challenges in the testing database. Detailed discussions about the detection time and a more elegant solution are provided in the next chapter.

3.4 Training

The SPM root and facial parts are trained using a fully supervised dataset with a complete annotation for the location of the face and its 4 parts. The model uses three components, one captures the frontal and near frontal faces (faces in which all facial parts are visible) and the other two capture the profile and near profile faces (faces with one eye not visible). The profile faces needed two components to be able to detect both the faces looking to the right and the faces looking to the left. The rest of the poses are recovered by the deformation embedded in the model for part locations as illustrated in Figure 3.5 through different training samples from each component. The upper part of the figure

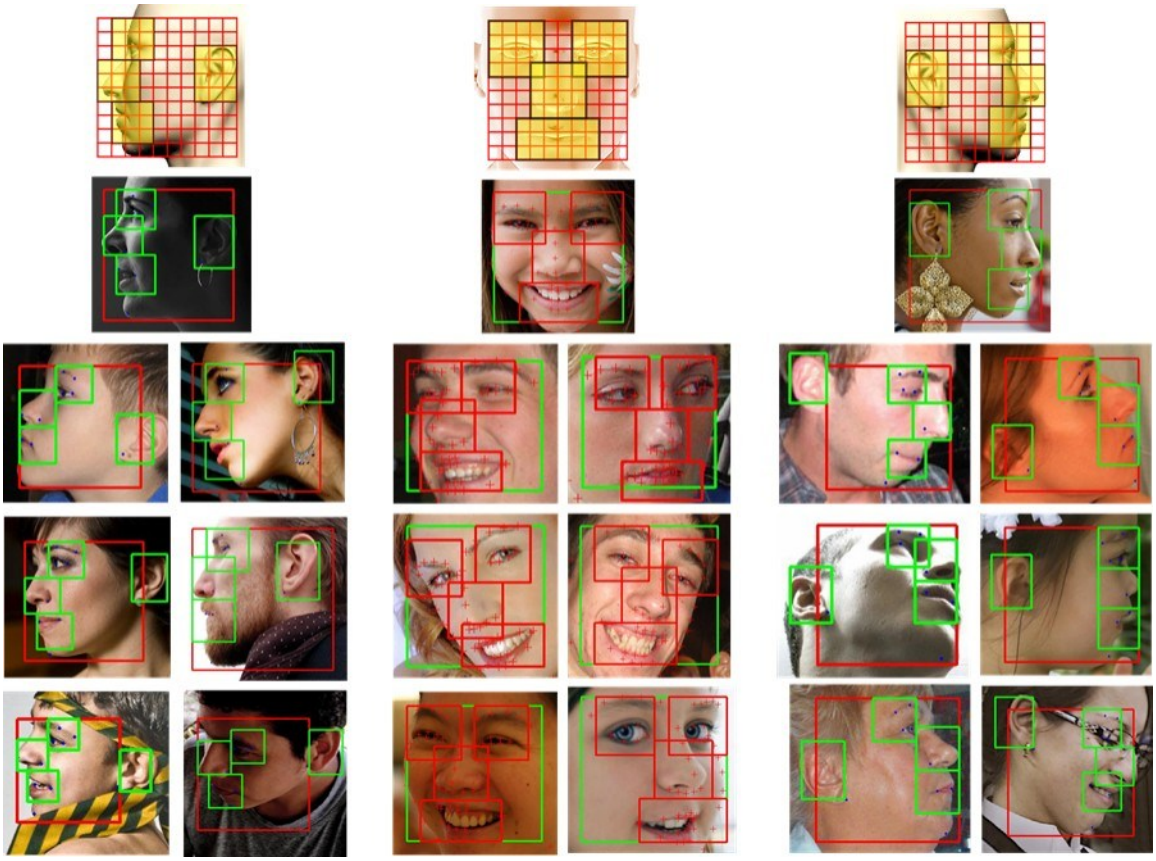


Figure 3.5: The face model is trained using three components: The frontal component is shown in the middle with different training samples that illustrates how near frontal faces can be captured by this component through parts deformations. The right profile and left profile components are shown in the right and left respectively with different training samples to illustrate the variations captured by parts deformation.

shows the anchor positions selected for each part with respect to a 10x10 cell mesh that represents the whole face. This corresponds to a root face of size 5x5, but shown over a 10x10 mesh because parts are computed at twice the resolution of the root. The lower part of the figure shows several training samples to illustrate how different poses are recovered through these three components. It can be seen from the figure that the parts of the frontal component are the two eyes, the nose and the mouth, while the parts of the profile components are the visible eye, the nose, the mouth and the visible ear.

Face pose in general is defined by three angles: Yaw, Roll and Pitch as illustrated in Fig 3.6. The three components of the SPM are mainly to handle the yaw angle variations,

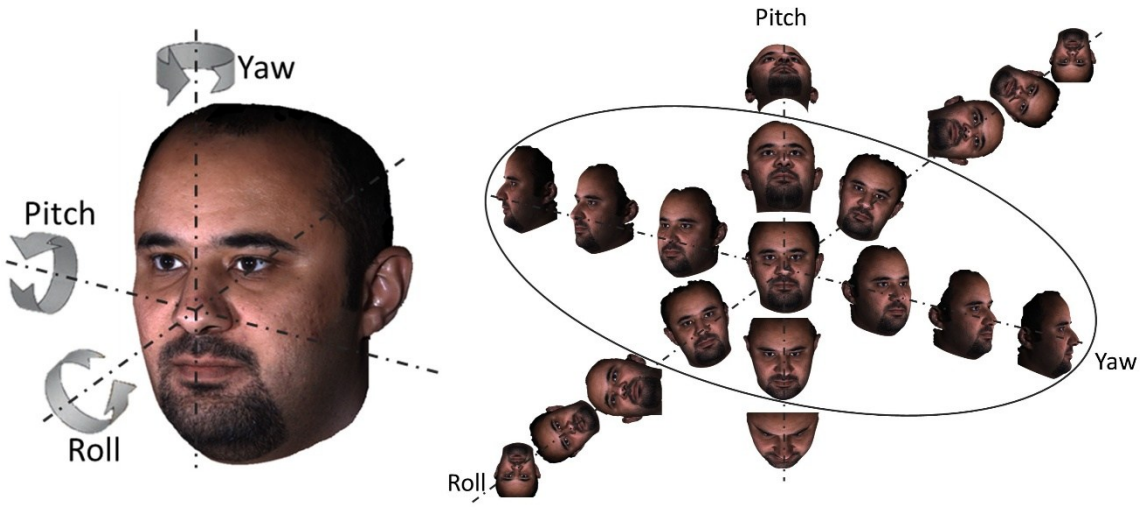


Figure 3.6: Illustration of all possible face poses using Yaw, Pitch and Roll angles. The range of extreme face poses in Roll and Pitch is not detectable by the three components of the SPM.

where yaw angles roughly from -45^o to 45^o can be detected by the frontal component, yaw angles roughly from 45^o to 90^o can be detected by the left profile component, and yaw angles roughly from -45^o to -90^o can be detected by the right profile component. Small roll and pitch variations are handled by the deformation. Large roll angles that exceeds 30^o will need several separate components to be detected, but it is not of interest to this work because it is not common to have such high in-plane rotation in face images. If these rotated faces are of interest they can be trained in separate components easily by rotating the training data. Similarly large pitch angles result in huge change in the facial parts appearance and cannot be detected without additional components. The right part of Figure 3.6 shows different simulated variations of the three angles where the ellipse encloses all the poses that can be detected by the SPM three components and their deformations.

3.4.1 Automatic Part Annotation

To train a robust face detector that can handle faces in unconstrained conditions, a large dataset of faces captured in real life images is needed. In addition, this large number of

training faces must be annotated. The annotation for a fully supervised training of the SPM requires the bounding boxes for the face and its four parts which is not available in any public face database. However, there exists several databases which are annotated with facial landmark points such as the Helen database [39], the LFPW database [40], and the AFLW database [41]. In this section, an automatic algorithm is proposed to transform the annotation of these facial landmark points to the bounding boxes of the face and its four parts which match the requirements for training the SPM. In this way, the manual annotation of thousands of images can be avoided. First, a brief description of these "in-the-wild" public datasets is provided:

Helen dataset: The Helen dataset consists of 2,330 faces in 2,330 high resolution images collected from Flickr with a broad range of appearance variations but with no profile views. The faces are annotated with detailed 194 landmark points using Amazon Mechanical Turk (Mturk) as shown in Figure 3.7. Each image is cropped to include the annotated face and a proportional amount of the background which may include other unannotated faces.

Helen Dataset LFPW Dataset AFLW Dataset

Figure 3.7: Examples of training images from Helen, LFPW and AFLW datasets.

LFPW dataset: The Labeled Face Parts in-the-wild (LFPW) dataset consists of 1432 faces in 1,287 images downloaded from google.com, fickr.com, and yahoo.com. The images contain large variations including pose, expression, illumination and occlusion but again with no profile views. The faces are annotated with 35 landmark points using Mturk as shown in Figure 3.7.

AFLW dataset: The Annotated Facial Landmarks in the Wild (AFLW) contains 25,993 faces in 21,997 images downloaded from Flickr. The images contain a wide range of natural face poses and occlusions including profile faces. The faces are annotated with 21 landmark points that are marked only if the corresponding point is visible as illustrated in Figure 3.7. The dataset provides a SQLite database to allow retrieving faces with specific poses or specific sets of visible landmarks among many other useful queries.

For the training of **near frontal component**, around 7000 faces were retrieved from Helen, LFPW and AFLW datasets. In Helen and LFPW, I used the re-annotation of the two datasets through the 300 Faces in-the-wild Challenge [42] which provides a common annotation scheme of 68 points per face as illustrated in the first row of Figure 3.8. Those 68 points are distributed over the parts as 11 point for each eye, 9 points for the nose, 20 points for the mouth and 17 points for the face boundary. Figure 3.8 explains the details of the algorithm for automatic annotation of the bounding boxes for the face and its four parts from the landmark points of images from Helen and LFPW datasets.

The inputs for the algorithm are the training images I_i and the 68 landmark points P_i where $i \in \{1, 2, \dots, m\}$ and m is the total number of positive training images. The outputs from the algorithm are the Face Bounding Box FBB_i and the Parts bonding Boxes BB_{ik} of

each training image I_i for $i \in \{1, 2, \dots, m\}$ and $k \in \{1,2,3,4\}$. The face and parts bounding boxes are aligned over a common mesh of 10x10 cells for all the training data such that each part has the same size in cells for all the data, also the deformations of parts are only allowed in steps of complete cells. The parameter ppc_i accounts for the number of pixels per cell for each image I_i which absorbs the differences in sizes of the training images and unify all of them on the same mesh of 10x10 cells.

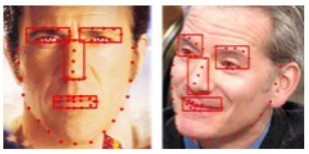
The algorithm applies five main steps for each image. The first step is to find initial bounding boxes for each part such that it tightly contains all of its points. The bounding box in general is defined by four parameters: *X* and *Y* representing the coordinates of the upper left corner point of the box together with *W* and *H* representing the width and height of the box respectively. These initial part boxes will differ in size and aspect ratio from one image to the other as shown in the two examples of Figure 3.8. The second step is to find the initial face bounding box such that it tightly contains all the parts points. This box is adjusted in the third step to be a square and an exact multiple of 10 cells which define the parameter ppc that measures the differences in size between different training images.

In the fourth step, each part is expanded to match its preset dimension in cells which in this case is 4x4 cells for the two eyes and the nose; and 3x6 cells for the mouth. Each part is expanded around the center of its initial bounding box. After that the initial point of the part is shifted to match the starting point of the closest cell so that the parts are exactly aligned over the face mesh of cells as shown in Figure 3.8. The final step accounts for rounding errors by comparing the final box of each part with its landmark points and with the whole face bounding box. For example in Fig 3.8, by comparing the eyes locations in the first image beside step 4 with the face bounding box, it is clear that they are shifted

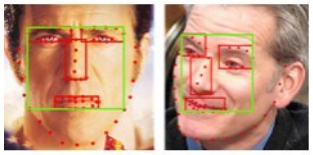
Input: $(I_i, P_i),\ i \in \{1, 2, \dots, m\}$ where:
I_i: training images, m: number of positive training images,
$P_i = (x_{ij}, y_{ij}),\ j \in \{1, 2, \dots, 68\}$: landmark points of I_i

For $i = 1 : m$

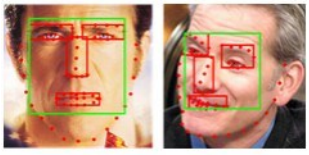
1. Find initial Bounding Boxes for each part BB_{ik}:
   $BB_{ik} = (X_{ik}, Y_{ik}, W_{ik}, H_{ik}),\ k \in \{1,2,3,4\}$ such that
   $X_{ik} = \min(x_{ij}), Y_{ik} = \min(y_{ij})\ \forall j \in Part_k$
   $W_{ik} = \max(x_{ij}) - X_{ik}, H_{ik} = \max(y_{ij}) - Y_{ik}\ \forall j \in Part_k$



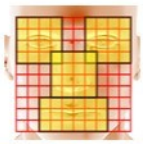
2. Find initial Face Bounding Box FBB_i:
   $FBB_i = (X_i, Y_i, W_i, H_i)$, where
   $X_i = \min(X_{ik}), Y_i = \min(Y_{ik}), W_i = \max(X_{ik}) - X_i,$
   $H_i = \max(Y_{ik}) - Y_i, \quad k \in \{1,2,3,4\}$

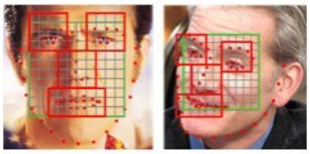


3. Adjust FBB to be square and exact multiple of 10 cells:
   $W_i = 10 * ceil(W_i/10)$, $H_i = 10 * ceil(H_i/10)$
   $if\ W_i > H_i$: $Y_i = Y_i - (W_i - H_i)/2$; $H_i = W_i$
   $else$: $X_i = X_i - (H_i - W_i)/2$; $W_i = H_i$
   Pixels per cell: $ppc_i = W_i/10$

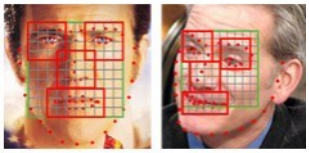


4. Expand each part to match its dimension in cells:
   Center of BB_{ik}: $X_{ikc} = X_{ik} + W_{ik}/2, Y_{ikc} = Y_{ik} + H_{ik}/2$
   $W_{ik} = 4 \times ppc_i, H_{ik} = 4 \times ppc_i$ for , $k = 1, 2, 3$
   $W_{ik} = 6 \times ppc_i, H_{ik} = 3 \times ppc_i$ for , $k = 4$
   $X_{ik} = X_{ikc} - W_{ik}/2, X_{ik} = Y_{ikc} - H_{ik}/2$
   $X_{ik} = X_i + round((X_{ik} - X_{ik})/ppc_i) \times ppc_i,$
   $Y_{ik} = Y_i + round((Y_{ik} - Y_{ik})/ppc_i) \times ppc_i,$





5. Adjust BB_{ik} according to several heuristics to account for rounding error problems by comparing the final box of each part with its landmark points and with FBB_i . If needed add or subtract ppc_i to X_{ik} or Y_{ik} to move the part one cell.



Output: $(I_i, FBB_i, BB_{ik}, ppc_i),\ i \in \{1, 2, \dots, m\}$ where:
I_i: training images, m: number of training images,
FBB_i: Face Bounding Box, BB_{ik}: Parts Bounding Boxes of I_i
Where $k \in \{1,2,3,4\}$, ppc_i: number of pixels per cell for I_i

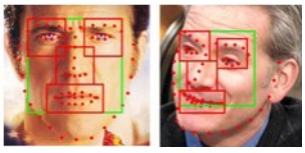


Figure 3.8: The automatic algorithm for annotating the bounding box of the face and the bounding boxes of its four parts from the landmark points using training images from Helen and LFPW datasets.

unnecessarily outside the face while they can stay inside it and still contain all its points so they are shifted one cell down. Similarly in the second image, the mouth is shifted one cell to the right to better match its points. This algorithm is important because it allows for extracting the appearance features and the deformation features required for the training of SPM in a fully supervised setting where the locations of parts are annotated over all the images on the same mesh of cells.

In the AFLW dataset, the faces with absolute yaw angle less than 45^o, absolute roll angle less than 25^o and absolute pitch angle less than 40^o were retrieved, then only faces with all 19 facial points visible (all 21 points except the 2 ear points) are kept. Those 19 points are distributed over the parts as 6 points for each eye, 3 points for the nose, 3 points for the mouth and 1 point for the chin. The algorithm in Figure 3.8 does not depend on the number of points per part so it was used with slight modifications to find the bounding boxes of the face and its four parts from these 19 points of the AFLW dataset.

For the **profile components**, around 2500 faces are retrieved from AFLW dataset by searching for faces with yaw angles between 45^o and 90^o or -45^o and -90^o, absolute roll angle less than 25^o and absolute pitch angle less than 40^o. Only faces with at least one visible point per each of the visible eye brow, the visible eye, the nose, the mouth and the visible ear are used. Right looking faces are horizontally flipped so that the final profile dataset are all looking to the left to train the left profile component. On the other hand, the right profile component is trained by horizontally flipping all the faces to look right.

The algorithm of Figure3.8 was modified to find the bounding boxes of the face and its four parts. The four parts in the profile case are the visible eye, the nose, the mouth and the

visible ear. The number of points per each part is small and not consistent based on the exact pose because in this database if a point is not visible, it is not annotated. The visible eye contains 4 to 6 points, the nose and the mouth each contains 2 points, and the chin and the visible ear each contains 1 point. Due to that small number of points in the nose and the mouth, the algorithm used the points of the eye and the mouth together with the points of the nose to initialize the bounding box of the nose. Similarly, it used the points of the nose and the chin together with the points of the mouth to initialize the bounding box of the mouth. For the ear, the single point provided in the annotation was considered the lower right corner of the ear bounding box for the right profile faces and the lower left corner for the left profile faces. The same mesh of cells was aligned to the face and its parts to provide the final bounding boxes of the face and its four parts as shown in Figure 3.5. It is worth mentioning, that in the profile case the eye, nose and mouth were set to a size of 3x3 cells while the ear was set to a size of 4x3 cells. These sizes were decided empirically from the data using the average sizes of each part and by trying different sizes and visually inspecting the results.

The **occluding objects** training images were collected from the web with 500 images per each category namely the sunglasses, caps and hands. The algorithm to annotate these objects is based on manually selecting two points that represent the upper left corner and the lower right corner of the bounding box, then automatically adjust the bounding box to satisfy the required size in cells that matches the face mesh of 10x10 cells. The sunglasses size was selected as 4x10 cells, the cap size was selected as 6x10 cells, and the hand size was selected as 5x5 cells. Figure 3.9 shows examples of the manual selection of the two points and the final bounding box which is slightly modified to match the preset aspect

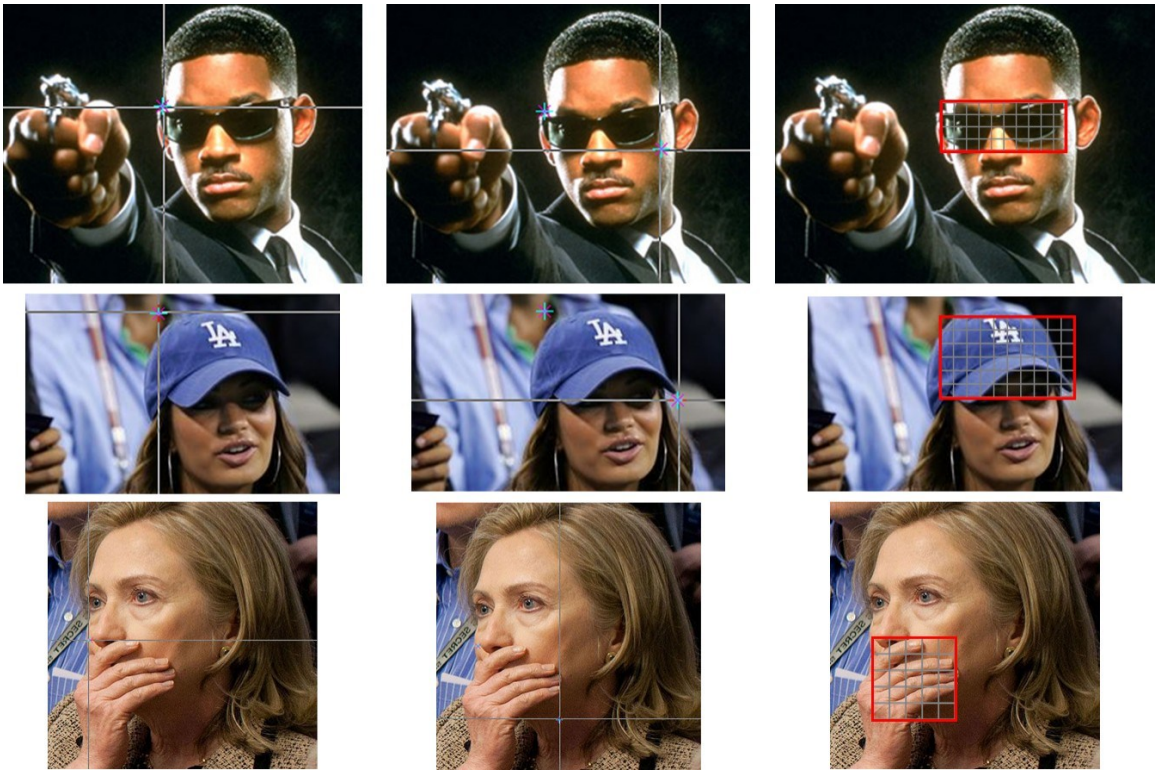


Figure 3.9: Examples of training images for sunglasses, caps and hands illustrating the manual annotation of each object using two points.

ratio and the corresponding size in cells. The left column shows the manual selection of the first point in each case, while the middle column shows the selection of the second point. The right column shows the final bounding box with the mesh of cells overlaid on top of it for clarification. The output of this manual annotation algorithm is the bounding box of each object and the parameter *ppc* representing the number of pixels per cell. These objects are trained separately using only appearance features. The deformation feature would require more complicated annotation that is unpractical in this case.

In addition, one more part was added which represents the upper part of the body including the head and the shoulders. The annotation for this part used 1000 images obtained from the frontal data by selecting the images that has the upper part of the body completely visible. The size of the part was designed to be 8x8 cells which is applied in a

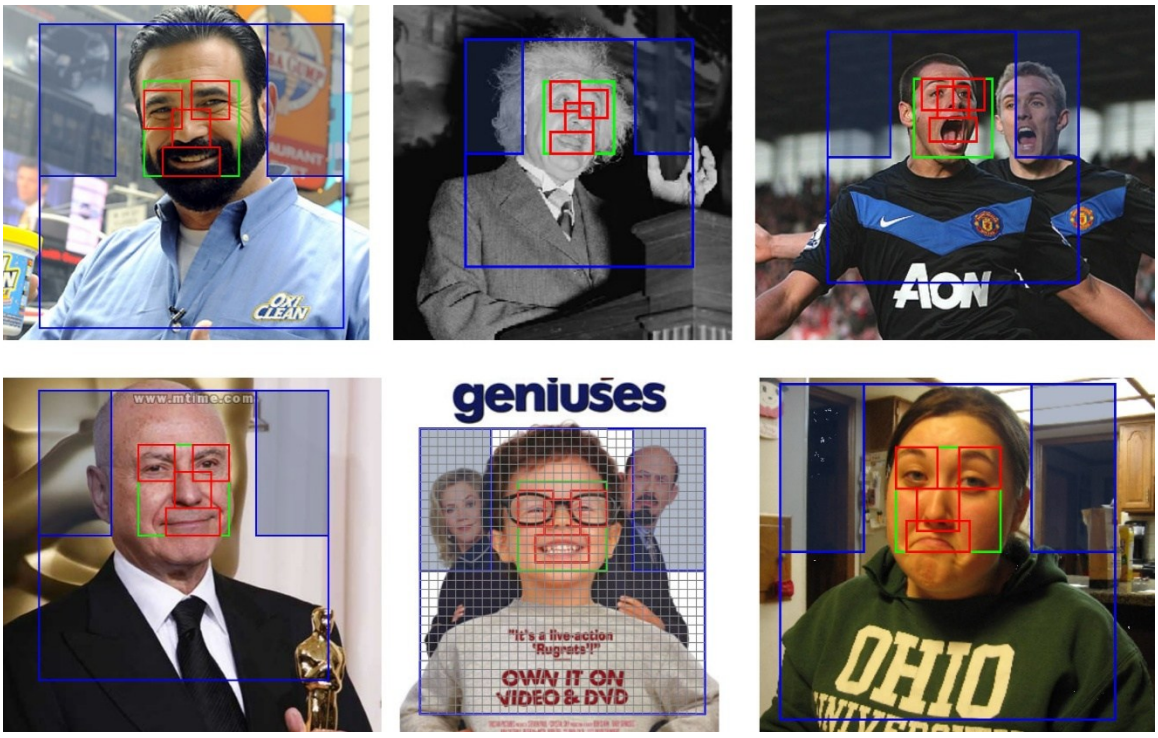


Figure 3.10: Examples of the training data for the upper body part which is selected from the frontal face data.

resolution corresponding to half the resolution of the root. This means that in the resolution of the root it will be 16x16 and in the resolution of the parts it will be 32x32 cells. This is clear on the mesh overlaid on the center image of Figure 3.10 where the root filter appears in green corresponding to 10x10 cells and the upper body part corresponding to 32x32 cells all of them are shown relative to the parts resolution. In the detection the upper body will be used in one level and the corresponding root will be in double this resolution while the corresponding facial parts will be in double the root's resolution.

The annotation shown in Figure3.10 was made in a semi-automatic way by considering the root bounding box as 10x10 cells and annotate the upper body bounding box around it with a size of 32x32 cells such that there is 11 cells to the right and left of the root and 6 cells up with 16 cells bellow the root as illustrated in the figure. Many images were then eliminated automatically because the upper body bounding box went outside the image,

the rest were inspected manually to make sure that they include a reasonable part of the shoulders and that the general silhouette of the head and shoulders is captured. Due to the nature of the shoulders being wider than the head, the blue shaded region of Figure 3.10 was replaced by zeros in the training so that it does not affect the model since it is always just part of the background. In the detection the corresponding regions of the model coefficients are also zeros to reduce the effect of the variations in the background over the detection of the upper body part.

3.4.2 The Training algorithm

The face detector is a classifier that classifies each window coming from the sliding window approach over an image pyramid as a face or a non-face. The training of any classifier requires a training set $(V_1, l_1), \dots, (V_M, l_M)$ where $V_i \in R^n$, $l_i \in \{-1, +1\}$ and M is the total number of training samples. In this case +1 represents the label of the face class and -1 represents the label of the non-face class. The feature vector V_i has a size n and it is extracted from the positive samples and the negative samples to represent the image information. In the SPM model, this vector is composed of a concatenation of the features extracted from the root and the facial parts while the occlusion parts are trained separately.

For example, a positive sample in the training set consists of an image containing a face, the face bounding box, the parts bounding boxes and the number of pixels per cell $(I_i, FBB_i, BB_{ik}, ppc_i)$ as was illustrated in the output of the annotation algorithm of Figure 3.8. The image is first cropped with the face bounding box, and resized to make the number of pixels per cell equals 8. Which makes a face of size 5x5 cells contains 40x40 pixels. From each cell, a HOG feature of size 31 is calculated and concatenated for all cells leading

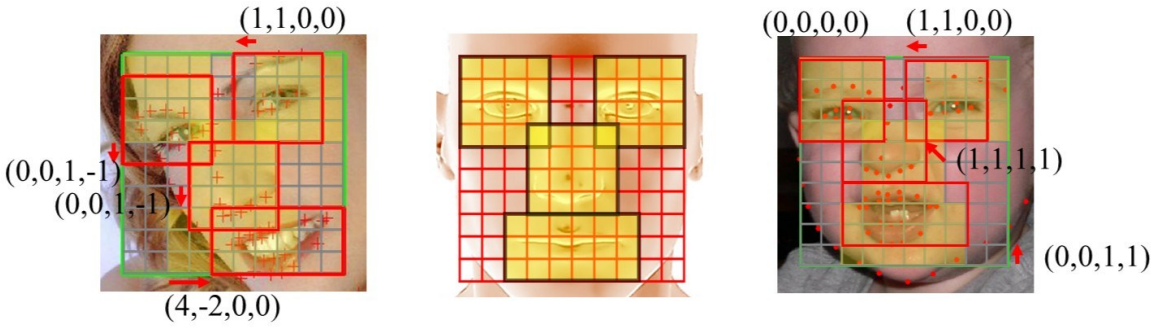


Figure 3.11: Examples for the calculation of the deformation features $\varphi_d = \left(dX_k^2, dX_k, dY_k^2, dY_k\right)$ for each part.

to a root vector of size 5x5x31=775. Similar cropping and resizing is done for each part leading to a vector of size 4x4x31=496 for each of the two eyes and the nose; and a vector of size 3x6x31=558 for the mouth. Besides the HOG appearance feature, there is a deformation feature vector of size 4 for each part. In addition, a 1 is added to the vector to correspond to the bias term of the model to result in a total vector of size $(775)_{root} + (496+4)_{LEye} + (496+4)_{REye} + (496+4)_{Nose} + (558+4)_{Mouth} + (1)_{bias}$ =2838.

The deformation feature captures the movement of that part from its anchor position. It is defined as $\varphi_d(dX_{ik}, dY_{ik}) = (dX_{ik}^2, dX_{ik}, dY_{ik}^2, dY_{ik})$ where $k \in \{1,2,3,4\}$ represents the four parts, $i \in \{1,2,\dots,m\}$ with m representing the total number of positive training images [6]. In this equation, $(dX_{ik}, dY_{ik}) = A_k - \left((X_{ik}, Y_{ik}) - (X_i, Y_i)\right)/ppc_i$ where (X_i, Y_i) is the upper left corner of the face bounding box in the image I_i, (X_{ik}, Y_{ik}) is the upper left corner of part k bounding box in image I_i, ppc_i is the number of pixels per cell in Image I_i, and A_k is the anchor position of part k measured in cells with reference to the root mesh. The term $\left((X_{ik}, Y_{ik}) - (X_i, Y_i)\right)/ppc_i$ gives the position of part k measured in cells with reference to the root mesh which when subtracted from A_k gives the difference in cells between the actual position of the part and its anchor position.

Figure 3.11 illustrates these deformation features with several examples where the yellow shading indicate the anchor position of each part and the red box indicate the annotation bounding box of the part. The red arrow indicate the deformation of each part from its anchor position. It is clear from the figure that one cell shifting to the left results in a deformation feature vector of (1,1,0,0), while one cell shifting to the right corresponds to (1,-1,0,0). Also, one cell shifting up results in a deformation feature vector of (0,0,1,1), while one cell shifting down corresponds to (0,0,1,-1). It is also clear that the quadratic term is always positive to ensure the deformation penalty and exhibit a quadratic growth that lead to very large penalties for large deformations.

The scoring function for each training sample can be written as:

$$S_i = F_0 \,.\, \varphi_a(X_i, Y_i) + \sum_{k=1}^{4}\left[F_k \,.\, \varphi_a(X_{ik}, Y_{ik}) - d_k . \varphi_d(dX_{ik}, dY_{ik})\right] + b, \qquad (3.6)$$

where F_0 is the root filter which is of size 5x5x31 to match $\varphi_a(X_i, Y_i)$ the appearance feature (HOG) of the root. F_k for $k \in \{1,2,3,4\}$ are the part filters which are of sizes 4x4x31 for each of the two eyes and the nose, and of size 3x6x31 for the mouth again to match $\varphi_a(X_{ik}, Y_{ik})$ the appearance features (HOG) of the parts. d_k for $k \in \{1,2,3,4\}$ are vectors of size 4 that define the deformation cost of each part relative to the corresponding deformation feature vector $\varphi_d(dY_{ik}, dY_{ik})$. Finally, b is a bias term used to shift the scores such that most of the positive training samples lead to a positive score and most of the negative samples lead to a negative score.

Equation (3.6) can be written in a vector form as:

$$S_i = \omega . V_i, \qquad (3.7)$$

where ω is the model parameters defined as:

$$\omega = (F_0, F_1, \dots, F_4, d_1, \dots, d_4, b), \qquad (3.8)$$

and V_i is the concatenated feature vector that represents each training sample by combining the appearance and deformation features extracted from it which can be defined as:

$$V_i = (\varphi_a(X_i, Y_i), \varphi_a(X_{i1}, Y_{i1}), \dots, \varphi_a(X_{i4}, Y_{i4}), \varphi_d(dX_{i1}, dY_{i1}), \dots, \varphi_d(dX_{i4}, dY_{i4}), 1) \quad (3.9)$$

The target of the training process is to obtain the optimum value of the vector ω that would lead to a scoring function capable of discriminating between the positive and the negative training samples with a maximum margin. This problem is solved using the support vector machines that was explained in details in Chapter 2.

The positive samples used in the training were explained in details, now the negative samples on the other hand were extracted from 2000 images collected from the internet that do not contain any faces. A naive way would be to just take random windows from these images. A better way is to use these random windows just for training an initial model that is then used to obtain more hard negatives by running it as a detector over the negative images and obtain only the samples that pass a very low threshold. These samples can now be used to train a better model.

3.4.3 Adjusting the bias

An important post training step for the SPM is to adjust the bias term of the model for the total score, the root score, and each of the different part scores such that each of them tends to produce a positive score for the positive training samples and a negative score for the negative samples. This is important because it will allow focusing on the partial occlusion problem which should result in negative scores for the occluded parts. These bias terms facilitate the selection procedure of the model by providing common basis for

comparing part subtypes and for comparing facial parts with common occluding objects such as sunglasses, caps and hands.

The following algorithm is used to find the optimum bias term for the total score, the root score and each of the four parts scores. Recall that the model is trained using equation 3.6 which calculates the total score for each training sample using a single bias term. A different bias term for each part of the score could not be used at the training stage because all the terms are added up in one equation. After all the model parameters are calculated through the training, the algorithm of finding a separate bias term for the different parts of the score starts with recalculating the different parts of the total score for each training sample separately as follows:

$$S_{r_i} = F_0 \,.\, \varphi_a(X_i, Y_i) \tag{3.10}$$

$$S_{P_{ik}} = F_k \,.\, \varphi_a(X_{ik}, Y_{ik}) - d_k . \varphi_d(dX_{ik}, dY_{ik}) \tag{3.11}$$

$$S_{t_i} = S_{r_i} + \sum_{k=1}^{4} S_{P_{ik}} \tag{3.12}$$

where S_{r_i} is the root score for each training sample $i \in \{1,2,\dots,M\}$. $S_{P_{ik}}$ are the part scores with $k \in \{1,2,3,4\}$ for each training sample i. S_{t_i} is the total score without the bias term for each training sample i.

After that, the histogram for the different scores of the positive samples and the negative samples are calculated separately. Figure 3.12 shows the distribution for the scores of the positive training samples in red and the scores of the negative training samples in green for the total score S_{t_i}, the root score S_{r_i} and the four facial parts scores $S_{P_{ik}}$ where $k \in \{1,2,3,4\}$ and $i \in \{1,2,\dots,M_P\}$ for the positive training samples and $i \in \{1,2,\dots,M_N\}$ for the negative training samples. The vertical red line shows the mean of the positive

training samples scores while the green line shows the mean of the negative training samples scores. The black line represents the optimum bias term that should be subtracted from each score to make most of the positive training samples have positive scores and most of the negative training samples have negative scores.

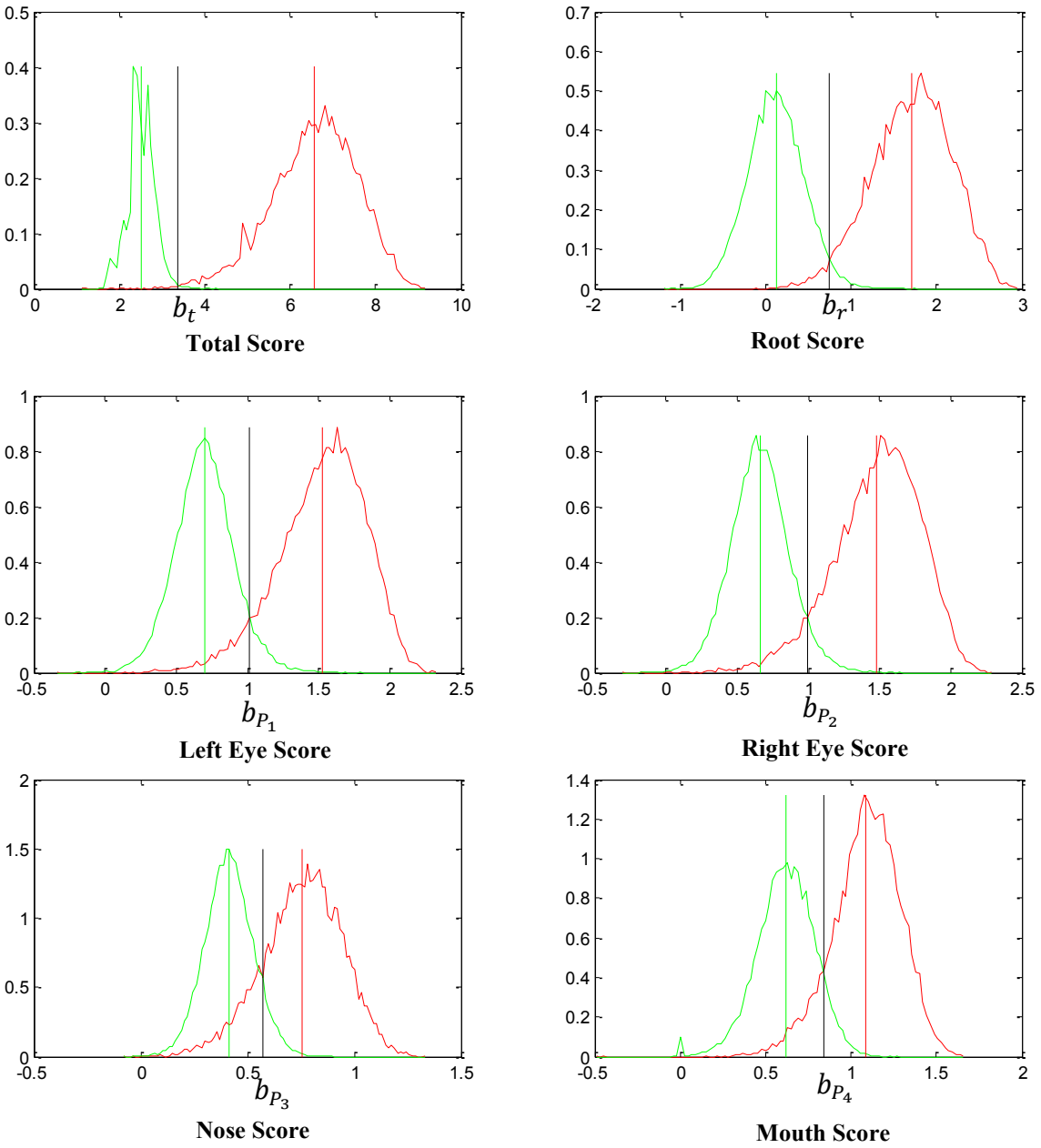


Figure 3.12: The probabilistic distribution for the scores of the positive and negative training samples used to determine the bias term (represented by the vertical black line) that will be subtracted from each score. The red curve represents the positive samples distributions while the green curves represent the negative samples distributions. These curves are for the **frontal** component of the model.

Recall from Baye's theorem that the posterior probability of a training sample to be of class $l = 1$ (positive sample) given its score S=s can be written as:

$$P(l = 1 \mid S = s) = \frac{P(S = s \mid l = 1).P(l = 1)}{P(S = s)},$$

where $P(S = s \mid l = 1)$ is the likelihood probability of a training sample to have a score S=s given it is a positive sample. This is calculated using the normalized histogram of the positive samples scores by simply dividing the range between the minimum score and the maximum score of all the training samples (both negative and positive) into 100 bins, then counting the number of positive training samples at each score bin divided by the width of the bin and the total number of positive samples. The prior probability $P(l = 1)$ of the positive class was set to 0.6 to give more importance to the positive class. Similarly, the posterior probability of a training sample to be of class $l = -1$ (negative sample) given its score S=s can be written as:

$$P(l = -1 \mid S = s) = \frac{P(S = s \mid l = -1).P(l = -1)}{P(S = s)},$$

In both equations, the score S is replaced by the total score S_t, the root score S_r and the four facial parts scores S_{P_k} with $k \in \{1,2,3,4\}$ to produce the 6 graphs in Figure 3.12. The bias term to be subtracted from each score is selected at the intersection point of the two posterior distributions to match the threshold of a Baye's classifier to zero. This gives an insight on "the visibility of a part" based on its score, because if a part is occluded it should now produce a negative score. The equations 3.10 to 3.12 can now be rewritten as

$$S_{r_i} = F_0 \,.\, \varphi_a(X_i, Y_i) - b_r \tag{3.15}$$

$$S_{P_{ik}} = F_k \,.\, \varphi_a(X_{ik}, Y_{ik}) - d_k.\varphi_d(dX_{ik}, dY_{ik}) - b_{P_k} \tag{3.16}$$

$$S_{t_i} = S_{r_i} + \sum_{k=1}^{4} S_{P_{ik}} + (-b_t + b_r + \sum_{k=1}^{4} b_{P_k}) \tag{3.17}$$

where b_r is the root bias, b_{P_k} are the facial parts biases with $k \in \{1,2,3,4\}$, and b_t is the total bias. Note that in equation 3.17, b_t is subtracted while all the other biases are added to remove their embedded effects in S_{r_i} and $S_{P_{ik}}$ hence leaving only the effect of the total bias such that the total score can also be thresholded at zero. This is important because different components of the model are all operated with the same total threshold and need all to be biased correctly. In other words, the total score is the only remaining bias but the other biases are only added locally to add a meaning to their individual scores when occlusion is considered.

It is clear from Fig 3.12 that the best separation of the positive and negative training samples is obtained in the total score because that is the score that was optimized for maximum separation in the SVM training. Also because it has the longest feature vector and hence larger discrimination ability. It can also be seen that the root has better separation than the parts and that the two eyes have better separation than the nose and the mouth. This can be due to the large variability in the appearance of the mouth with different expressions which is difficult to capture with the same model. For the nose, it can be due to the fact that the nose has the lowest amount of details among all the four facial parts because of its simple structure.

Similar analysis is conducted for the profile component of the model. Figure 3.13 shows the distribution for the scores of the positive training samples in red and the scores of the negative training samples in green for the total score S_{t_i}, the root score S_{r_i} and the four facial parts scores $S_{P_{ik}}$ where $k \in \{1,2,3,4\}$ and $i \in \{1,2,\dots,M_P\}$ for the positive training samples and $i \in \{1,2,\dots,M_N\}$ for the negative training samples. Recall that the parts here are the visible eye, the nose, the mouth and the ear. The separation in the parts

of the profile component is not as good as the frontal component due to the larger variability in the appearance of profile parts and due to smaller part sizes leading to smaller feature vectors sizes and hence less discriminative ability.

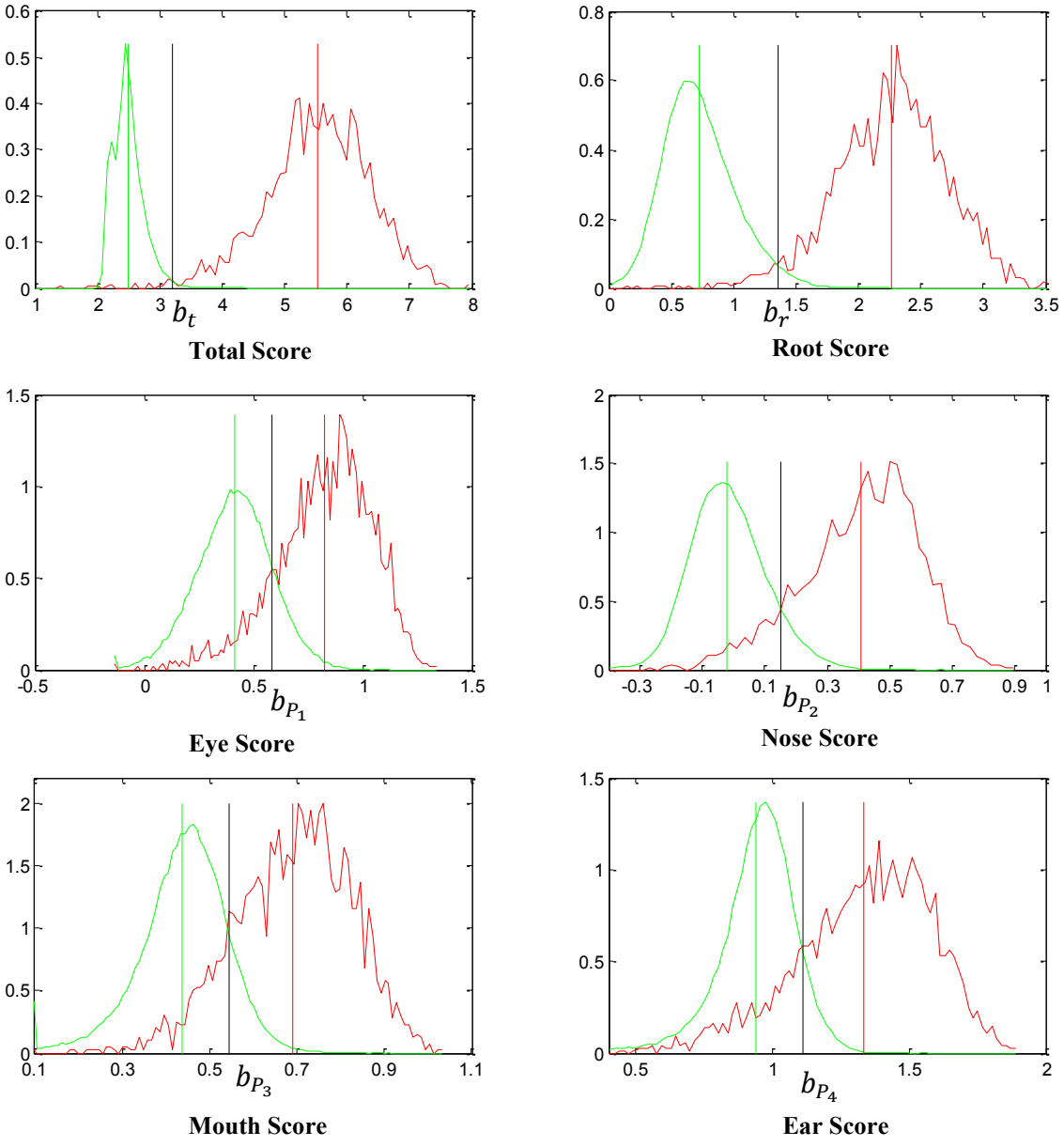


Figure 3.13: The probabilistic distribution for the scores of the positive and negative training samples used to determine the bias term (represented by the vertical black line) that will be subtracted from each score. The red curve represents the positive samples distributions while the green curves represent the negative samples distributions. These curves are for the **profile** component of the model.

The occlusion parts on the other hand are trained separately with only appearance features as stated earlier. This leads to the following much simpler scoring function:

$$S_{P_{ik}} = F_k \,.\, \varphi_a(X_{ik}, Y_{ik}) - b_{P_k}, \tag{3.18}$$

where $k \in \{5,6,7\}$ represents the sunglasses, the caps and the hands respectively. The bias b_{P_k} in each case is obtained in the same way. Figure 3.14 shows the distribution for the scores of the positive training samples in red and the scores of the negative training samples in green for the parts representing the occluding objects $S_{P_{ik}}$.

In the detection, the deformation penalty is added to equation 3.18 to allow for changes in the locations of these occluding objects as follows

$$S_{P_{ik}} = F_k \,.\, \varphi_a(X_{ik}, Y_{ik}) - d_k . \varphi_d(dX_{ik}, dY_{ik}) - b_{P_k} \tag{3.19}$$

where $d_5 = (0.1, 0, 0.1, 0)$ was selected empirically as the deformation coefficients of the sunglasses. Recalling that the deformation feature is defined as $\varphi_d(dX_{ik}, dY_{ik}) = (dX_{ik}^2, dX_{ik}, dY_{ik}^2, dY_{ik})$. The intuition of the d_k values is that a penalty of 0.1 times the square of the deformation of the object from its anchor position in X or Y direction is subtracted from the score. For example, a shift of 3 cells from the anchor position would result in a 0.9 reduction in the score.

Similar reasoning was used to use $d_6 = (0.04, 0, 0.004, -0.05)$ for the cap to give it more freedom to move around its anchor position since it is not attached to a specific place like the sunglasses. Note also that movements in the Y direction are less penalized and moving up is even less by the use of -0.05 as a coefficient for dY_{ik} which is positive if the part moved up as can be seen from the right image of Figure 3.11.

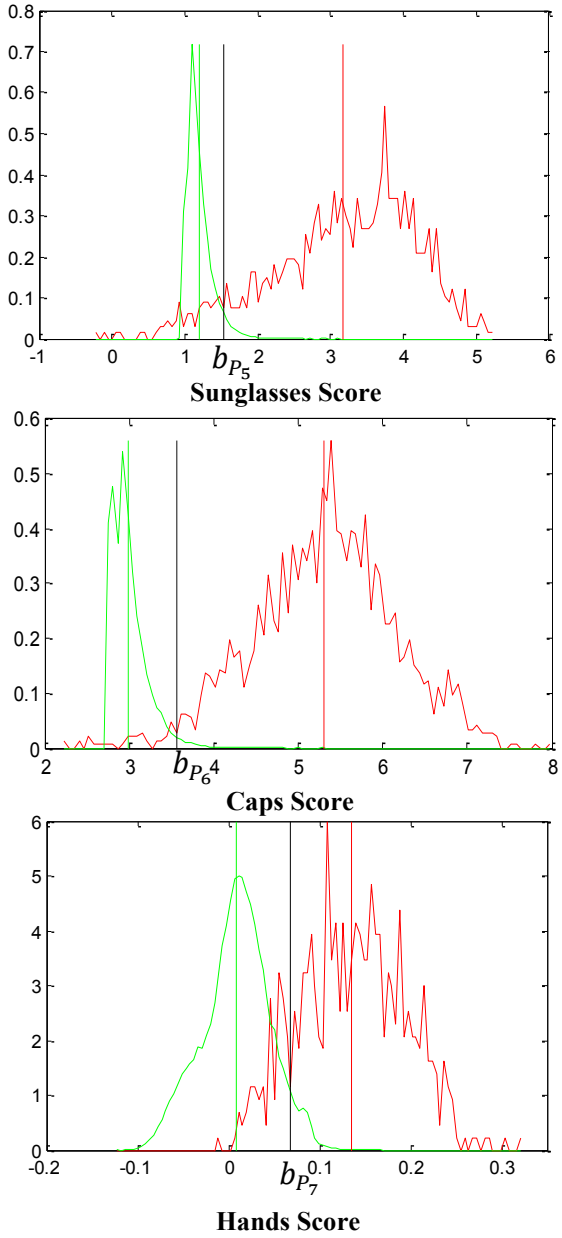


Figure 3.14: The probabilistic distribution for the scores of the positive and negative training samples used to determine the bias term (represented by the vertical black line) that will be subtracted from each score. The red curve represents the positive samples distributions while the green curves represent the negative samples distributions. These curves are for the parts representing the **occluding objects**.

For the hands, $d_7 = (0.001, 0, 0.001, 0)$ to allow moving around the anchor which is selected at the center of the face. Due to the nature of the hand it can be in front of any part in the face.

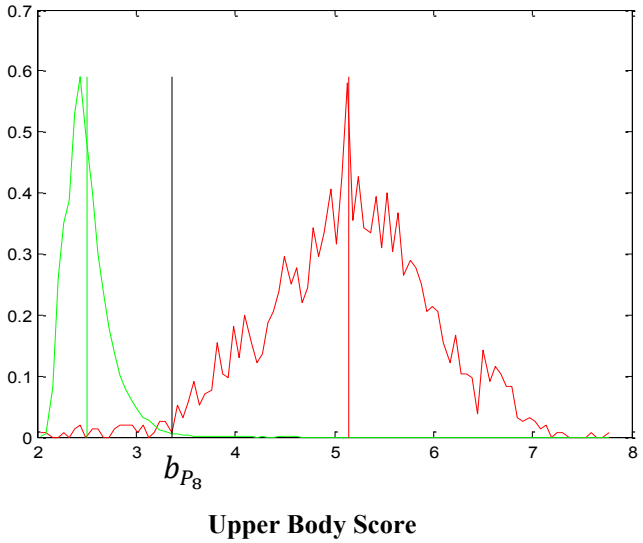


Figure 3.15: The probabilistic distribution for the scores of the positive and negative training samples used to determine the bias term (represented by the vertical black line) that will be subtracted from each score. The red curve represents the positive samples distributions while the green curves represent the negative samples distributions. This curve is for the **upper body** part.

Finally, the upper body part is also trained separately similar to the occlusion parts and it is added to the total score as an optional part to reinforce the total score if its score is high. Because some faces may appear in the image without the shoulders being visible either because they are outside the field of view, or because they are occluded in crowd scenes or even because the face is just a picture in the image with no body. This means that a face should not be penalized if the upper body part score is negative but it should be reinforced if it is positive. The upper body adds to its own face a context information that can help a lot if the face is heavily occluded. It also helps in detecting faces that are partially occluded by the shoulders of another person in a manner similar to faces occluded by other faces explained in Figure 3.4.

3.5 Post Processing

This chapter is concluded with two important post processing steps needed after completing the detection step to produce good results, namely the Non-Maximum

Suppression (NMS) and the false positive reduction using skin. In general, the detection using the sliding window approach results in the same face being detected several times over adjacent windows in the same scale and also over windows on different adjacent scales as will be illustrated in more details in the next chapter. The NMS is usually used to keep from these repeated overlapping detections the one that corresponds to the maximum score and literally suppress the candidates with non-maximum scores.

The algorithm is simple, it just arranges all the detections produced by the detector according to their scores in descending order and then loops through them from the higher score candidates checking its overlap with all the lower candidates, if that overlap exceeds a predefined threshold then the candidate with the small score is suppressed. This operation does not affect non-overlapping candidates, it only affects highly overlapping candidates leaving only the one with the highest score. This process is critical in the crowded scenes where different faces can be overlapping as shown in Figure 3.4. The threshold of the overlap should be selected carefully so as not to remove these different faces.

Another important post processing step is reducing the false positives using the skin information as in the RSFFD method [27] explained at the end of the previous chapter. The idea simply is that the HOG features used in the SPM is based on the edge information, the skin on the other hand is based on the color which carries complimentary information to the HOG. Many false positives that resemble the edges of the face and its facial parts will easily be removed because simply they are completely different in color from the skin. The problem is some images are gray scale and carry no color information, and some images are colored but have grey scale picture of a face inside it, also some faces have a skin color

that is not captured by the skin model. To avoid these problems the skin color is only checked if:

1. The image has 3 channels meaning it is not gray scale with single channel and no color information.
2. The image's 3 channels are not identical meaning that it is not a gray scale image that was just saved using 3 equal channels but still has no color information.
3. The total score is less than 1.5 meaning that if the score is very high it is most probably a face and the color does not need to be checked to avoid the removal of obvious faces just because their color is not captured by the skin model.

For the candidates that satisfy all of these conditions the skin model explained at the end of the previous chapter is used by counting the skin pixels inside the candidate to decide whether it is a face or not.

3.6 Conclusion

This chapter introduced the core of this dissertation through the Selective Part Model that explicitly focus on the partial occlusion problem in face detection. It models the face as a collection of parts that takes into account the regular facial parts and optional facial accessories and body parts that might occlude the face in real life settings. The SPM has a lot of potential for partial face detection. The execution time is an important factor of the detector performance especially when the input is a video. The next chapter discusses in details the detection time analysis and suggests how it can be accelerated for real time performance.

CHAPTER 4

VIDEO FACE DETECTION

This chapter completes the discussion of the previous chapter with special focus on the time aspect of the detector. Generally speaking, the detection time of any face detector is a very important factor because face detection is usually just the first step in another facial analysis pipeline, so it needs to be fast even if the input is a single image. Particularly in videos, the detection time is crucial because in addition to the previous reason, the performance needs to be optimized to detect at least one frame per second to maintain the real time response for video face detection. To achieve that, part of the discussion will be valid for fast face detection in general whether from single images or frames of a video, and another part will be only suitable for videos because it utilizes the temporal component of the video.

The chapter starts with a discussion of the face detection problem versus the face tracking problem in videos. Then, a detailed analysis of the SPM detection time performance is provided to give an intuition about the time bottlenecks and how it can be attacked. This is followed by a discussion about the different types of redundancy in the calculations performed in the SPM detector being a sliding window and part based face detector. Then several factors are investigated to achieve real time performance. Finally, a brief analysis for occlusion in videos is provided.

4.1 Detection vs. Tracking

The video face detection problem is closely related to the face tacking problem. A simple settings face tracking problem can be defined as: given a detected single face or several faces in one frame of a video the target is to track these faces in the following frames of the same video. Several factors are important to define the challenges of the tracking problem including, whether the camera is moving or not, whether the same face is going outside the field of view then back in, whether the face is going to be occluded by other objects in the scene including other faces, and even whether all the video is captured using the same camera or using several cameras such as in movie or TV series settings.

For faces in the wild, simply there is no control on the settings which includes all the previous challenges and much more. These challenges makes it difficult for the tracking to just start from a single detection and take it from there. In addition, well performing object detectors in general and face detectors in particular have led to association based tracking approaches, which detect the objects over all frames and then associate corresponding detections of the same object over short sequences of frames into what is called “object tracklets”. These object tracklets are then linked into longer tracks to produce the tracking results of the whole video.

For example, Roth et al. [43] proposed an approach for multi pose face tracking by first detecting the faces over all frames and then linking these detections over two stages using multiple cues. The low level stage produces short tracklets from separate faces detected in consecutive frames based on similarities in location, size, and pose. For two faces detected in consecutive frames to be connected, the similarity in their location, size, and pose need

to exceed a first threshold and also need to exceed the similarities with other detections in the frame by a second threshold. This two threshold strategy results in a set of reliable short tracklets. Then, the high level stage is used to link these tracklets based on three different cues. The face ID cue which compares facial features of the most frontal face views in the tracklets, the appearance cue that uses color features from the face and cloth beneath the face to build an online discriminative classifier, and the constraint cue which encourages natural association in terms of motion and pose compatibility between tracklets.

These linking strategies need to be initialized with face detection over the video frames. For this task they used the modified census transform face detector of [44] trained over 11 yaw angles from -90^{o} to 90^{o} with a step of 15^{o} and 5 roll angle from -45^{o} to 45^{o} with a step of 22.5^{o}. The different combinations of yaw and roll angles resulted in 47 components of the detector. They used a separate eye detector to localize the eyes inside the face to use it in alignment for the face id cue. Their reported average detection time per frame of size 1024x576 was 1.84 seconds. On the other hand the low level association and high level association combined took an average of 25.2 milliseconds. Thus, the detector they used took as much as 98.6% of the total computation time.

Another closely related problem is the face clustering in videos which also takes as an input the detected faces from all the video frames and partition them in disjoint clusters which is similar to face recognition but in unsupervised settings [45]. This problem focuses on linking short tracklets over the whole video where face detections of the same person should be in the same cluster whether they are in consecutive frames or not. The clustering and linking of short tracklets were addressed simultaneously in [46] to link face detections of the same person over long videos. They used the Viola Jones face detector to obtain the

detections in all frames and reported in their conclusion that the performance of face clustering and tracklet linking can benefit from using more sophisticated face detection methods. In this work, the focus is only on producing a robust face detector that can be applied to all the frames of a video, then using the more sophisticated output of the SPM that include information about the parts and their possible occlusions other methods such as [43] and [46] can be used to link the different detections or cluster them.

4.2 Detection Time Analysis

In this section, the performance of the basic SPM with only four facial parts and three pose components is analyzed over a standard image to give an intuition about the bottlenecks of face detection and how it can be attacked. The detection in an image starts with constructing an image pyramid to allow for detecting faces of different sizes using the same model designed to detect faces with fixed size. For example in the SPM, The root filter size of the model determines the face size that can be detected. If a root filter of size 5x5 cells is used with cell size of 8x8 pixels, then the model can only detect faces of sizes around 40x40 pixels. This determines the minimum size of faces that can be detected with the model. However to detect larger faces, the image is down sized over different scales to construct an image pyramid as shown in Fig 4.1. A face of size larger than 40x40 will be reduced when the image is down scaled until in one level of the pyramid it will be close to 40x40 and that is where it will be detected.

In Fig 4.1, the image is of size 480x640 and has two faces one of them is around 150x150 and the other is around 50x50 pixels. The image is scaled with 5 levels per octave. This means that the number of levels of the pyramid between any scale and half of the same

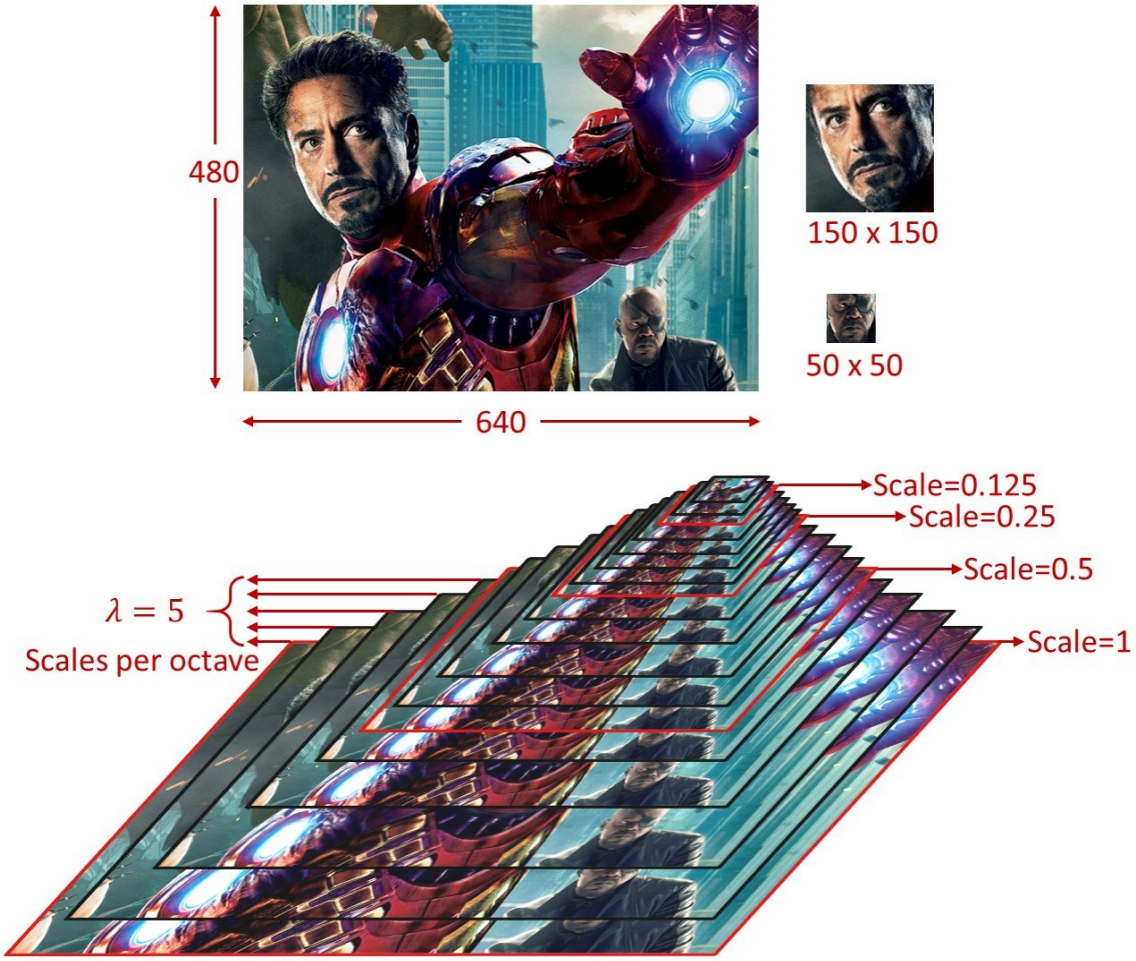


Figure 4.1: Example of a test image with 2 faces of different sizes and their image pyramid of 18 levels where the first level in each octave is high-lighted with a red frame.

scale is 5. For example, in Fig 4.1 the original image is of scale 1 and lies in the base of the pyramid or the first level which is of size 480x640. The sixth level of the pyramid will have half the size namely 240x320 and then the eleventh level will be halved again to 120x160 and so on. Table 4.1 shows a detailed analysis of that pyramid in Figure 4.1. The first column is the level index where level 1 is the bottom of the pyramid and level 18 is the top. The second column shows the down scaling factor used in this level starting from 1 to 0.0947. This leads to image sizes from the original image of size 480x640 to the top of the pyramid of size 46x61 as shown in the third column. From each image in the pyramid a

HOG feature map is extracted where for example in level 1 it is of size 58x78x31. The 58 cells height correspond to 58x8=464 pixels height which is reduced from the original image by at most 1 cell from each side. Similarly the 78 cells width corresponds to 78x8=624 pixels and the 31 is the HOG feature extracted from each cell. These feature maps are padded by 4 cells from each side to allow for detecting objects partially outside the field of view of the camera. Reflective padding was used so that if half the face is in the border of the image, the padding will complete the rest with symmetry which worked well for frontal faces because of symmetry. The root filter with size 5x5x31 is convolved over the

Table 4.1. Image pyramid analysis

Level	Scale	Image Size	Feature Map Size	Score Matrix	Face Size
1	1.0000	480x640	58x78x31	62x82	40.0000
2	0.8706	418x558	50x68x31	54x72	45.9479
3	0.7579	364x486	44x59x31	48x63	52.7803
4	0.6598	317x423	38x51x31	42x55	60.6287
5	0.5743	276x368	33x44x31	37x48	69.6440
6	0.5000	240x320	28x38x31	32x42	80.0000
7	0.4353	209x279	24x33x31	28x37	91.8959
8	0.3789	182x243	21x28x31	25x32	105.5606
9	0.3299	159x212	18x25x31	22x29	121.2573
10	0.2872	138x184	15x21x31	19x25	139.2881
11	0.2500	120x160	13x18x31	17x22	160.0000
12	0.2176	105x140	11x16x31	15x20	183.7917
13	0.1895	91x122	9x13x31	13x17	211.1213
14	0.1649	80x106	8x11x31	12x15	242.5147
15	0.1436	69x92	7x10x31	11x14	278.5762
16	0.1250	60x80	6x8x31	10x12	320.0000
17	0.1088	53x70	5x7x31	9x11	367.5835
18	0.0947	46x61	4x6x31	8x10	422.2425

feature map moving one cell at a time to produce one number of the score matrix at each location of the feature map. This leads to a score matrix of size 62x82 for the first level.

The part filters on the other hand are evaluated at double the scale of the root filter leading to a score matrix of double the size which is then down sampled to exactly match the root score matrix. The interpretation of 62x82 score matrix is very important because it reflects the sliding window approach used for detection. Each entry in the score matrix corresponds to one window in the sliding window approach. This means that in the first level of the pyramid which is of size 480x640 pixels there is a sliding window of size 40x40 pixels and a step 8 pixels which results in 62x82=5084 different windows to be checked if it contains a face or not based on its score. This means 21,903 windows over the whole pyramid. For each window, the score equation is evaluated as explained in the previous chapter in equations 3.1 to 3.5.

Finally, the last column explains a very interesting property of the idea of using an image pyramid with fixed model size. The face size that can be detected in the first level is of size around 40x40 which is the smallest size that can be detected. The second level scale the image down by a factor of 0.8706 to produce an image of size 418x558. The faces detected on this level are also of size 40x40 because the model size is fixed but these detection boxes are then returned to the original image size by dividing 40 over 0.8706 resulting in 45.9479 which corresponds to faces in the original image that are around 46x46 pixels. The last column of the table shows the actual face size in the original image that will be of size 40x40 in this level of the image. This means that for the faces in Figure4.1, the 50x50 face can be detected between the second and third levels while the 150x150 face can be detected between the tenth and eleventh levels of the pyramid. It is worth mentioning

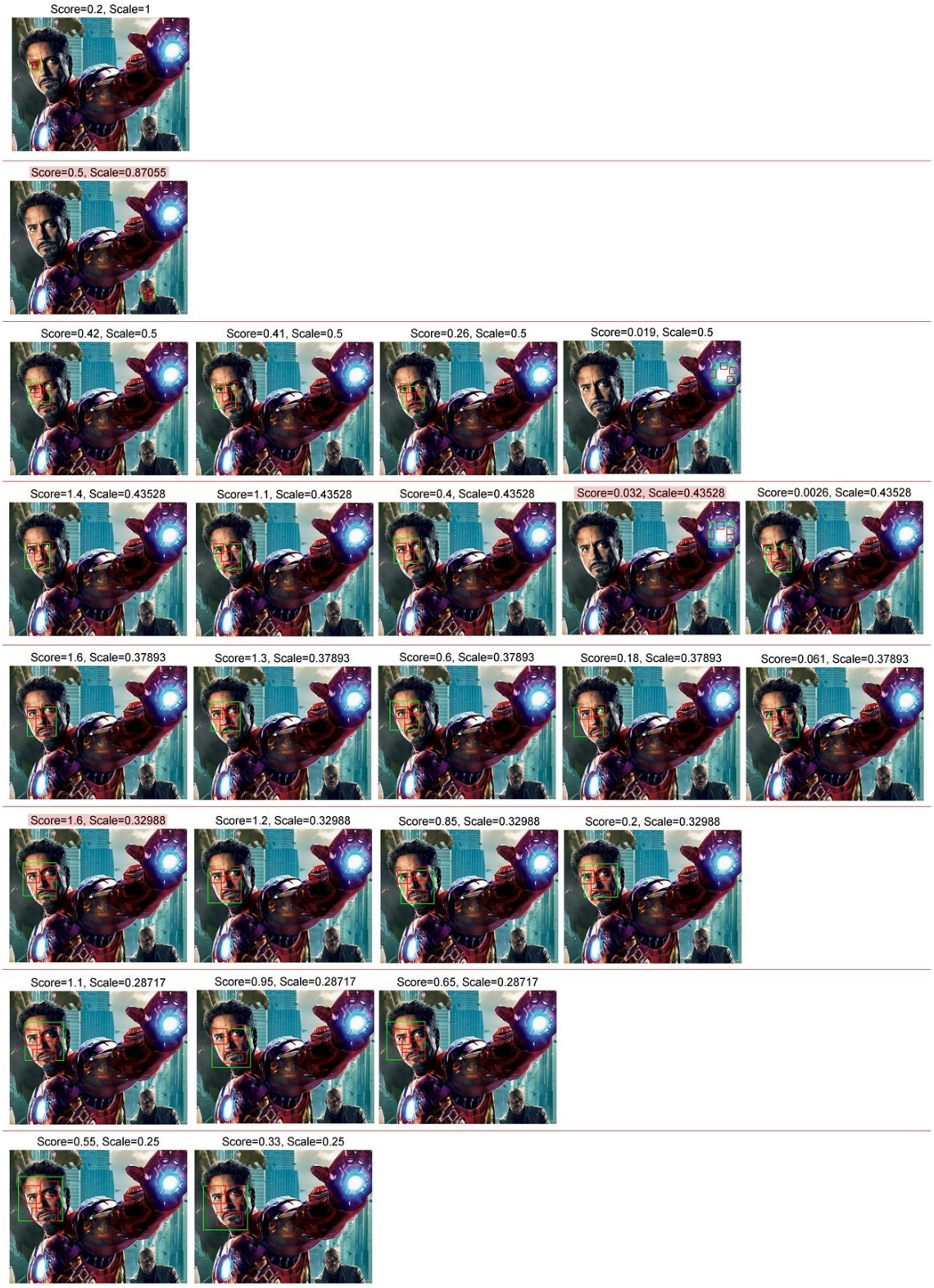


Figure 4.2: Detections over the image pyramid levels. The image size is 480x640 and the image pyramid has 18 levels by resizing the image with scale factors from 1 to 0.095. From the 18 levels only 8 produced detections. The detections in each of these 8 levels are each shown in a separate row and arranged according to score. From the 25 candidates, the NMS will keep only 3 non-overlapping candidates over all scores. The score of these three candidates are highlighted in red.

that these sizes are approximate because it can change based on how tight the detector will be detecting the face as illustrated in Fig 4.2.

Figure 4.2 takes the extra mile of showing the details of the face candidates detected in each level of the pyramid. Only 8 levels of the pyramid produced candidates that exceeded the threshold of the total score. Each of these levels is shown in a separate row with each face candidate shown over a separate image and arranged according to score. The levels of the pyramid that did not produce any detections are omitted. It can be seen that there are three main candidates, the first one is in the second row from the top corresponding to the small face, the second one corresponds to the large face and is detected over 6 different consecutive levels of the pyramid, and the third one corresponds to a false positive (in the hand of iron man). By investigating the detections of the large face, where the green rectangle is the face bounding box and the red rectangles are the facial parts bounding boxes, it can be seen that the same face can be detected several times with different bounding box sizes over different levels of the pyramid and it also can be detected several times within the same level in horizontally and vertically adjacent windows.

In this example the large face was detected 20 times with different scores ranging from 0.0026 up to 1.6. This illustrates two types of redundancy, the first one corresponds to detecting the same face several times in the same level of the pyramid because the face can appear in adjacent windows in the sliding window approach. The second redundancy corresponds to detecting the same face several times over different scales because the face can be considered of different sizes based on how tight the bounding box is used. This suggests that the search space can be reduced effectively by removing these redundancies. For example by starting the detection from the top of the pyramid, then for a specific face

location if its score exceeded a high threshold say 1 in this case which happens in the second row from the bottom then this location is excluded from the calculations of the lower levels of the pyramid. This will save the unnecessary calculations that detected the same face 15 extra times. More details about these redundancies will be revisited in the next section.

The remaining question is how to find the number of levels to use in each image to construct the pyramid. The process starts by deciding λ that corresponds to how many levels per octave to be used (this is a design parameter that is typically taken in the literature to be from 5 to 10). Then finding out how many times the minimum size of the image can be divided by two and still remains greater than the root filter size. This number and the divisions by two correspond to the number of octaves that can be used before the image become too small to be compared with the root filter. For example, in an image of size 480x640 with root filter of size 40x40 pixel the question becomes what is p such that $480/2^p = 40$, rearranging this means $p = log_2(480/40) = 3.58$ octaves. Considering 5 levels per octave this means $3.58 \times 5 \approx 18$. In general, the number of levels in the pyramid can be found using the following equation:

$$L = ceil\left(\frac{\ln(\min Image\ size/root\ filter\ size}{\ln 2^{1/\lambda}}\right) \tag{3.20}$$

where $log_2 x$ is just replaced by $lnx/ln2$ and then λ is raised to the power of 2 inside the ln function. This factor $2^{1/\lambda}$ is the scaling factor that relates each level with its adjacent level. In this example $1/2^{1/5} = 0.8705$ which is clear in Table 4.1 where each scale is obtained from the above scale by multiplying it by 0.8705.

Also, it is worth mentioning that the parts are detected at double of the resolution of the root filter to reflect more details. The first 5 levels in the pyramid have no levels corresponding to double the resolution so either the image is scaled to twice the resolution or as suggested in [6], the first five levels are used to construct feature maps first at 8pixels per cell and then at 4 pixels per cell which results in double the resolution. This approximation leads to faster feature extraction performance and is adopted in this work.

Finally, the detection algorithm can be divided into six parts with respect to time analysis. The feature extraction of the whole pyramid which takes 0.26 seconds. The convolution of all the root and facial parts filters over all the levels of the pyramid which takes 0.3 seconds. The generalized distance transform that accounts for picking the maximum part score over all possible deformations which takes 0.2 seconds. The selection of the windows with scores exceeding the threshold and finding their respective bounding boxes with respect to the original image which takes 0.01 seconds. The post processing steps of the Non-Maximum Suppression (NMS) to combine the overlapping candidates into single detections which takes negligible time of 0.0001 seconds and the post processing of rejecting false positives using skin which takes 0.01 seconds. The total time is 0.78 seconds for detecting the two faces in the 480x640 image of Figure 4.1 with 5 levels per octave and using only the three pose components with only four facial parts per component. These times are measured on a machine with a 3.2 GHz Intel Xeon processor.

The effect of the **image size** can be seen if the image is doubled in size and became 960x1280 which is similar to concatenating 4 images of size 480x640, then the total detection time becomes 2.7 seconds. Now, the image pyramid contains 23 levels and 83,615 window instead of 18 levels and 21,903 windows. The feature pyramid takes 1

second to construct, the score of all the windows including both the convolution and the deformation takes 1.7 seconds. The NMS takes 0.0002 and the skin takes 0.06 seconds where the detector produces 36 candidates corresponding to 4 different candidates of which two false positives are rejected using skin.

Similarly, the effect of using 10 **levels per octave** instead of 5 can be seen in raising the total detection time to 1.46 seconds over the image of size 480x640. Here, the image pyramid contains 36 levels and 41,114 window instead of 18 levels and 21,903 windows. The feature pyramid takes 0.49 second to construct, the score of all the windows including both the convolution and the deformation takes 0.97 seconds. The NMS takes 0.0002 and the skin takes 0.02 seconds where the detector produces 43 candidates.

Adding **more parts** to the model increases the time of calculating the score of all windows. For example, adding the Sunglasses which is of size 4x10 cells adds 0.1 seconds for the additional convolution and deformations over all the windows. On the other hand, adding the cap which has a larger size of 6x10 cells adds 0.13 seconds. Similarly adding the hands, or part subtypes will similarly add up to the time of calculating the score of all windows which is redundant for the easy negative windows as will be discussed in the next section. Finally, adding all the previous factors (image size, number of levels per octave, and number of parts) together can significantly increase the detection time.

4.3 Computational redundancy

The target of this section is to explore how face detection can be accelerated by reducing the redundancy in its computations. There are several types of redundancy in the face detection computations resulting from several factors including: the sliding window

approach over an image pyramid, the part based approach within each window, and even the calculations of the feature map pyramid itself.

The sliding window over different scales in an image pyramid brute forces through all the possible scale, spatial, and temporal locations of faces resulting in the following three layers of redundancy. The first one is over adjacent windows in different scales of the image pyramid, where the same region of the image is analyzed several times over adjacent scales searching for faces with different sizes. The second layer is in spatially adjacent windows over the same scale, where the common part of these windows is analyzed several times searching for faces in different spatial locations. The previous two types where illustrated in the previous section and are valid whether the input is a single image or several frames from a video, on the other hand if the input is a video there is a third layer of redundancy in the calculations of adjacent windows over the temporal component.

From a different perspective, in part based detectors there is another type of redundancy in the calculations within each window due to the structure of the model as root and several parts. For each window several scores of root and parts are calculated to decide whether it is a face or not. While this is important in the windows containing faces and face-like objects, in many other windows containing completely face-unlike objects only the root score can be enough for a clear negative decision. This means that there is redundancy in the number of scores from root and parts used to take that negative decision. This factor is very important due to the unbalanced nature of the number of negative and positive windows in an image where the positive windows are usually very few in number compared to the negative windows.

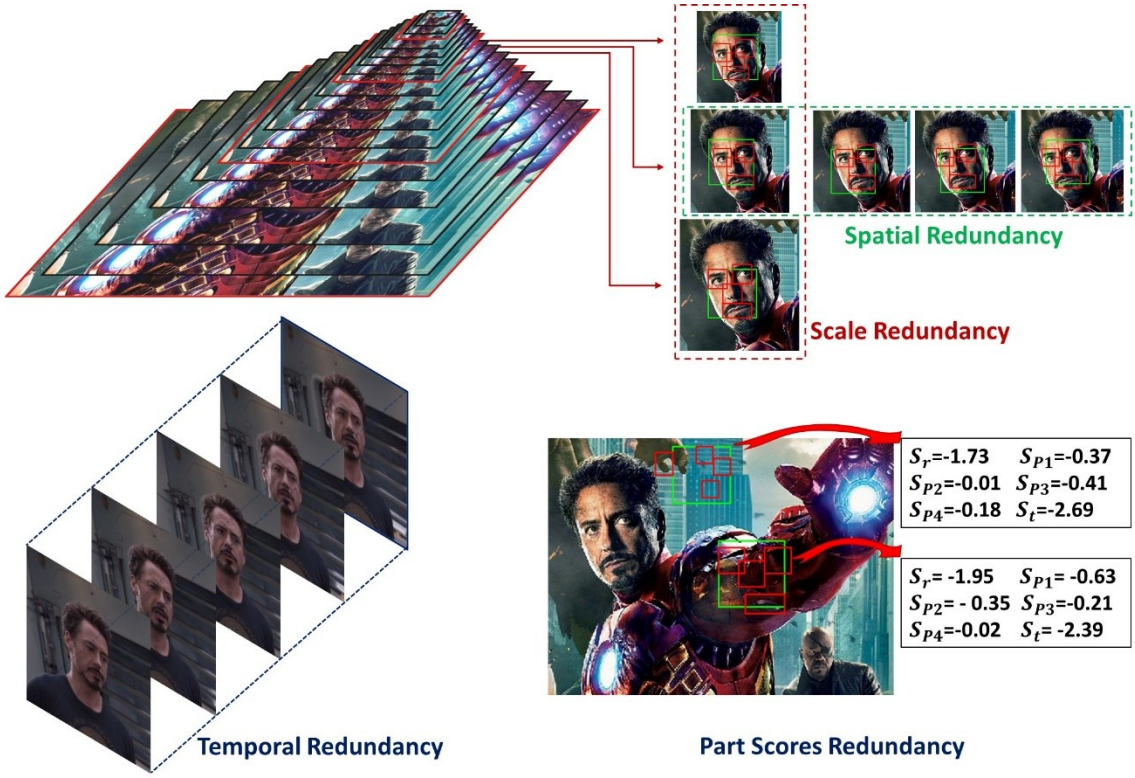


Figure 4.3: Different types of computational redundancy in part based models that uses sliding windows over an image pyramid. The temporal redundancy is only added if the input is a video.

Figure 4.3 pictorially illustrates these different types of computational redundancies found in any sliding window detector that is based on parts. The scale redundancy shows how the same face is detected over three adjacent scales where the bounding box is very tight at lower levels and larger at higher levels. The spatial redundancy on the other hands shows over the same scale that adjacent windows can still contain the same face where the bounding box location with respect to the face is slightly shifted horizontally or vertically. The part scores redundancy shows two examples of negative windows that have very large negative scores for the root filter which is enough to decide they are negative windows and adding the rest of the parts scores just add unnecessary additional computations. Finally, the temporal redundancy is only if the input is video and shows that in many consecutive frames the positive and negative windows may not change much, note how the actor just

changed his pose slightly while he is speaking but his location in consecutive frames remains close unless the camera is switched to a different view.

In addition, there are also some hidden redundancies in calculating the feature map pyramid itself. For the HOG features as explained in the previous chapter, the gradient of each pixel is discretized into different orientation partitions. The gradient magnitude of each pixel is then added to the corresponding orientation bins in four cells around it with bilinear interpolation weight. The gradient itself is a simple difference operation in the x and y direction, the bottleneck is in finding the magnitude and orientation corresponding to this gradient.

Yan et al. [47] suggested the use of a Look Up Table (LUT) to generate exactly the same HOG features. This can be analyzed as a method for tackling the redundancy of the HOG calculations by pre-calculating all possible values needed in HOG and storing them in arrays replacing runtime computations with simpler and more efficient array indexing operations. Since pixel values are in the range of 0 to 255, the gradient in x and y directions can only take values from -255 to 255. All the possible combinations of differences can be stored in a table of size 511x511 with their corresponding magnitude and orientation partitions. Matrix index operations can replace the much harder magnitude and orientation calculations needed in HOG.

Felzenszwalb et al. [48] suggested the use of cascades of models of ascending complexities similar to the idea utilized in the Viola Jones framework explained in Chapter 2 to accelerate the DPM object detection. For a model with n+1 part (including the root) they obtained a sequence of n+1 models where the first model starts only with the root and

then each model adds 1 part to the previous model in the sequence. This means that windows that are clearly negative would be classified only by the first and the simplest model that contains only the root. On the other hand, harder windows which are fewer in numbers will keep evaluating more parts until a good decision is made.

4.4 The Multi-Layer SPM

The cascade idea is a natural match to the selective part models explained in the previous chapter. Because the cascade can be seen as a consecutive selection procedure that computes only the necessary response from the root and parts filters to achieve fast negative decisions for easy negative windows. The complete computations of all parts are executed only for the confusing windows. This means that using a large pool of parts (facial parts subtypes, sunglasses, caps, hands, and upper body) to select from in the selection procedure of the SPM can still be efficient because it needs only to be computed at a small number of windows and not over the whole pyramid.

In this section, the Multi-Layer SPM (ML-SPM) is introduced as a way of augmenting the cascade idea into the SPM framework by doing the part selection of the SPM over several layers. The first layer starts with only the root filter with a very low threshold that passes all positive windows and hard negative windows but successfully classify the easy negative windows as negatives before evaluating any parts. This reduces the number of windows the parts will be calculated on enormously. The second layer applies only the basic four facial features on the remaining windows with a low threshold that again tries to pass all the positive windows and few number of very hard negatives. The third layer applies the Selective Part Model as was explained in the previous chapter but only on the

few remaining windows which mean that the optional parts whether for occluding objects or for part subtypes are only applied on a small number of windows.

This adds a new dimension to the power of the selection idea of the SPM. In addition to the selection between part subtypes and between facial parts and possible occlusion objects, it also selects what parts needed to be evaluated in each window. This attacks the part scores redundancy explained in Figure 4.3 by focusing mainly on the redundancy in the calculations of the negative windows through the use of low thresholds that classify different negative windows with only the minimum needed operations.

On the other hand, the detection of the same positive face many times over adjacent positive windows results in redundancy in the positive windows. This can be attacked by starting from the top of the pyramid and the use of a high threshold that if reached the window is classified as a strong positive and then all the adjacent windows spatially or over the remaining scales which basically defines the same location are not evaluated any more. In Figure 4.2, this means that instead of detecting the same face 21 times it will only be detected few times until it reaches the high threshold and then the same location will not be evaluated any more.

The question now, is how to find these low and high thresholds that can be used to reduce the computational redundancy of the SPM. The answer comes from deeper analysis to the scores of the root and the different parts over the positive and negative windows of a validation dataset that is used to gain deeper understanding for the distribution of these scores. This seems similar to the analysis that was conducted on the training data to distribute the bias in Figure 3.12-3.15. However the difference is: the previous analysis

was only done on the training data which has no redundancy, each sample represents a new positive or negative data, while here it will be conducted over all the windows of test images which contain a lot of redundancy.

Table 4.1 explained how an image of size 480x640 was tested over a pyramid of 18 levels each level resulted in a score matrix that reflects the number of root windows evaluated in this level. The target now is to find the root score of all the positive windows (corresponding to faces) separated from all the negative windows (corresponding to background). Any window that when transformed to the original image has a 50% (overlap over union ratio) with one of the ground truth faces is considered a positive window and the rest are considered negative windows. For example, the image of Figure 4.1 has 21,903 windows over all scales as explained earlier, with three root filters (corresponding to the frontal, right profile, and left profile) applied to each window, this leads to a total of 65,709 window. These windows are found to be 65,571 (99.79%) negative windows and 138 (0.21%) positive windows. 66 of the 138 positive windows correspond to the large face and 72 correspond to the small face. Again, these are the windows over different scales and spatial locations that when transformed to the original image match the 50% overlap over union area with one of the two ground truth bounding boxes.

The left image of Figure 4.4 shows all the 138 positive windows (transformed to the original image size) as green boxes. It also highlights from them the two windows with the highest root scores for each face as blue boxes. These two windows correspond to a root score of 0.5 and 0.8 for the large face and 0.3 and 0.4 for the small face. The ground truth of the two faces are shown as red boxes. On the other hand, the right curve of the figure shows the distribution of the root scores for all the 65,571 negative windows. The negative

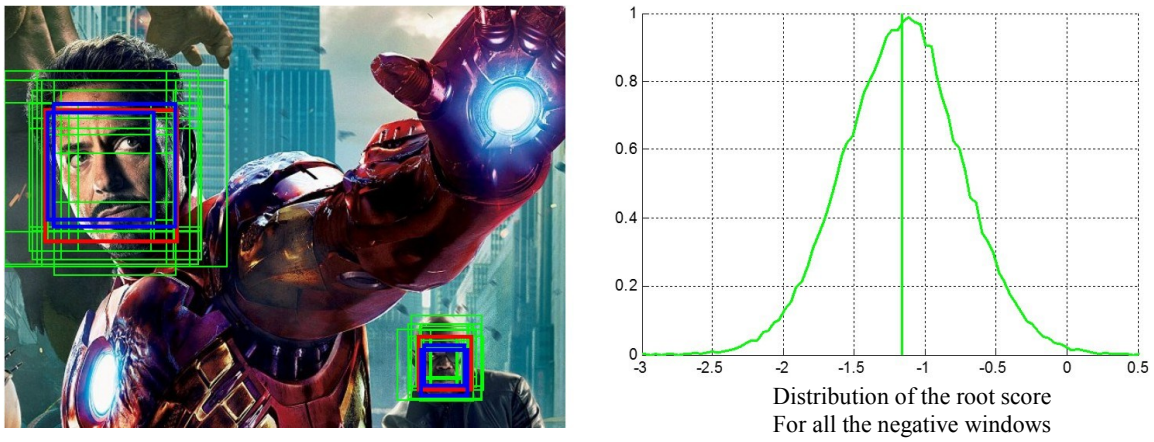
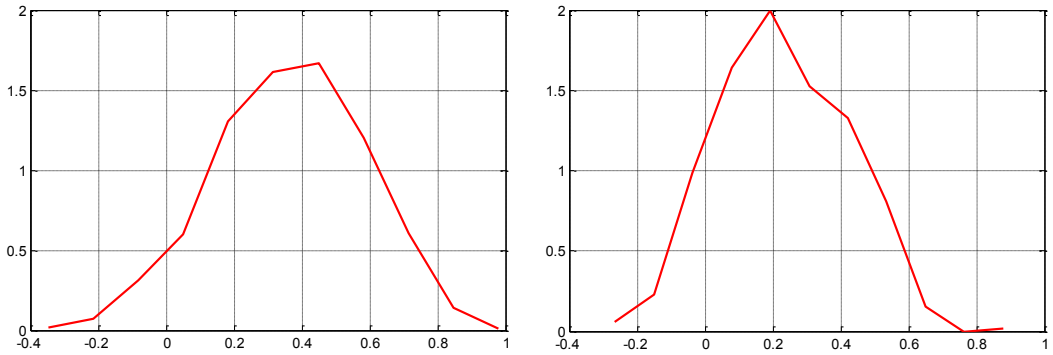


Figure 4.4: The left image shows all the positive windows over all scales in green, the ground truth annotation in red, and the two positive windows with the highest root score for each face in blue. The right curve shows the distribution of the root score for all the 65,571 negative windows over all scales.

windows root scores lie between -3 and 0.5 with a mean of -1.2. To clarify the point, there are 61,876 negative windows corresponding to a root score that is less than -0.5; that is 94.4% of all the negative windows can be classified as negative if only the root filter is used with a -0.5 threshold. It is worth mentioning that 91 windows from the 138 positive windows also have a root score that is less than -0.5 and will also be classified as negative windows which is good because it will reduce the redundancy in detecting the same face many times.

This example gives deep insight about the benefits of this idea, but one image with two faces is not enough to decide the value of the low threshold applied on the root score. To find a good value for that threshold, the detector is evaluated over a validation set of 2000 annotated images and then the root scores of the windows that matched the ground truth face locations are recorded for both the frontal and profile components (both profile left and right components were grouped together). Figure 4.5 shows the distribution of the root scores for these positive windows. The low threshold applied for the root filter is selected such that it will pass all of the faces with a safety margin to account for unseen conditions.



Distribution of the root score for the **frontal** component

Distribution of the root score for the **profile** component

Figure 4.5: The distribution of the root score for the successful detections of the frontal and profile components.

This suggests that a threshold value of -0.5 for the root filter score will be safe enough to pass all the positive windows and reject many false negative windows. To get a feeling of how that will be used, Figure 4.6 shows the binary masks resulting from the image of Figure4.4 after applying a threshold of -0.5 over the root filter score. Each scale in the pyramid will have three masks corresponding to the frontal and two profile components. Only the frontal masks are shown in this figure. All the black cells correspond to windows that were classified as negative using only the root score. The remaining white cells correspond to windows that still need to be classified. Each window is represented by its center cell which makes these windows overlapping with step one cell.

These masks correspond to the root filter resolution which is half the resolution used for the part filters. This means that these masks need to be resized to be used in deciding where the part filter should be computed. It is also important to realize the effect of the deformation property for the parts. This means that any location still need to be classified in the root resolution correspond to 4 locations in part resolution and then a 3x3 cells padding is also added around it to account for possible deformation. Only the four main facial parts are evaluated in the second layer, and the same procedure used with the root in

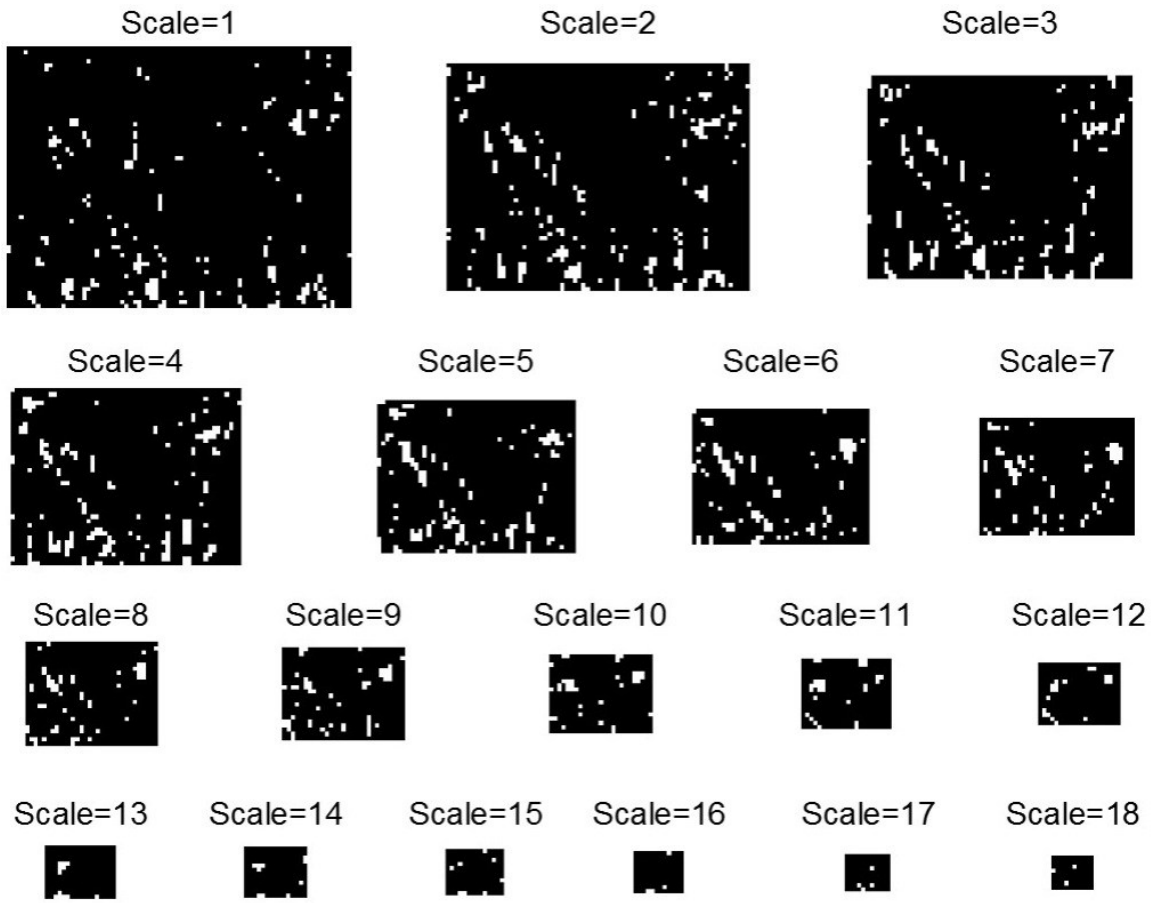


Figure 4.6: The binary masks at each scale resulting from classifying easy negative windows using only the root score (corresponding only to the frontal component of the model).

the first layer is followed to reject more negative windows. The remaining windows in third layer are finally subject to all the available parts including the sunglasses, caps, hands, upper body and the optional part subtypes as explained in the previous chapter.

For videos, similar binary masks can be used to limit the search space based on constraints from the spatial location and scale of consecutive frames. A full detection over the whole frame should be conducted every T frames to account for new faces entering the scene. The parameter T is decided based on the video settings. For example in a video that is captured by a single camera, T can be chosen to correspond for several seconds. While in movie like videos that can be captured by more than one camera T should not exceed

the number of frames corresponding to one second because the whole scene can be changing if the camera changed. In general, tracking techniques should be utilized to work hand in hand with the detection when videos are considered. The task of the detection aims to provide fast separate detections over the frames and the tracking task is to connect them.

4.5 Analyzing occlusion in videos

This work focuses on the partial occlusion problem of detecting faces in unconstrained conditions. Recalling that the detection is usually the first step in another facial analysis pipeline such as face recognition. The primary target of this work is to provide the rest of the pipeline with the bounding box of the faces even if they are hard to detect because of partial occlusion. The secondary target is to provide additional information about the visible parts of the face. For example, in face recognition it is useful to know that the eyes are not visible because of sunglasses to remove them from the recognition and avoid errors resulting from trying to compare sunglasses with eyes of different subjects in the database. The videos have the advantage that some of the facial parts can be occluded in part of the video and then become visible. The purpose of this section is just to give some insights regarding how the SPM additional information about facial parts visibility can be used to analyze the facial parts of the same face that are detected many times over the video frames to provide the recognition with better information.

Current video face recognition techniques avoid the best frame approach because it does not utilize all the information provided in the video and only selects a best frame to use it in recognition. On the other hand, a straight forward recognition from all the frames is prohibitively expensive [49]. The SPM detection can provide the video face recognition

with visibility analysis of the different facial parts over the video frames to allow it to utilize more information from the video.

For example, Figure 4.7 shows 10 frames taken every 20 frames from a 7 seconds video with 30 frames per second. The video shows a person wearing sunglasses in frontal view,

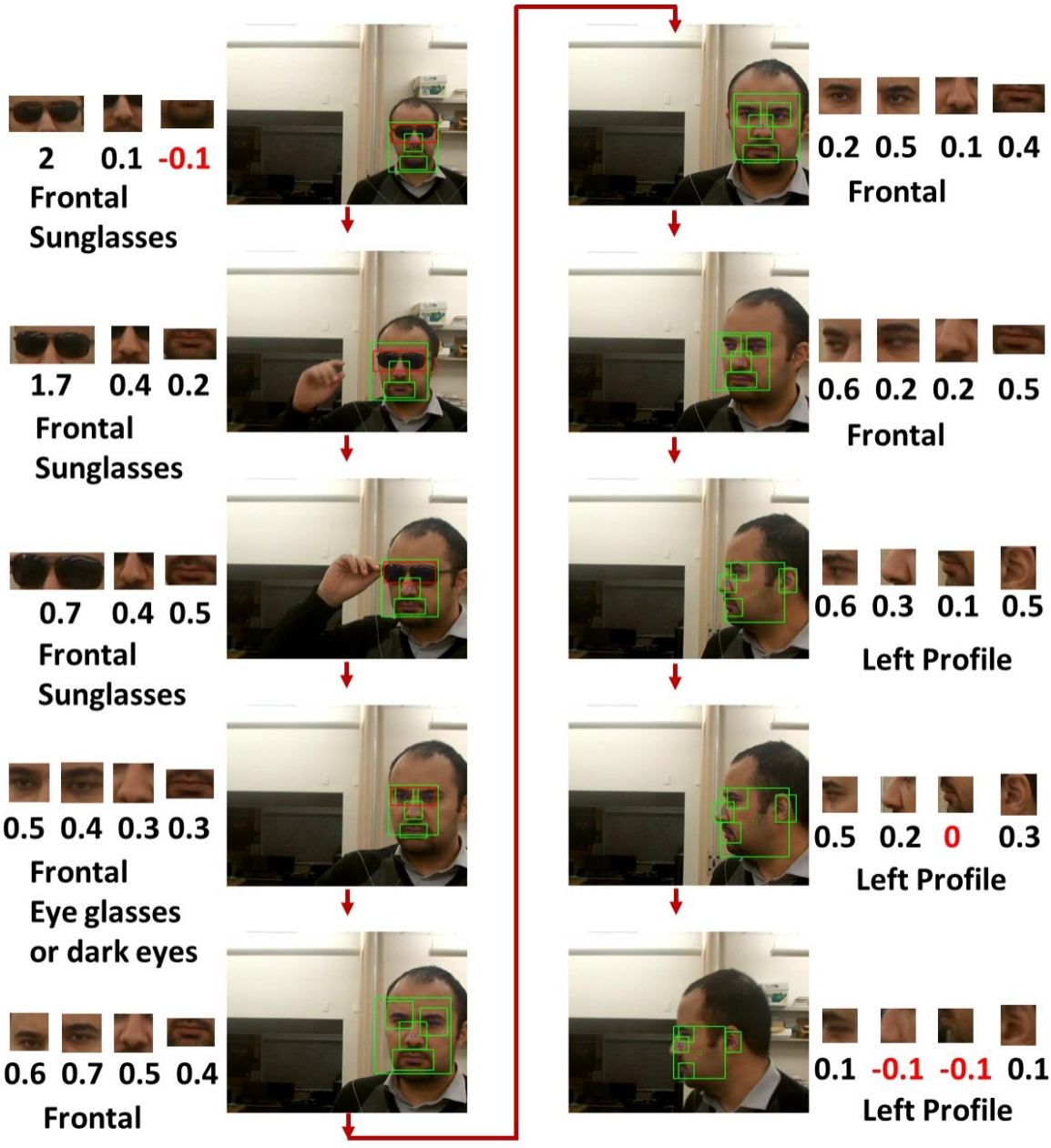


Figure 4.7: Analysis of video frames showing ten video frames taken every 20 frames

then takes it off while moving forward, and finally moves to his right giving a left profile with respect to the camera. The SPM detector provides three degrees information about the pose (Frontal, left profile, or right profile) where a frontal view was detected in the first seven frames and a left profile was detected in the last three frames. It also provides information about the visibility of the facial parts through their scores. For the eyes, if sunglasses are detected and the eyes scores are negative this means the eyes should not be used in recognition from these frames because they are not visible. On the other hand, if sunglasses are detected but the eyes scores are positive this indicates either eyeglasses, or dark eyes which can result from illumination effects. The eyes from such frame can still be used if needed. In general, a part with negative score indicates that the visibility of this part is poor and the lower the score for a part the less it should be depended on for recognition.

4.6 Conclusion

Video face detection is closely related to the face tacking problem. For faces in the wild, detection is first performed over the video frames using a powerful and fast face detector then these detections are linked into face tracks using tracking techniques. This chapter provided a detailed analysis for the detection time of the SPM detector proposed in the previous chapter, then illustrated different sources for computational redundancies in its performance. The ML-SPM was then proposed to reduce these redundancies and accelerate the detector performance. This is beneficial to applying the detector in both images and videos. The chapter was concluded by illustrating the advantages of the additional parts visibility information that the detector provides when the same face is detected many times in a video. This allows the subsequent recognition analysis to make the best use of the data available in the video.

CHAPTER 5

EXPERIMENTAL RESULTS AND APPLICATIONS

This chapter completes the discussion of the previous chapters by providing supporting experimental results and applications that show the importance of the proposed methods. Four challenging databases are used for testing the proposed detector. First the SPM is tested on the Face Detection Database and Benchmark (FDDB) which is a very recent benchmark for face detection in unconstrained environments. A main contribution of this work from the experimental point of view is a new thorough analysis of the FDDB from the occlusion perspective. Further testing is provided on the Partially Occluded Faces (POF) database which is a new database introduced in this work to bring more attention to the partial face detection problem. Comparisons of the SPM with state of the art face detectors on both the FDDB and the POF databases are provided in the form of ROC curves and tables analyzing the performance on different types of partial occlusion. In addition, the SPM is tested on the latest and largest face detection in the wild benchmark that is only released this year: “The Fine-grained Evaluation on Face detection in the wild”.

This chapter also discusses two applications for the proposed detector, the first is in security for Face Recognition at A Distance (FRAD) in which the BOSS database is used. The second application is in Human Robot Interaction (HRI) which illustrates the diversity of applications that can benefit from the proposed detector. In this application, a humanoid

robot is used to help in teaching children with autism. The detector is a first block in recognizing the children for a natural human robot interaction. The discussion is completed by proposing an application where the robot help in teaching the children how to draw simple shapes.

5.1 Testing on the FDDB Database

The FDDB was introduced in 2010 by Jain et al. [50] to serve as a benchmark for face detection in unconstrained environments. It contains 5171 faces in 2845 images with a wide range of challenges including partial occlusion, difficult poses, low resolution and out of focus (blurred) faces. The annotations of the faces are ellipses, which alluded us to report the results also as ellipses so that the overlap used in the evaluation leads to better results. Although the root filters that defines the face candidates are rectangular, the fact that the four facial parts are also detected enables deriving an ellipse from these information that matched the annotation very well in most of the images.

The evaluation schemes proposed in the benchmark ere followed, which uses two types for scoring the detections in an image. The discrete score uses the ratio of the intersection area to the union area of detection and annotation. If this ratio is greater than 0.5 then this detection is considered a true positive, otherwise it is considered a false positive. The continuous score on the other hand uses this ratio itself as a score for the detected region [50]. Figure 5.1 shows a comparison of both discrete and continuous ROC curves for the SPM method with different other methods using the evaluation code provided with the benchmark which was published in the IEEE International Conference on Image Processing (ICIP), and presented in Paris in October 2014 [36].

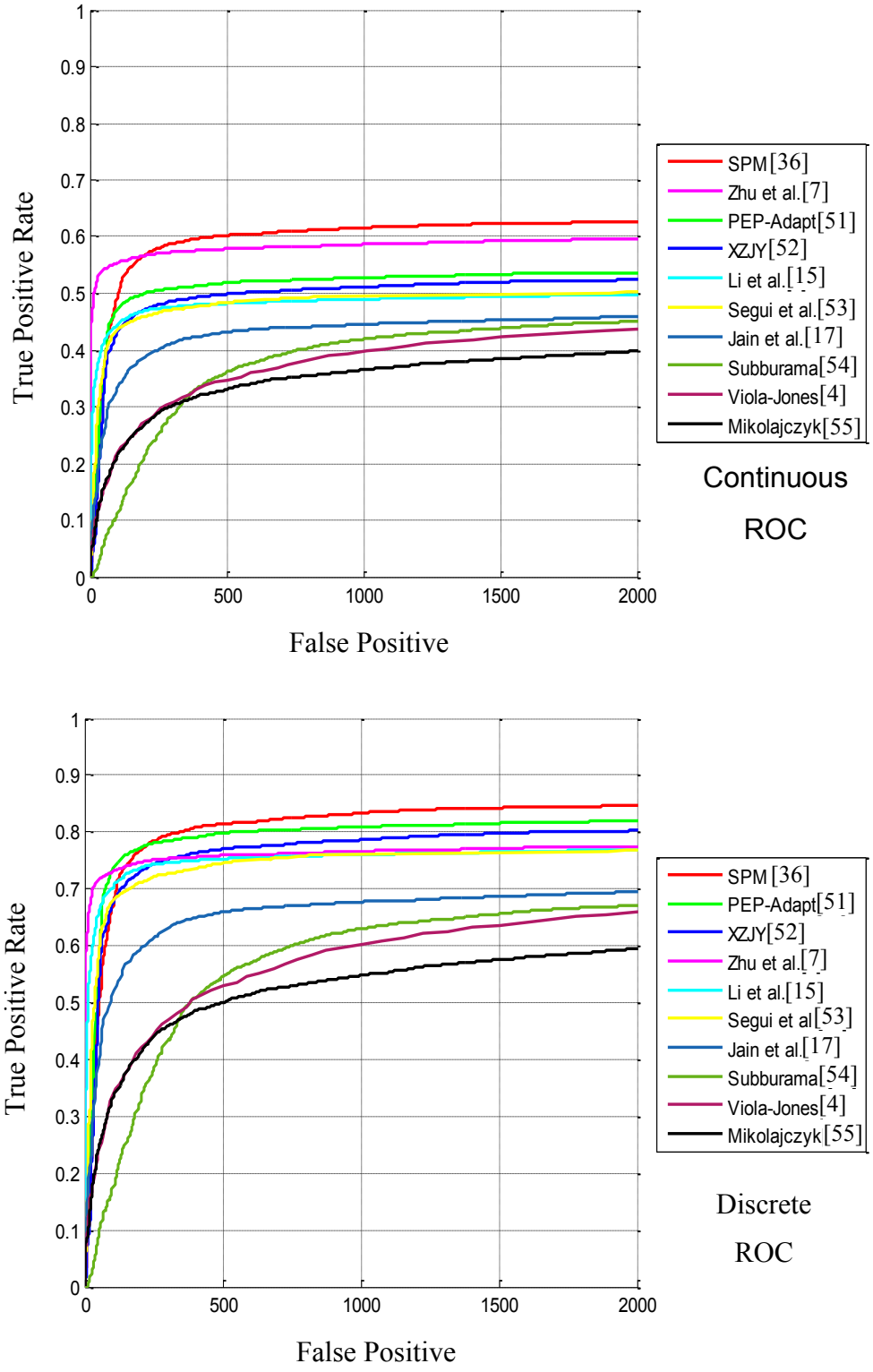


Figure 5.1: A comparison of ROC curves for different methods applied on the FDDB database.

In addition, a new thorough analysis of the 5171 faces of the FDDB is made by manually grouping these faces into 3 main categories according to occlusion then each occlusion category is further divided into sub-categories as was described in Chapter 1. Another important category that is not related to partial occlusion but it is worth mentioning to understand the FDDB results further is the blurred faces category which represents 8% of the total number of faces with more than 70% of them completely out of focus which makes them useless for any further analysis and not of interest to most applications. Examples of these faces are shown in the last row of Figure 5.2 for illustration.

I believe that with the current maturity of the face detection problem, more focus needs to be devoted for evaluating how good is an algorithm in solving certain challenges besides the overall performance that can be biased by a large number of easy near frontal faces

Table 5.1. Categorization of faces in the FDDB database

Face categories	Number of faces	FN (SPM)	FN (Zhu et al.)
No Occlusion	**3647**	**124 (3.5%)**	**653(17%)**
Self-Occlusion	**655**	**84(12.8%)**	**345(52.7%)**
– Profile	298	54 (18.1%)	229(77%)
– Caps	251	14 (5.5%)	59(24%)
– Sunglasses	55	5 (9%)	22(40%)
– Hands	51	11 (21.5%)	35(69%)
External Occlusion	**430**	**225(52.3%)**	**350(81.3%)**
– Faces occluded by other people	209	123 (59%)	166(80%)
– Faces occluded by other objects	179	86 (48%)	144(81%)
– Out of field of view	42	16 (38%)	40(95%)
Blurred faces	**440**	**313 (71%)**	**426(97%)**

Figure 5.2: Examples of FDDB results showing detections as green ellipses and annotation as red dashed ellipses. The last row shows examples of the completely blurred faces annotated in this database.

hiding deficiencies in certain challenges. Table 5.1 shows these categories and the False Negative (FN) count of each category from the proposed method (SPM) compared with the detector of Zhu et al. [7] which is selected because it is publically available and showed great performance on the FDDB database as seen in the ROC curves.

From the table, it can be seen that 70% of the FDDB faces have no occlusion at all (still 3.5% of them are not detected due to other challenges like very small size and extreme poses or expressions). The FDDB contains 21% of the total number of faces (1085 face) that are occluded with at least one part of the eyes, nose and mouth being not visible in the image. Faces were grouped into two main types: **self-occlusion** which can result from pose or other objects that belong to the same subject such as sunglasses, caps and hands; **external occlusion** which can result from other objects in front of the face including other faces and objects, or being partially outside the field of view of the camera. For self-occlusion, there is a total of 655 faces and we successfully detect 88% of them. It can also be seen from the table that caps and sunglasses showed a performance better than hands due to the large possible variations in the appearance of the hand that could not be captured easily by the model. Figure 5.2 shows some results with detections in green, and annotation in dashed red.

5.2 Testing on the POF Database

In this work, a new database for face detection in the wild focusing on the problem of partial occlusion is introduced. The images in the POF database are collected from the internet with each image containing at least one face with partial occlusion. It still contains some faces with no occlusion which are also important so that the database is not biased

toward occlusion only settings. The POF database consists of 500 images with 777 faces including different types of occlusion. Many of the images include celebrities to show that despite the severe occlusion in some images, these images are still useful for recognition because people can still recognize the celebrities in these images. The images are selected carefully so that it contains no ambiguity in the annotation for deciding whether a face should be annotated or not. The occluded faces must contain at least one part that is completely visible. The POF database contains no faces that are completely out of focus like many faces in the FDDB, although some slight blurring is still accepted to keep all the challenges in the database as long as the faces can still be useful for recognition or other further analysis. Both image dimensions are less than 1000 pixels with the faces annotated as rectangles with both sides at least 40 pixels. Because the annotation of the POF is in the form of rectangles not ellipses, we followed the same concept we did in FDDB and reported our results in the same form of the annotation.

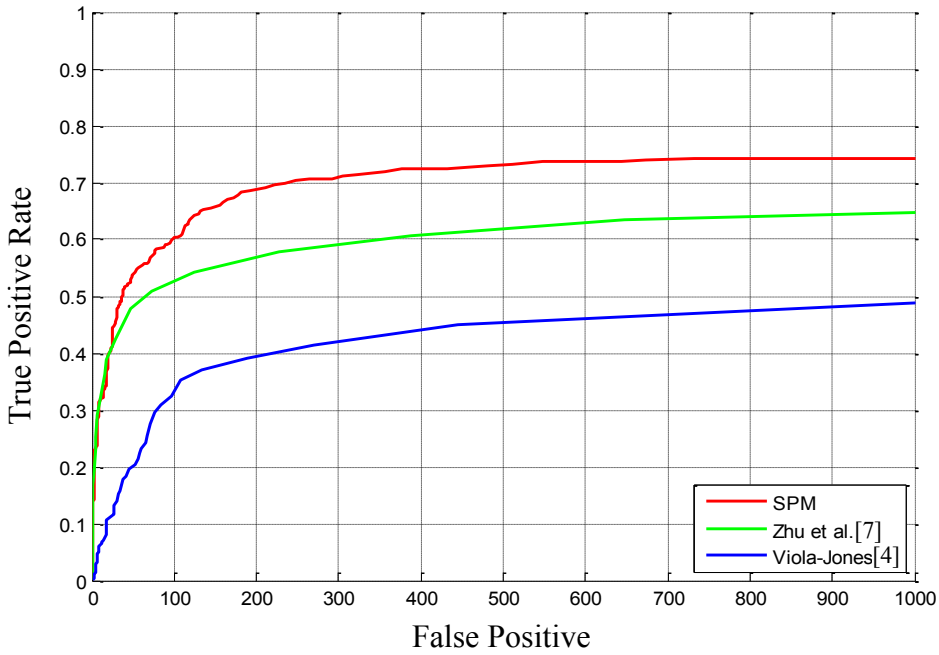


Figure 5.3: A comparison of ROC curves for different methods applied on the POF database.

To evaluate the performance of the SPM on the POF database, its results are compared to the detector of Zhu et al. [7] and the VJ face detector [4] which is considered as a baseline. The ROC curves are shown in Figure 5.3 comparing the three methods using the discrete score described in the FDDB. The advantage in the POF over the FDDB for evaluating partial face detection is that the ROC curves are not biased by a large number of faces that contain no occlusion as in the FDDB and hence the curves can reflect the performance of each method in handling occlusion problems in unconstrained settings with non-cooperative subjects.

To better analyze the performance, Table 5.2 shows a categorization of the 777 faces in the database into three main categories. Faces with no occlusion which are 15% only of the total number of faces compared to 70% in the FDDB. Faces with self-occlusion representing 60% of the database with 40% of them having more than one type of self-

Table 5.2. Categorization of faces in the POF database

Face categories	**Number of faces**	**FN (SPM)**	**FN (Zhu et al.)**
No Occlusion	**117**	**16 (14%)**	**27(23%)**
Self-Occlusion	**470**	**106(23%)**	**147(31%)**
– Profile only	38	14 (37%)	20(53%)
– Caps only	95	22 (23%)	28(29%)
– Sunglasses only	63	6 (9%)	12(19%)
– Hands only	85	21 (25%)	25(29%)
– Mix	189	43 (23%)	62 (33%)
External Occlusion	**190**	**67(35%)**	**76(40%)**
– Faces occluded by other people	40	18 (45%)	18(45%)
– Faces occluded by other objects	150	49 (33%)	58(39%)

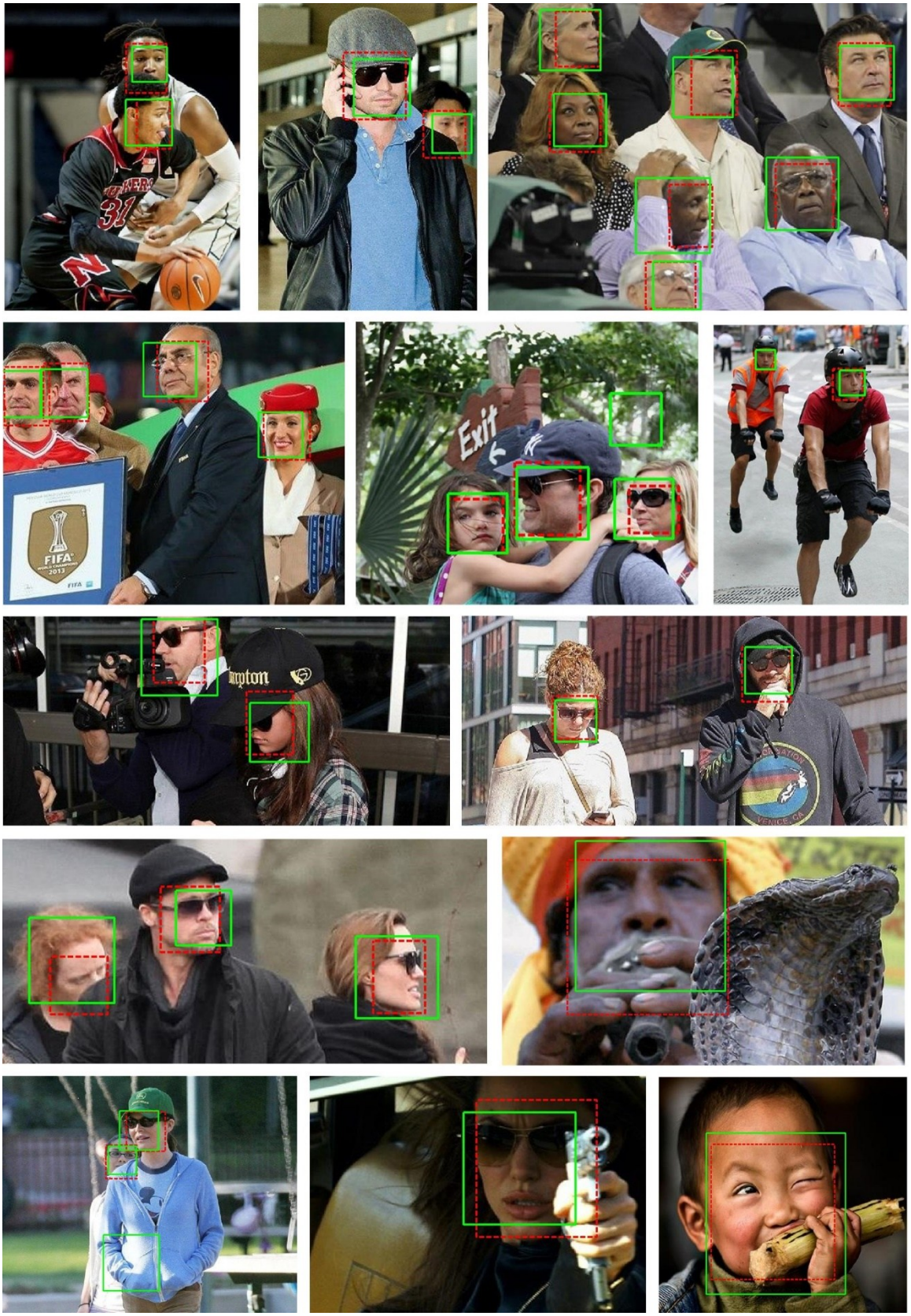


Figure 5.4: Examples of POF results showing the detections as green rectangles and the annotation as red dashed rectangles.

occlusion, for example caps hiding eyes, with hands hiding mouth which leaves the nose as the only visible parts of the face (referred to in the table as 'Mix'). External occlusion on the other hand represents the remaining 25% of the faces with most of them occluded by other object which can still be weekly detected using the visible parts. Figure 5.4 shows examples of the images in the POF database with detections as green rectangles and annotation as red dashed rectangles.

5.3 The Fine-grained Evaluation on Face Detection in the Wild

This fine-grained evaluation and benchmark was introduced by Yan et al. [56] in 2014 as a part of the evaluation in the upcoming IEEE International conference on Automatic Face and Gesture Recognition (FG 2015). The benchmark was completed in September 2014 to be the largest face detection benchmark containing 11,931 faces in 5250 images. The SPM was one of the methods participating in that evaluation which had the submission deadline on October 31st, 2014.

This benchmark provides annotations that include a square bounding box of the faces and several additional attributes to make the fine-grained analysis of face detection results possible. The additional attributes include pose level of yaw, pitch and roll as small, medium, or large. It also includes an ignore flag for faces which are smaller than 20x20 or extremely difficult to recognize which are 838 faces, this is important because these faces are not counted as false negatives if not detected but they are also not counted as false positives if detected because a good detector capable of detecting them should not be penalized. In addition other Boolean attributes are also included in the annotation including: gender, isBaby, isWearingGlasses, isOccluded, and isExaggeratedExpression.

Their annotation strategy followed the following guidelines: the bounding box style is similar to the AFLW dataset [41] which tries to contain the eyebrow, the chin and the cheek, while keeping the nose located approximately at the center. In order to keep their annotation style consistent, the bounding box was annotated by two persons and examined by one, and the ignore flag was annotated by one person. Finally, the gender attribute of babies was discarded due to the ambiguity.

In this benchmark, besides measuring the overall performance, the evaluation is fine-grained because it also reports the specific performance with regard to occlusion, gender, glasses, expression, resolution and pose. In this way, one can clearly observe the advantages and disadvantages of different face detection algorithms from various aspects. The evaluation in different aspects can be generated by only taking the faces under each specific circumstance into consideration. This matches the previously mentioned contribution in the FDDB and POF databases, in which the faces were classified with respect to occlusion types to evaluate the performance according to them [36]. It is the same idea of not just reporting face detection over all the faces together where frontal easy faces usually exceed in numbers the challenging faces and hence it is difficult to know if a detector is good with respect to a specific challenge.

During the evaluation, only the images without annotations were released. Only 250 images were released with annotations to serve as examples for refining the bounding box and were not used for evaluation. The results are compared against several methods from the academia and the industry. From the academia, Viola Jones as implemented in OpenCV using two views frontal and profile [4], weighted sampling based boosting of Kalal et al.[57], SURF frontal of Li et al. [5], and tree structured model of Zhu et al. [7]. Three of

these methods were used in the FDDB analysis and two of them were used in the POF analysis. From the industry, Google Picasa and Face++ were used.

The evaluation produced 19 ROC curves comparing the SPM (tagged as CVIP-Run1) with the previously mentioned methods, the overall response over the whole data is shown in Figure 5.5 while the fine grained analysis are shown in Figure 5.6 to 5.11. The curves show good performance for the SPM method in the overall performance as well as the in the other attributes. Four methods are evaluated as single points because their source code is not available while the other methods are shown as ROC curves where each point corresponds to different threshold. Only the commercial Google Picasa exceeded the SPM and it is worth mentioning that this is an auto-tagging software that performs the whole recognition pipeline so produces very low false positives.

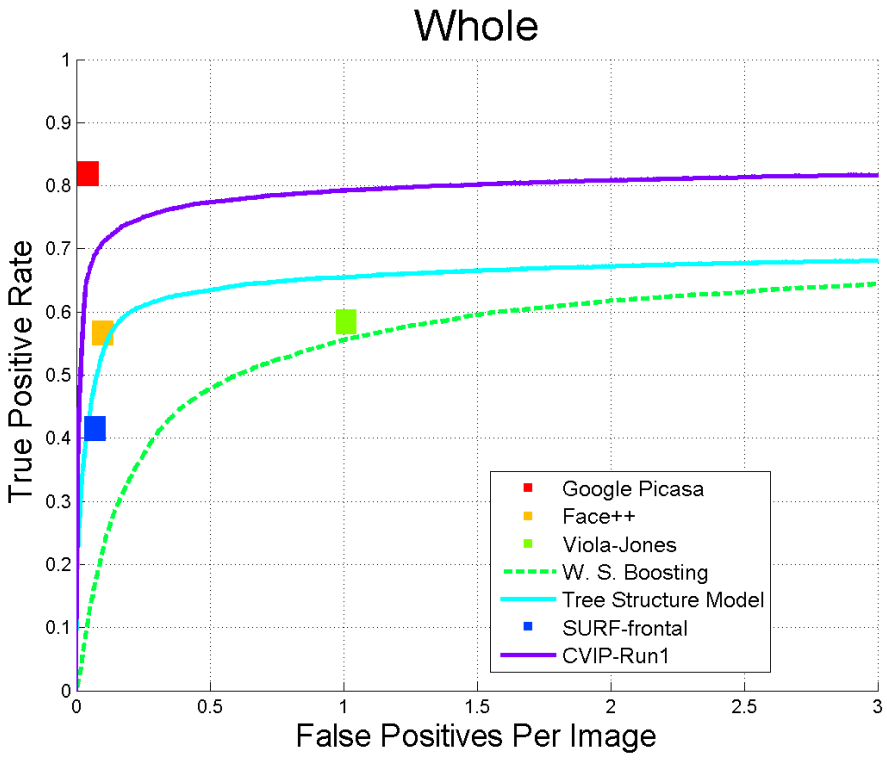


Figure 5.5: overall performance over all the data in the latest benchmark of face detection in the wild.

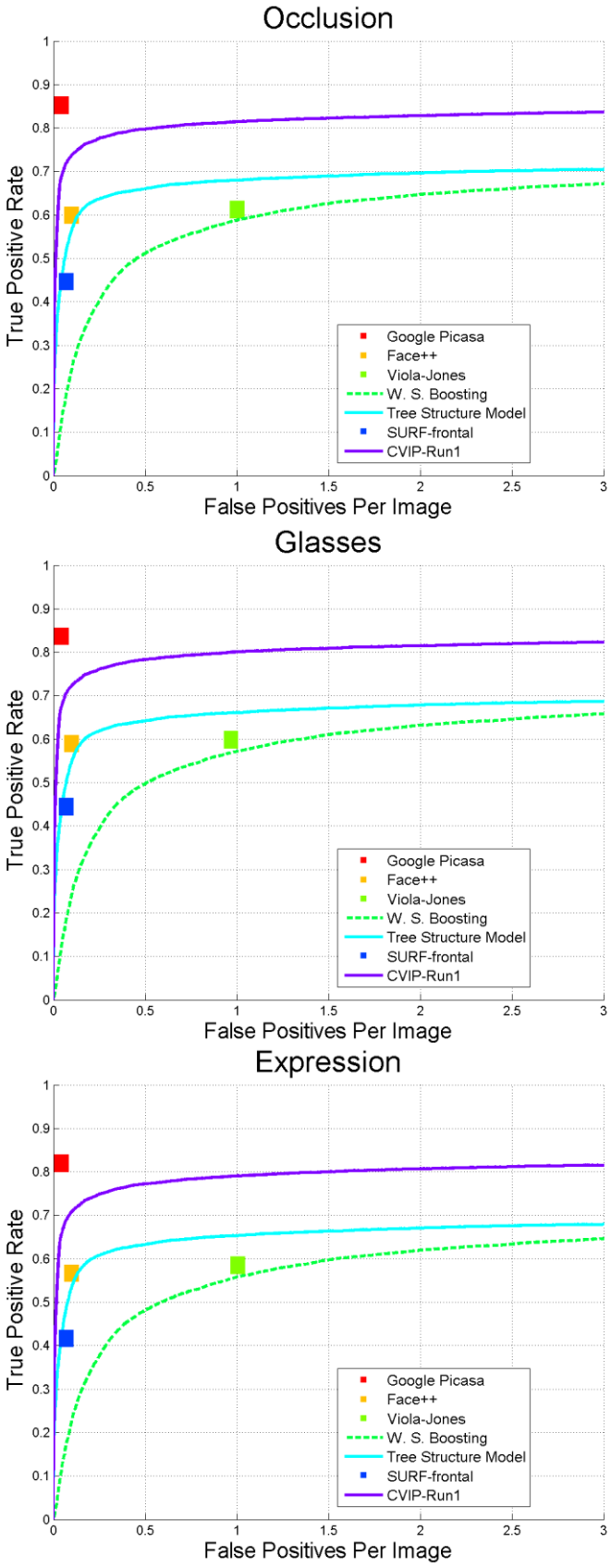


Figure 5.6: Fine-grained evaluation with respect to Occlusion, Glasses, and Expression.

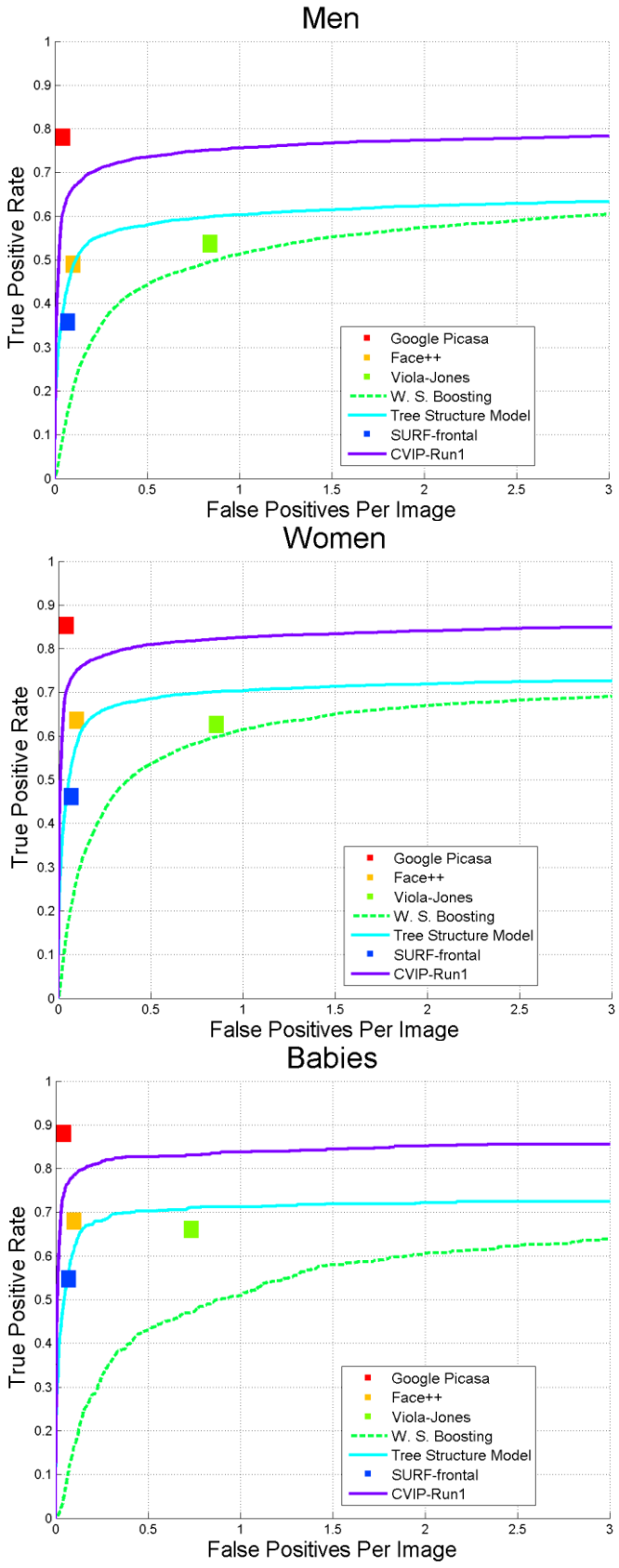


Figure 5.7: Fine-grained evaluation with respect to Gender: Men, Women, and Babies.

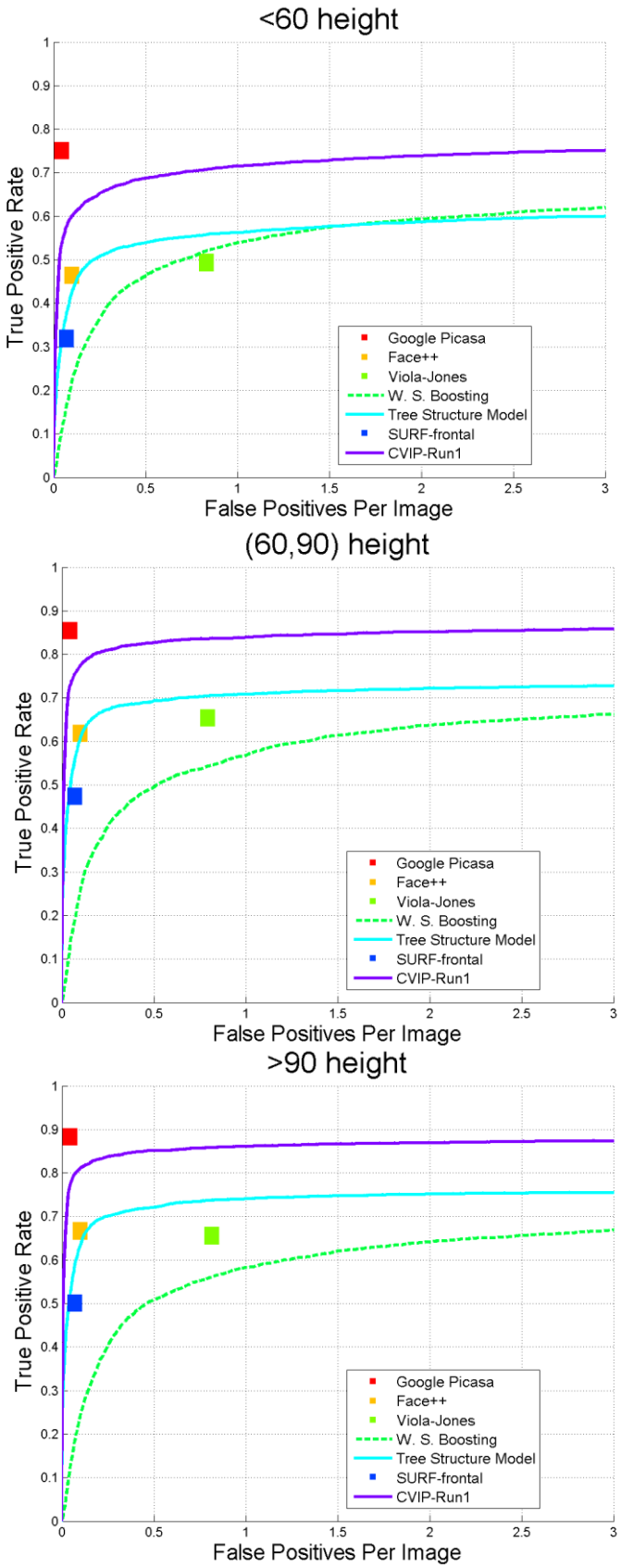


Figure 5.8: Fine-grained evaluation with respect to face size: <60, (60,90), and >90.

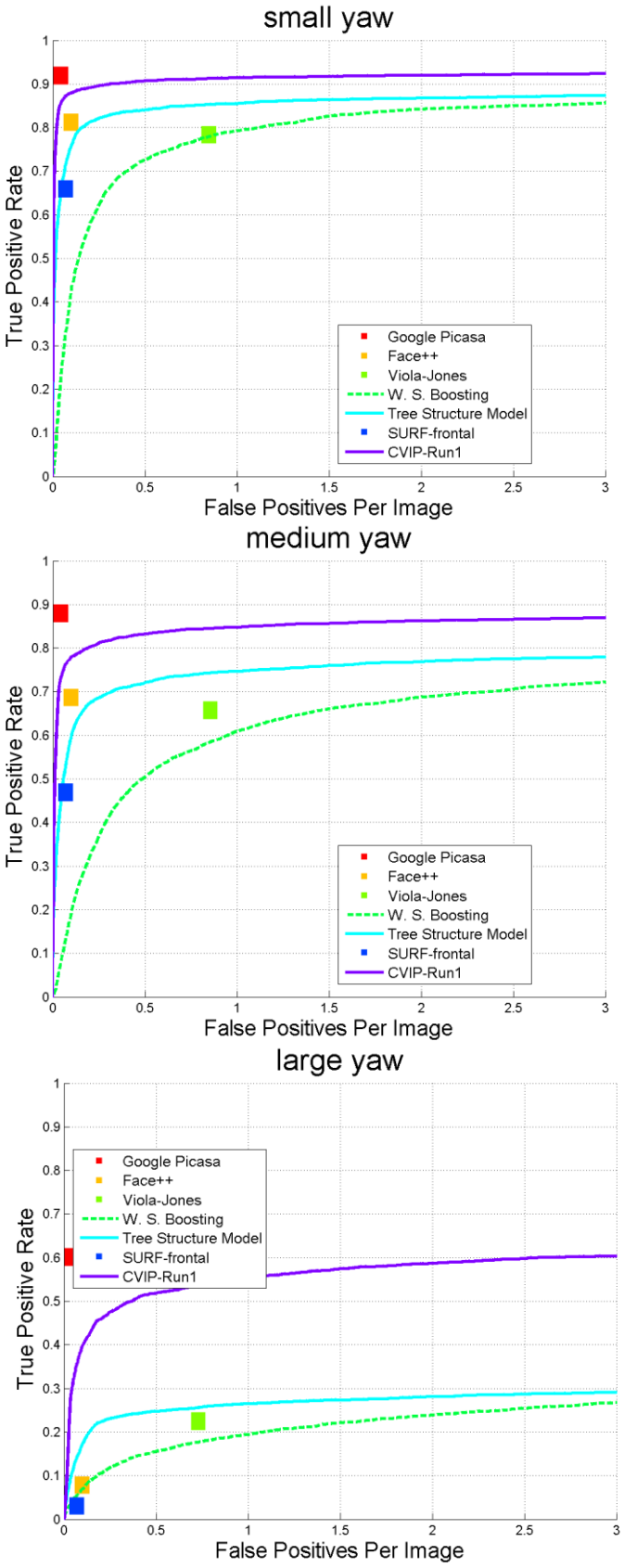


Figure 5.9: Fine-grained evaluation with respect to pose: Yaw angle.

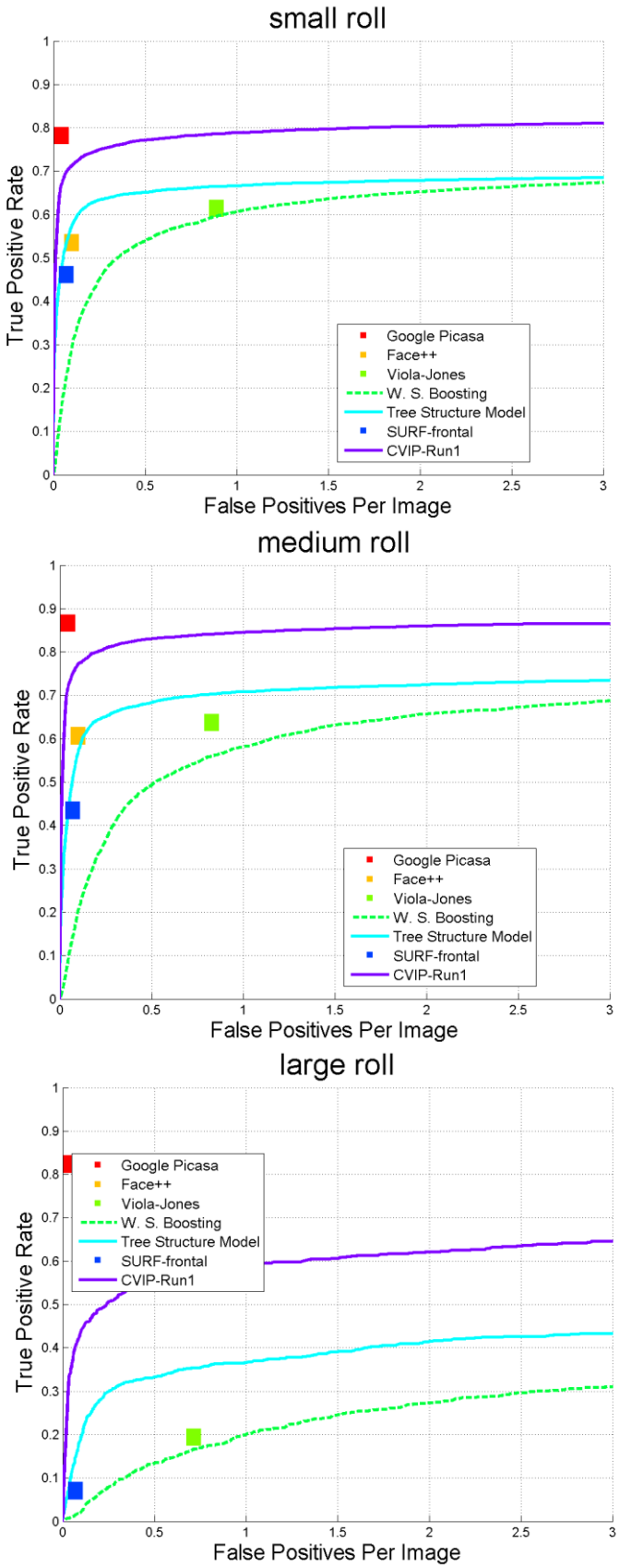


Figure 5.10: Fine-grained evaluation with respect to Pose: Roll angle.

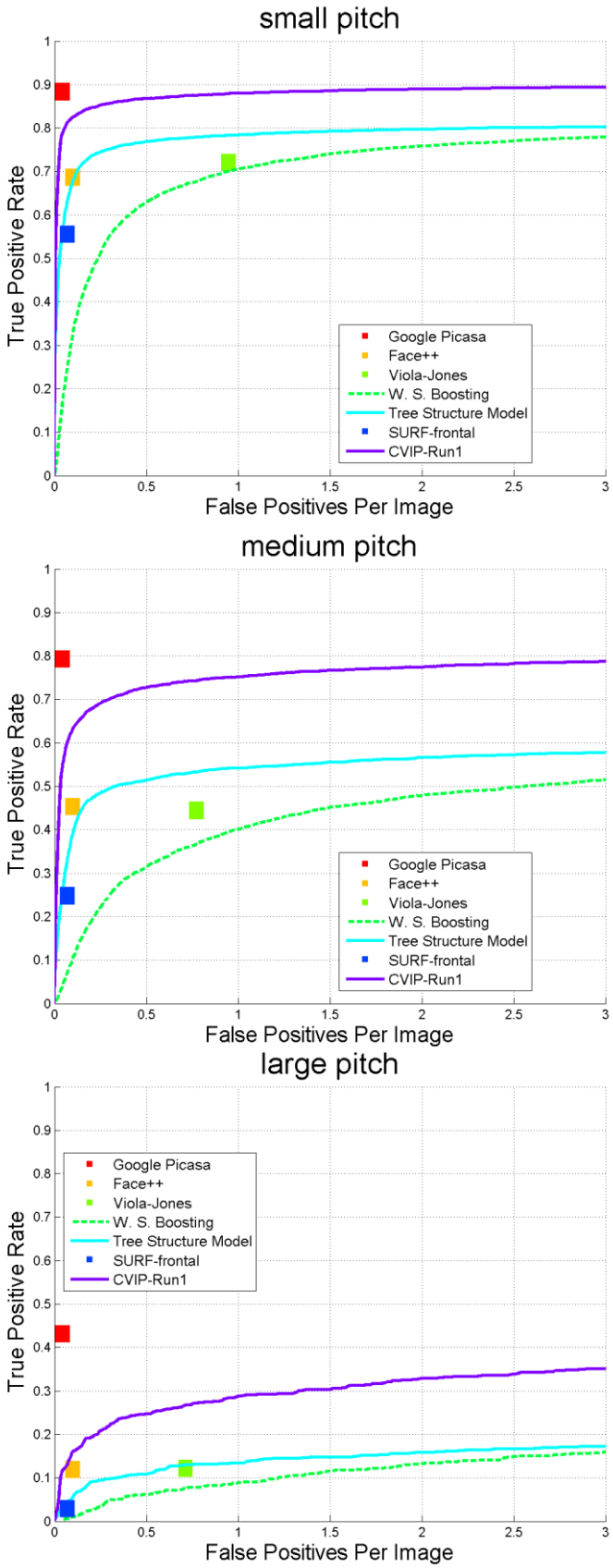


Figure 5.11: Fine-grained evaluation with respect to Pose: Pitch angle.

5.4 Application I: Face Recognition at A Distance

This section describes the experimental results of the RSFFD explained in chapter 2 which is the face detector used in the Biometric Optical Surveillance System (BOSS) project conducted by the CVIP laboratory and funded by the Department of Homeland Security for building a face recognition at a distance system (October 2010 to October 2012). The BOSS database will first be described explaining how it was collected then the results of the RSFFD approach will be compared to previous state-of-the-art algorithms to determine its efficiency. A discussion of results will follow to study its advantages and limitations. The RSFFD is a generic framework that can be used with any face detector, so its skin part was used with the SPM detector as a post processing step to further enhance its performance as was explained in Chapter 3.

5.4.1 The BOSS Database

The BOSS database is collected outdoor using two NFOV cameras with 800 mm lenses at distances ranging from 30-meters to 150-meters. The database consists of 1191 image with 2076 faces. The number of subjects in each image varies from 1 to 12 as shown in Figure 5.13. The database has 120 different subjects including different ages, skin colors and genders. The lighting follow unconstrained day light conditions that varies from sunny to cloudy with also some shadow problems. The pose goes up to $\pm 45^{o}$ in the yaw angle and $\pm 10^{o}$ in the pitch and roll angles. The expressions are unconstrained including smiling, laughing, sadness and anger. The backgrounds are unconstrained complex outdoor environments with a wide variety of objects. The ground truth of the images is annotated manually as rectangles.

5.4.2 Results and Discussion

The database described above was used to test the RSFFD algorithm with the VJ face detector as its base detector. For comparison, the VJ detector is used with two versions of the RSFFD where version 1 proposed in the competition held by Parris et al. [26] did not use the skin and saliency scores explained in Chapter 2 which were developed later on for version 2 that was proposed in EL-Barkouky et al. [27]. Figure 5.12 shows the ROC curves obtained from applying the three methods (VJ and VJ with RSFFD version1 and 2) with different threshold values. The threshold of 7 was selected according to that curve for a balanced performance in the tradeoff between true positive rate and false positives. A zoomed version of the three curves illustrates the difference between the three approaches near the acceptable range of false positives. With almost 2000 faces in the database 200 was considered as the maximum allowed false positive. The operating point on the approach was selected at a true positive rate of 0.87 and a false positive count of 79. The

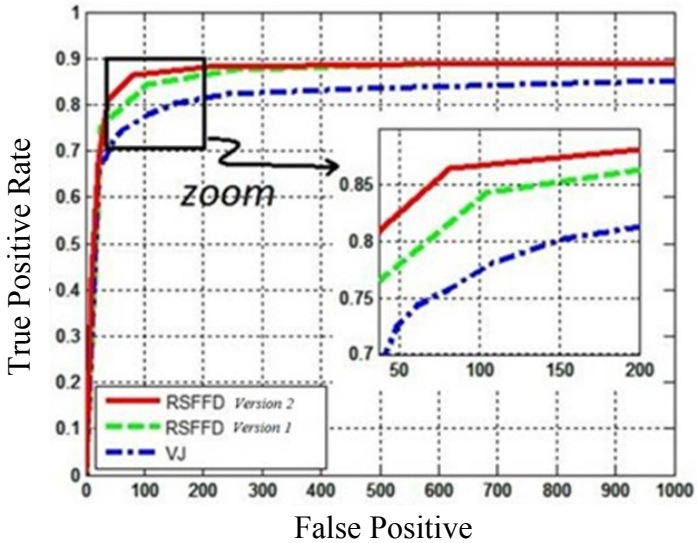


Figure 5.12: A comparison of ROC curves for the VJ face detector with the RSFFD version 1 and 2 that also use the VJ face detector as its base detector. The curves support the effectiveness of the method in reducing the false positives allowing the detector to operate on a better operating point.

Figure 5.13: Examples of outdoor images at different distances from 30 to 150 meters with different challenges. The candidates that pass a threshold of score of 7 are considered faces and displayed in solid green. The candidates that were rejected because they have score less than 7 are shown as dotted red rectangles just for illustration.

VJ was used in these results just as a proof of concept which can be replaced by the SPM detector or any other detector.

Figure 5.13 shows some of the results obtained by the RSFFD approach on images that are captured at distances ranging from 30 to 150 meters. The green solid rectangles are the candidates that passed a threshold of 7 in their final score. The red dotted rectangles are the candidates that were rejected because their scores were lower than the threshold. The images show the large number of false positives obtained due to the outdoor complex setup and illustrates the efficiency of the proposed approach to reject these false positives. It also one true face that was mistakenly rejected because it had a score of 6 in the left lower image. The threshold is obtained from the ROC for the best performance over all images.

5.5 Application II: Human Robot Interaction

In this application, the face detection is used in a completely different setting away from its common security applications that first hits the mind when face recognition in general is considered. With the current advancement in technology, most machines became smart in the sense that it contains a processor that enables them to analyze the data perceived from the environment. One of the key aspects to be analyzed by the machine is the human face which allows it to interact naturally with humans. This suggests tailoring the facial analysis development for the human machine interaction in general and for the human robot interaction which is of particular interest in this work.

In this context, an educational robotic system is proposed as a helping tool for teaching children with autism using a humanoid robot. According to the Centers for Disease Control and prevention, it is estimated that in the Unites States 1 in 88 children is diagnosed with Autism [58]. Children with autism suffer from problems in social interactions and communication. A robot can provide social interactions in a much simpler way because of its predictability and fewer external stimuli which often attract those children. For the robot to be able to interact with the children, two main tasks are required. First, the robot needs to be able to perceive the environment and in this aspect the focus is on doing this using cameras and face detection which is needed for natural interaction with the children. This will enable the robot to detect the child's face and facial parts and some of the occluding objects that are common in a typical class room settings and use them in the SPM framework proposed earlier. Besides detecting the children's faces for recognizing them, the robot should perform actions to interact with these children and help in teaching them.

Some of these actions will be talking, moving, dancing, and of special interest in this work: enabling the robot to write and draw simple shapes.

Although the main focus of this dissertation is on using the face detection in natural interaction of the robot with the children, this needed to be put into an application that supports their educational needs where the face detection and recognition can be used to enhance the interaction. The selected application was enabling the robot to write and draw simple shapes. These shapes can be coming from analyzing images either captured by its cameras or stored in its hard drive. In this context, a novel mapping from the image domain to the robot space is also proposed which will be explained in this section to complete the discussion of this application.

5.5.1 The drawing mapping

In this section, the robot's target is to draw a simple shape or an alphabetical letter from an image that is either captured by its cameras or stored in its hard drive. The required shape should be segmented and its contours are then extracted. Here, the focus is on how the points of the lines and curves that the robot is supposed to draw can be transformed from the image domain to the robot's joints angles so that the robot can produce the proper movement of its arm to draw the same lines and curves on a piece of paper as illustrated in Figure 5.14. This transformation is done over two stages: the first stage is to transform each point from the image domain to the paper domain and then the second one is to transform each point in the paper domain to the corresponding robot's joints angles that will move the robot arm, and hence the pen, to this point on the paper.

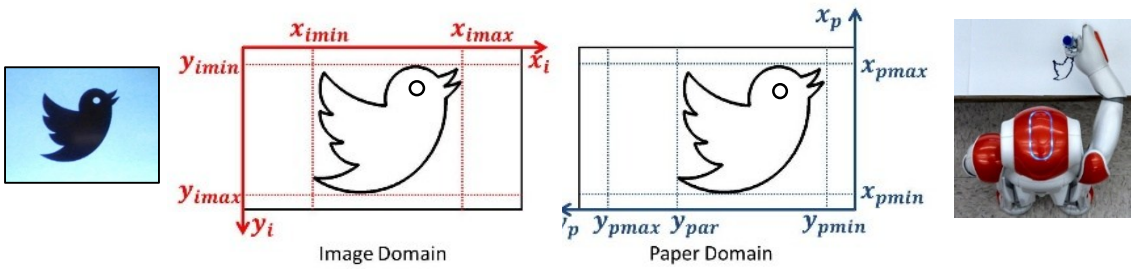


Figure 5.14: The transformation from the image domain to the paper domain.

5.5.1.1 From image domain to paper domain

A linear transformation is used to relate the image domain to the paper domain using the following equations:

$$x_p = x_{pmin} + \frac{x_{pmax} - x_{pmin}}{y_{imax} - y_{imin}} (y_{imax} - y_i) \tag{5.1}$$

$$y_p = y_{pmin} + \frac{y_{par} - y_{pmin}}{x_{imax} - x_{imin}} (x_{imax} - x_i) \tag{5.2}$$

Where (x_i, y_i) is any point in the image domain and (x_p, y_p) is the corresponding point in the paper domain. The values x_{imin}, x_{imax}, y_{imin} and y_{imax} are the minimum and maximum values of x_i and y_i for all the points of the lines and curves the robot is supposed to draw in this image and hence they will change from one image to another. x_{pmin}, x_{pmax}, y_{pmin} and y_{pmax} are the boundaries of the selected region in the paper for the robot to draw in as shown in Figure 5.14. To maintain the aspect ratio of the shape that the robot is drawing y_{pmax} is replaced in the transformation by y_{par} which is chosen such that:

$$\frac{y_{imax} - y_{imin}}{x_{imax} - x_{imin}} = \frac{x_{pmax} - x_{pmin}}{y_{par} - y_{pmin}} \tag{5.3}$$

Then finally Solve for y_{par}:

$$y_{par} = y_{pmin} + \frac{x_{imax} - x_{imin}}{y_{imax} - y_{imin}} \left(x_{pmax} - x_{pmin}\right) \quad (5.4)$$

5.5.1.2 From paper domain to robot's joints angles

The right arm of the humanoid robot NAO has 6 degrees of freedom (DOF) as shown in Figure 5.15. One DOF is in the hand, the robot asks for the pen while its hand is open. Then when the user gives the pen to the robot, he should activate the touch sensor on its hand which is programmed to let the robot close its hand holding the pen. The other five DOF are used to let the robot raise its right arm parallel to the paper. The shoulder roll and elbow roll angles are used to enable reaching the different points on the paper required for drawing. The shoulder pitch angle is set to zero for the robot to draw and to a negative value for the robot to raise his hand slightly causing the pen not to touch the paper which is used to move between different contours. The elbow and wrist yaw angles are set to 0 and 90 degrees respectively to let the pen be perpendicular to the table. The length of the upper arm is referred to as "$a = 10.5\ cm$", the sum of the lower arm and hand offset as "b=11.37cm" and the elbow offset as "d=1.5cm" which are shown in Figure 5.15 and 5.16.

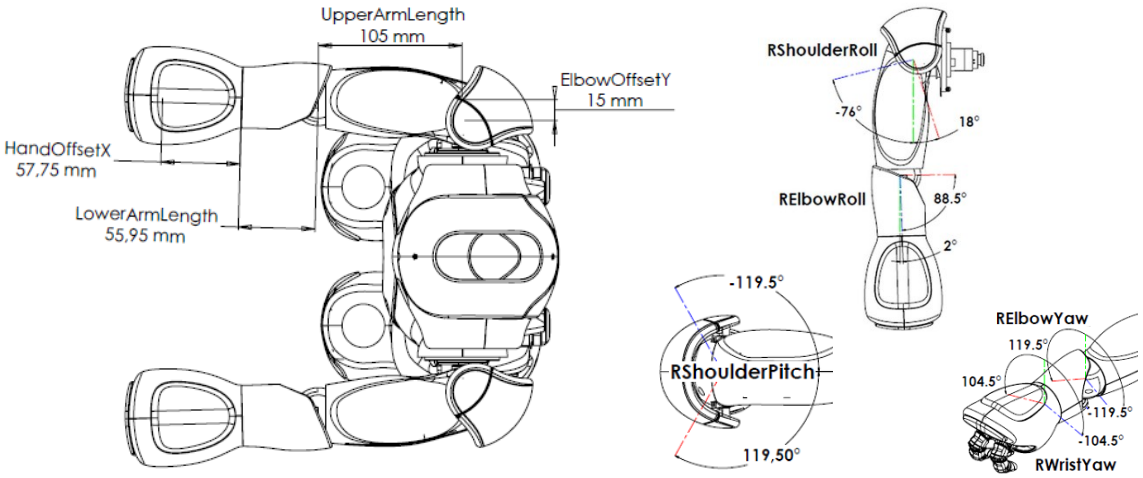


Figure 5.15: NAO's right arm joints and dimensions.

The point (x_p, y_p) in the paper domain should be transformed to the angles S and E which represent the right arm shoulder roll and elbow roll angles respectively as shown in Figure 5.16. Starting from the inverse transformation namely from S and E to(x_p, y_p) :

$$x_p = a\cos(S-\delta) + b\,cos(S-\delta+E+\theta) \tag{5.5}$$

$$y_p = a\sin(S-\delta) + b\,sin(S-\delta+E+\theta) \tag{5.6}$$

where "a" and "b" are the lengths of the upper arm and forearm respectively while "δ" and "θ" are the angles resulting from the offset "d" between the shoulder joint and the elbow joint in the y direction as shown in Figure 5.16. The angles "δ" and "θ" can be calculated as 8.21 and 15.79 degrees respectively from the geometry of the arm using:

$$\delta = sin^{-1}\left(\frac{d}{a}\right),\ \theta = \pi - cos^{-1}\left(\frac{d}{b}\right) - \left(\frac{\pi}{2} - \delta\right) \tag{5.7}$$

To find the reverse transform, equations (5.5) and (5.6) are squared then added leading to:

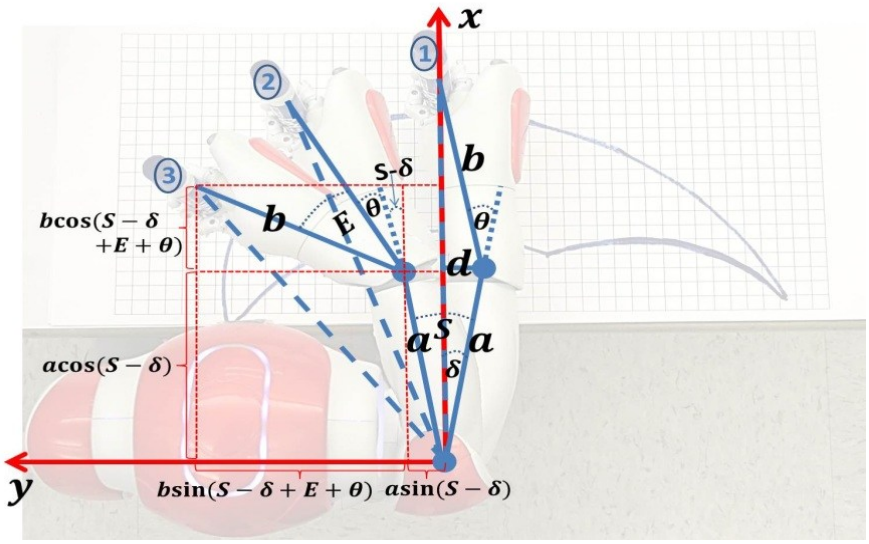


Figure 5.16: The transformation from the paper domain to the robot joints angles.

$$x_p^2 + y_p^2 = a^2 + b^2 + 2abcos(E + \theta) \tag{5.8}$$

Then by solving for E:

$$E = -\theta + cos^{-1}\left(\frac{x_p^2+y_p^2-a^2-b^2}{2ab}\right) \tag{5.9}$$

To get S we expand the cosine and sine functions in equations (5.5) and (5.6) in terms of $(S - \delta)$ and $(E + \theta)$:

$$x_p = (a + b\cos(E + \theta))\cos(S - \delta) - bsin(E + \theta)\sin(S - \delta) \tag{5.10}$$

$$y_p = (a + b\cos(E + \theta))\sin(S - \delta) + bsin(E + \theta)\cos(S - \delta) \tag{5.11}$$

Solving equations (5.10), (5.11) for $cos(S - \delta)$ and $sin(S - \delta)$ by multiplying equation (5.10) by $(a + bcos(E + \theta))$ and equation (5.11) by $bsin(E + \theta)$ then adding and subtracting we get:

$$\cos(S - \delta) = \frac{x_p(a+bcos(E+\theta))+y_p bsin(E+\theta)}{a^2+b^2+2abcos(E+\theta)} \tag{5.12}$$

$$\sin(S - \delta) = \frac{y_p(a+bcos(E+\theta))-x_p bsin(E+\theta)}{a^2+b^2+2abcos(E+\theta)} \tag{5.13}$$

Dividing equation (5.13) by equation (5.12) and solving for S we get:

$$\mathrm{S} = \delta + tan^{-1}\frac{y_p(a+bcos(E+\theta))-x_p bsin(E+\theta)}{x_p(a+bcos(E+\theta))+y_p bsin(E+\theta)} \tag{5.14}$$

Equations (5.9) and (5.14) will be used to transform any point on the paper plane (x_p, y_p) to the angles S and E that will move the pen to this point.

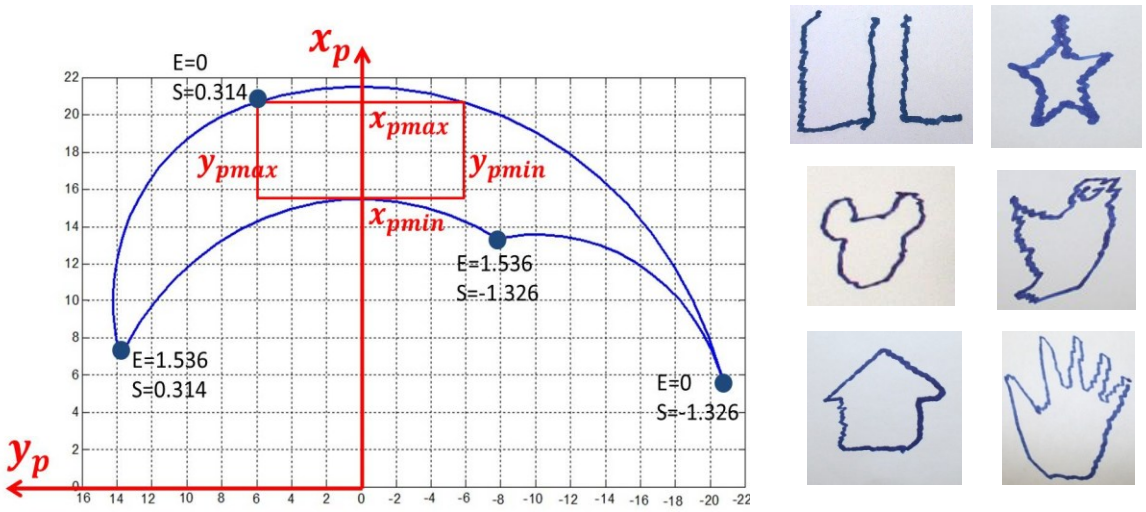


Figure 5.17: The drawing region for Nao and some examples of Nao drawings.

The region of drawing in the paper is shown in blue in Figure 5.17 along with some examples of Nao drawings. The rectangle shown in red was selected within that region to be the limits of the drawing area with $x_{pmin}, x_{pmax}, y_{pmin}$ and y_{pmax} equal to 15.5, 20.5, -6 and 6 respectively. These values controlled the region of the paper that Nao can draw in without walking. Of course, Nao can walk a step to the left, right or back to move that rectangle in case he needs to draw several images.

5.5.2 Face detection with Nao

The humanoid robot Nao has 25 degrees of freedom, 2 of them are in his head allowing it to rotate in the yaw 119.5º to the right or left, and in the pitch 29.5º looking down or -38.5º looking up. The robot's head is equipped with two identical video cameras one of them located in the forehead and the other located in the mouth. They provide up to 1280x960 resolution at 30 frames per second if used on the local processor, and if transferred to a laptop the frame rate is based on the network and the resolution used. These technical specifications for Nao's head movement and for its cameras including the camera's field of view and locations are illustrated in Fig 5.18.

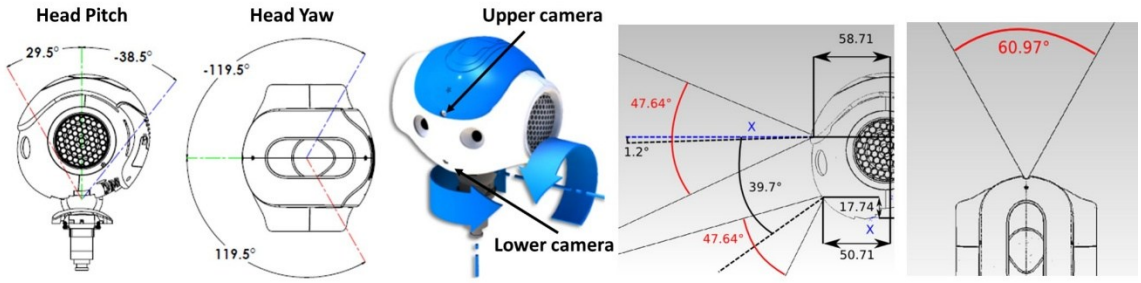


Figure 5.18: Nao's head movement and cameras.

The processing of video frames captured by Nao's camera can either be done on the local processor inside the robot or though the WiFi network on a laptop. The laptop method was adopted in which the frames captured by Nao's camera are sent over the WiFi and then the processing of the frames are done on the laptop just like applying the SPM to any other image or video. The frame resolution used is 480x640 (VGA) at a rate of 1 frame per second. The target is to keep the robot aware with the kids around it.

For example, one scenario can be that Nao and the kids are sitting around a table for the drawing application. In this case, a yaw head movement gradually from -45^{o} to 45^{o} is used to span the whole space around the table and locating how many kids are sitting and their locations. Then if the robot wants to look to one of them, it can move its head until the detected face is in the middle of the frame. For a more natural interaction, the face detection should be the first step in a recognition module that also recognize the kids' faces. Another scenario for the robot's interaction with the kids is that Nao is dancing in front of the kids and the kids are mimicking his movements. In this settings the head movement is not necessary because the distance between the robot and the kids makes the camera's field of view large enough to capture all the kids.

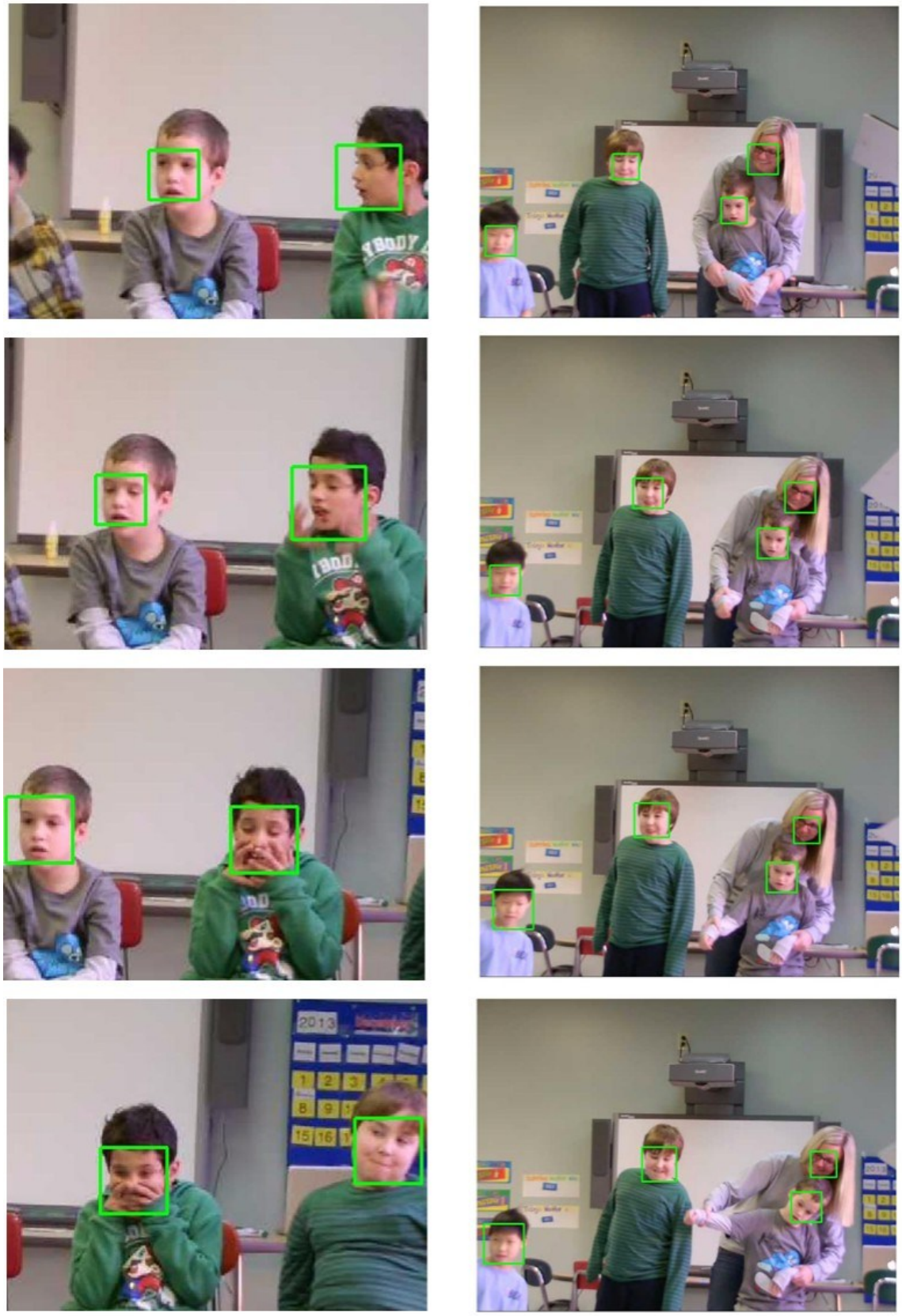


Figure 5.19: SPM tested on images captured at the Bluegrass center for autism.

Figure 5.19 shows some of the results of the SPM detector on frames captured at the Bluegrass center for autism. The frames show a success for the SPM face detector on frontal and profile poses with different expressions. It also shows successful detections for faces partially occluded by hands for the kid on the left sequence. It also shows successful detections for the teacher's face partially occluded by child head on the right sequence.

This is part of the autism robotics project in the CVIP lab in collaboration with the Bluegrass Center for Autism. It was awarded first place in the graduate research symposium at the University of Louisville in 2013 and the Diebold research award from the Speed School of Engineering at the University of Louisville in 2014. It was published in the ICIP 2013 and presented at the conference in Australia [59]. In 2014, it was also recognized in the WHAS 11 TV channel and the University of Louisville alumni magazine as part of the research work in the CVIP lab as shown in Figure 5.20.

Figure 5.20: The autism robotics project at the CVIP lab recognized in the WHAS 11 TV channel and the University of Louisville alumni magazine.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Faces in the wild impose difficult challenges even on the state of the art face detectors. The wide variability in the unconstrained real life environments complicates the process of building a face model that can detect faces with different poses, expressions and lighting conditions from a complex background. The partial occlusion of these faces adds further complications that can prevent even commercial face detectors from detecting these partial faces. In this work, the problem of partial face detection in uncontrolled environments is tackled from a scene understanding point of view by modeling some of the common objects that can occlude faces in a selective part model framework that can be used not only to detect the faces but also to provide information about the visible parts of these faces which can be used in any further facial analysis step.

6.1 Conclusion

A detailed overview of the face detection problem was first provided focusing on two major general frameworks: the Adaboost framework leaded by the Viola Jones approach and the part based framework leaded by the DPM approach. Those two frameworks can be considered the basis of most current face detectors. The SPM proposed in this work belongs to the part based framework and is distinguished from other methods by its explicit focus on the partial occlusion problem. The idea was demonstrated on self-occlusion resulting

from sunglasses, caps and hands and on external occlusion resulting from other people's faces and shoulders. The SPM has a lot of potential for partial face detection. A detailed analysis of the detection time and computational redundancies of the SPM was provided leading to the ML-SPM as a modification that reduces these computational redundancies and accelerates the detector performance.

The SPM was tested on the FDDB which is a recent benchmark for face detection in unconstrained conditions. It showed good performance on both its discrete and continuous scores. Since SPM is designed for partial face detection, further analysis was conducted through a categorization of the faces in the database according to partial occlusion and evaluating the performance in these categories. The current version handles self-occlusion well with the hands having the lowest performance due to the wide variability in its appearance. For external occlusion, the SPM handles occlusion by other faces and by the upper part of the body which also acts as a context for its own face besides enforcing the faces occluded by shoulders in crowd scenes. The occlusion by other objects can also be modelled in some special applications like for example in sports where the ball is a common object to occlude the players' faces.

A new database was also introduced in this work to bring more attention to the partial face detection problem where all the images in this new POF database have at least one face that is partially occluded. Many of the images include celebrities to show that despite the severe occlusion in some images, these images are still useful for recognition because people can still recognize the celebrities in these images. The SPM was also tested on this new database and the same categorization of partial occlusion types was used to evaluate the method according to its performance in these different categories.

The fine grained evaluation on face detection in the wild is the latest and largest face detection dataset and benchmark. It provides fine grained analysis of the performance with respect to gender, occlusion, glasses, face size, face pose and expression. Detailed analysis about the SPM performance was provided with respect to all of these factors compared to several methods from the academia and the industry.

The face recognition at a distance is the first application visited in the work and the RSFFD method was proposed as a generic post processing step to enhance the performance of any face detector. Experiments were conducted using the VJ face detector as a baseline to test the framework. The BOSS database was used to test the idea by comparing the base face detector performance with and without the score fusion of the RSFFD to illustrate its effect on reducing the false positive. Based on that, the complimentary information of the skin was combined with the SPM framework to enhance its performance.

The human robot interaction is the second application examined in this work where the humanoid robot Nao is used to help in teaching children with autism. The face detection is used to achieve a better natural interaction for the robot with the children. A drawing application was designed as a framework of interaction with the children that supports their educational needs where the face detection and recognition are intended to be used to enhance this interaction.

6.2 Future work

This work tackled the problem of face detection in the wild from a scene understanding point of view by explicitly considering other objects in the face model. The new advances of object detection and scene understanding should be explored to be incorporated further

in the face detection framework. With the current maturity of the face detection algorithms the remaining challenges to be solved in this area are very hard and should be considered from the broader perspective of scene understanding. This work touched this direction and more research need to be conducted along the same lines.

Another good direction for research is how to combine the additional information about the visibility of parts resulting from the SPM into the face recognition framework. This dissertation illustrated that if sunglasses are detected hiding the face then the eyes should not be used for the recognition which is a good point but the remaining question is how to handle such a face in the recognition step. The deeper question is if the detector is providing information about the different part scores and about the pose of the face, how to utilize these information in the recognition. Also in the more general case of a video how to combine these information for the same face detected over different frames to enhance the recognition results.

The human machine interaction in general and the human robot interaction in particular is another important direction that should be investigated. The question is how to utilize the results of a detector like the SPM to combine the previous two points of scene understanding and face recognition to reach better and more natural human robot interaction or human machine interaction.

REFERENCES

[1] Yang, M.; Kriegman, D.; Ahuja, N., “Detecting faces in images: a survey,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.24, no.1, pp.34, 58, Jan 2002.

[2] Zhang, C.; Zhang, Z., “A survey of recent advances in face detection,” Technical report, Microsoft Research, 2010.

[3] Moeslund, T.; Hilton, A.; Krüger, V.; Sigal, L., “Visual Analysis of Humans, Looking at People ” Springer 2011.

[4] Viola, P.; Jones, M., “Robust real-time face detection,” International Journal of Computer Vision, IJCV, 2004.

[5] Li, J.; Zhang, Y., “Learning SURF Cascade for Fast and Accurate Object Detection,” Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, vol., no., pp.3468, 3475, 23-28 June 2013.

[6] Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D., "Object Detection with Discriminatively Trained Part-Based Models," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.32, no.9, pp.1627,1645, Sept. 2010.

[7] Zhu, X.; Ramanan, D., "Face detection, pose estimation, and landmark localization in the wild," Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, vol., no., pp.2879, 2886, 16-21 June 2012.

[8] Orozco, J.; Martinez, B.; Pantic M., "Empirical analysis of cascade deformable models for multi-view face detection," Image Processing (ICIP), 2013 IEEE International Conference Sep. 2013.

[9] Yan, Junjie; Xucong Zhang; Zhen Lei; Dong Yi; Li, S.Z., "Structural models for face detection," Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on , vol., no., pp.1,6, 22-26 April 2013

[10] Freund, Y.; Schapire, R., "A short Introduction to Boosting," Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999.

[11] Viola, P.; Jones, M., "Rapid object detection using a boosted cascade of simple features," Computer Vision and Pattern Recognition (CVPR), 2001 IEEE Conference on, June 2001.

[12] Lienhart, R.; Maydt, J., "An extended set of Haar-like features for rapid object detection," Image Processing. 2002. Proceedings. 2002 International Conference on, vol.1, no., pp.I-900, I-903 vol.1, 2002.

[13] Ojala, T.; Pietikainen, M.; Maenpaa, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.24, no.7, pp.971, 987, Jul 2002.

[14] Zhang, L.; Chu, R.; Xiang, S.; Liao, S.; Li, S.; Lee, S., "Face Detection Based on Multi-Block LBP Representation", Advances in Biometrics, Lecture Notes in Computer Science, Springer 2007.

[15] Li, J.; Wang, T.; Zhang, Y., "Face detection using SURF cascade," Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on , vol., no., pp.2183,2190, 6-13 Nov. 2011.

[16] Zhang, C; Zhang, Z, “Boosting-Based Face Detection and Adaptation” Synthesis Lectures on Computer Vision, Morgan and Claypool 2010

[17] Jain, V.; Learned-Miller, E., "Online domain adaptation of a pre-trained cascade of classifiers," Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, vol., no., pp.577, 584, 20-25 June 2011

[18] Deng, N.; Tian, Y.; Zhang, C., “Support Vector Machines: Optimization based theory, algorithms, and extensions,” CRC Press, 2012.

[19] Chang, C.; Lin, C., “LIBSVM: a library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[20] Joachims, T., “Making large-Scale SVM Learning Practical,” Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[21] Felzenszwalb, P.; and Huttenlocher, D, “Pictorial Structures for Object Recognition,” International Journal of Computer Vision, vol. 61, no. 1, pp. 55-79, 2005.

[22] Yan, Junjie; Zhang, Xucong; Lei, Zhen; Li, Stan Z., "Real-time high performance deformable model for face detection in the wild," Biometrics (ICB), 2013 International Conference on, vol., no., pp.1, 6, 4-7 June 2013.

[23] Mathias, M.; Benenson, R.; Pedersoli, M.; Van Gool, L., “Face detection without bells and whistles,” European Conference on Computer Vision (ECCV), 2014.

[24] Goldmann, L.; Monich, U.J.; Sikora, T., “Components and Their Topology for Robust Face Detection in the Presence of Partial Occlusions,” Information Forensics and Security, IEEE Transactions on, vol.2, no.3, pp.559, 569, Sept. 2007.

[25] L in, Y.; Liu, T.; Fuh, C., “Fast object detection with occlusions,” European Conference on Computer Vision, pp. 402–413, ECCV 2004.

[26] Parris, J.; Wilber, M.; Heflin, B.; Rara, H.; El-Barkouky, A.; Farag, A.; Movellan, J.; Castrilon-Santana, M.; Lorenzo-Navarro, J.; Teli, M.N.; Marcel, S.; Atanasoaei, C.; Boult, T.E., "Face and eye detection on hard datasets," Biometrics (IJCB), 2011 International Joint Conference on , vol., no., pp.1,10, 11-13 Oct. 2011

[27] El-Barkouky, A.; Rara, H.; Farag, A.; Womble, P., “Face detection at a distance using saliency maps,” Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on , vol., no., pp.31,36, 16-21 June 2012.

[28] Mostafa, E.; El-Barkouky, A.; Rara, H.; Farag, A., "Rejecting pseudo-faces using the likelihood of facial features and skin," Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on , vol., no., pp.365,370, 23-27 Sept. 2012.

[29] Li, J.; Levine, M.; An, X.; He, H., “Saliency Detection Based on Frequency and Spatial Domain Analyses,” In The 22nd British Machine Vision Conference (BMVC), 2011.

[30] Niebur, E., “Saliency map,” Scholarpedia, 2(8):2675.

[31] Koch, C.; Ullman, S., “Shifts in selective visual attention: towards the underlying neural circuitry,” Human Neurobiology 4:219-227 (1985).

[32] Hou, X.; Zhang, L., “Saliency detection: A spectral residual approach,” In IEEE Conference on Computer Vision and Pattern Recognition, 2007, CVPR 2007.

[33] Goferman, S.; Zelnik-Manor, L., “Context-aware saliency detection,” In Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2376-2383.

[34] Xingbao, W.; Chunping, L; Gong, L.; Long, L., Shengrong, G., "Pedestrian Recognition Based on Saliency Detection and Kalman Filter Algorithm in Aerial Video," In Seventh International Conference on Computational Intelligence and Security, 2011.

[35] Conaire, C.; O'Connor, N.; Smeaton, A., "Detector adaptation by maximizing agreement between independent data sources," IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum (OTCBVS), 2007.

[36] EL-Barkouky, A.; Shalaby, A.; Mahmoud, A.; Farag, A., "Selective Part Models for detecting partially occluded faces in the wild," the IEEE International Conference on Image Processing, ICIP 2014.

[37] Shengcai, L.; Jain, A.; Li, S., "Partial Face Recognition: Alignment-Free Approach," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.35, no.5, pp.1193, 1205, May 2013.

[38] Dalal, N.; Triggs, B., "Histograms of oriented gradients for human detection," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Conference on , vol.1, no., pp.886,893, June 2005.

[39] Vuong, L.; Jonathan, B.; Zhe, L.; Lubomir, B.; Thomas S. H., "Interactive Facial Feature Localization", ECCV2012.

[40] Belhumeur, P.N.; Jacobs, D.W.; Kriegman, D.; Kumar, N., "Localizing parts of faces using a consensus of exemplars," Computer Vision and Pattern Recognition (CVPR) IEEE Conference on, pp.545, 552, 20-25 June 2011.

[41] Kostinger, M.; Wohlhart, P.; Roth, P.M.; Bischof, H., "Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark

localization," Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on , pp.2144,2151, 6-13 Nov. 2011.

[42] Sagonas, C.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M., "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge," Computer Vision Workshops (ICCV Workshops), IEEE International Conference on, pp.397, 403, 2-8 Dec. 2013.

[43] Roth, M.; Bauml, M.; Nevatia, R.; Stiefelhagen, R., "Robust multi-pose face tracking by multi-stage tracklet association," Pattern Recognition (ICPR), 2012 21st International Conference on, pp.1012, 1016, 11-15 Nov. 2012.

[44] Küblbeck, C.; Ernst, A., "Face detection and tracking in video sequences using the modified census transformation," Image and Vision Computing, Volume 24, Issue 6, Pages 564-572, Elsevier, June 2006.

[45] Baoyuan Wu; Yifan Zhang; Bao-Gang Hu; Qiang Ji, "Constrained Clustering and Its Application to Face Clustering in Videos," Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp.3507, 3514, 23-28 June 2013.

[46] Baoyuan Wu; Siwei Lyu; Bao-Gang Hu; Qiang Ji, "Simultaneous Clustering and Tracklet Linking for Multi-face Tracking in Videos," Computer Vision (ICCV), 2013 IEEE International Conference on, pp.2856, 2863, 1-8 Dec. 2013.

[47] Junjie Yan; Zhen Lei; Longyin Wen; Li, S.Z., "The Fastest Deformable Part Model for Object Detection," Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp.2497, 2504, 23-28 June 2014.

[48] Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D., “Cascade object detection with deformable part models,” Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on , vol., no., pp.2241,2248, 13-18 June 2010.

[49] Ortiz, E.G.; Wright, A.; Shah, M., "Face Recognition in Movie Trailers via Mean Sequence Sparse Representation-Based Classification," Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp.3531, 3538, 23-28 June 2013.

[50] Jain, V.; Learned-Miller, E., “FDDB: A Benchmark for Face Detection in Unconstrained Settings,” Technical Report UM-CS-2010-009, University of Massachusetts, Amherst. 2010.

[51] Li,H.; Hua, G.; Lin, Z.; Brandt, J.; Yang, J., “Probabilistic Elastic Part Model for Unsupervised Face Detector Adaptation.” Computer Vision, 2013 IEEE International Conference on, ICCV 2013.

[52] Shen, X.; Lin, Z.; Brandt, J.; Wu, Y., “Detecting and Aligning Faces by Image Retrieval,” Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, vol., no., pp.3460, 3467, 23-28 June 2013.

[53] Segui, S.; Drozdzal, M.; Radeva, P.; Vitri, J., “An Integrated Approach to Contextual Face Detection,” International Conference on Pattern Recognition Applications and Methods, ICPRAM 2012.

[54] Subburaman, B.; Marcel, S., “Fast Bounding Box Estimation based Face Detection,” European Conference on Computer Vision 2010, Workshop on Face Detection, ECCV 2010.

[55] Mikolajczyk, K.; Schmidt, C.; Zisserman, A., "Human detection based on a probabilistic assembly of robust part detectors," European Conference on Computer Vision, ECCV 2004.

[56] http://www.cbsr.ia.ac.cn/faceEvaluation/

[57] Kalal, Z.; Matas, J.; Mikolajczyk, K., "Weighted sampling for large scale boosting," in British Machine Vision Conference, BMVC 2008.

[58] Baio, J., "Prevalence of Autism Spectrum Disorders: Autism and Developmental Disabilities Monitoring Network, 14 Sites, United States, 2008." Morbidity and Mortality Weekly Report, Surveillance Summaries March 30, 2012.

[59] El-Barkouky, A.; Mahmoud, A.; Graham, J.; Farag, A., "An interactive educational drawing system using a humanoid robot and light polarization," Image Processing (ICIP), 2013 20th IEEE International Conference on, vol., no., pp.3407, 3411, 15-18 Sept. 2013.

CURRICULUM VITA

Name: Ahmed EL-Barkouky

Address: 2232 S. Preston Street Apt3
Louisville, KY, 40217

Contact: (502) 807-4636
ahmed.elbarkouky@louisville.edu

Research Interests

Computer vision and image processing with focus on:
- Object Detection, Tracking and Scene Understanding.
- Robotics vision and Human Robot Interaction.
- Face Detection and Recognition.
- Machine Learning.
- Mobile applications.

Education

Ph.D. in Electrical and Computer Engineering (GPA: 4.0)
University of Louisville,
Louisville, KY, USA.
Aug.'10–Dec.'14
Dissertation: Mathematical Modelling for Partial Object Detection.

MSc in Engineering Mathematics and image processing (Distinction)
Ain Shams University,
Cairo, Egypt.
Feb.'04–Sep.'09
Thesis: Wavelet Transform applications in signal processing.

Diploma in Engineering Mathematics
Ain Shams University,
Cairo, Egypt.
Feb.'03–Feb.'04

BSc in Electrical and communication Engineering (Distinction with honor)
Ain Shams University,
Cairo, Egypt.
Sep.'97–Jul.'02
Project: Vehicle Tracking and Control System through GSM network.

Research Experience

CVIP Lab, **University of Louisville**, KY, USA
Aug.'10-Dec.'14
Research Assistant

- **BOSS Project**: The Biometric Optical Surveillance System is a project in the CVIP lab and EWA Government Systems, Inc. funded by the Department of Homeland Security with a budget of $ 5.2 million for designing the hardware and software of a face recognition system at large distances. The project was recognized in New York Times article August 2013.
  - Developed the face detection module of the system, different stages of the development are published in IJCB 2011, CVPRW 2012 and BTAS 2012.
  - Developed Selective Part Models for the detection and recognition of partially occluded faces, published in ICIP 2014.

- **Autism Robotics Project:** The humanoid robot Nao is used to help in teaching children with autism in collaboration with the Bluegrass autism center. Face recognition is also used here for natural human robot interaction. The project was recognized in the WHAS 11 TV channel and published in ICIP 2013.

- **Driving Assistance Project:** Pedestrian detection from thermal imaging for night driving assistance. The work is also used with the ATRV autonomous robotic vehicle and it is published in IJMT 2014.

Ain Shams University, Cairo, Egypt.
Sep.'03-Jul.'10
Research Assistant

- **Egyptian License Plate Recognition Project:** developed a license plate recognition system that was recognized on Egyptian TV 2009 and published in ICLE 2008.

Teaching Experience

University of Louisville, Ky, USA
Aug.'11-Present
Teaching assistant in the Electrical and Computer Engineering Department.
- DSP, Biometrics, Pattern recognition, Computer vision and Robotics.
- Design of "The UofL Treasure Hunt" robotics competition, fall 2013.

Ain Shams University, Cairo, Egypt
Sep.'03-Jul.'10
Teaching Assistant in the Engineering Mathematics Department.

American University in Cairo, Egypt
Sep.'06-Jul.'10
Teaching Assistant in the Mathematics Department.

Journal Papers

El-Barkouky, A.; Mahmoud, A.; Shalaby, A.; Farag, A., "Partial Face Detection using Multi-Layer Selective Part Models", To be submitted for review in The IEEE Transactions on Image Processing, **TIP 2014**.

Mahmoud, A.; **El-Barkouky, A.**; Graham, J.; Farag, A., "Pedestrian Detection using Thermal Imaging for Night Driving Assistance", International Journal of Multi-media Technology, Vol.4, No.2, pp.26-32, **IJMT 2014**.

Yampolskiy, R.; **EL-Barkouky, A.**, "Wisdom of Artificial Crowds Algorithm for Solving NP-Hard Problems", International Journal of Bio-Inspired Computation, Vol.3, No.6, pp.358-369, **IJBC 2011**.

Conference Papers

El-Barkouky, A.; Shalaby, A.; Mahmoud, A.; Farag, A., "Selective Part Models for Detecting Partially Occluded Faces in the Wild", The IEEE International Conference on Image Processing, **ICIP 2014**.

Mahmoud, A.; **El-Barkouky, A.**; Graham, J.; Farag, A., "Pedestrian Detection Using Mixed Partial Derivative Based Histogram of Oriented Gradients", The IEEE International Conference on Image Processing, **ICIP 2014**.

El-Barkouky, A.; Mahmoud, A.; Graham, J.; Farag, A., "An interactive educational drawing system using a humanoid robot and light polarization", 20th IEEE International Conference on Image Processing, **ICIP 2013**.

Mahmoud, A.; **El-Barkouky, A.;** Farag, H.; Graham, J.; Farag, A., "A Non-invasive Method for Measuring Blood Flow Rate in Superficial Veins from a Single Thermal Image", IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Perception beyond visible spectrum, **CVPR 2013**.

El-Barkouky, A.; Rara, H.; Farag, A.; Womble, P., "Face detection at a distance using saliency maps", IEEE conference on Computer Vision and Pattern Recognition, Workshop on Biometrics, **CVPR 2012**.

Mostafa, E.; **El-Barkouky, A.**; Rara, H.; Farag, A.; , "Rejecting Pseudo-Faces using the Likelihood of Facial Features and Skin", IEEE International Conference on Biometrics: Theory Applications and Systems, **BTAS 2012**.

Parris, J.; Wilber, M.; Heflin, B.; Rara, H.; **El-barkouky, A.**; Farag, A.; et al., "Face and eye detection on hard datasets", IEEE and IAPR International Joint Conference on Biometrics, **IJCB 2011**.

El-Barkouky, A.; El-Ramly, S.; Hassan, M., "Egyptian License Plate Recognition System Using DWT and Template Matching", Proceedings of the Eighth International Conference on Language Engineering, **ICLE 2008**.

Honors and Awards

The Guy Stevenson Award for "Excellence in Graduate Studies": the highest honor in the Ph.D. hooding ceremony selected for addressing the assembly and carrying the graduate school banner leading all Ph.D. graduates in the processional, fall 2014.

Dissertation Completion Fellowship UofL summer/fall 2014.

Outstanding Graduate Student Award UofL 2014.

The Diebold award Speed School of Engineering 2014.

The Theobald scholarship award ECE department 2014.

Research award in the Engineering-Expo UofL 2014.

Outstanding Student Employee Award UofL 2013.

First place award in the graduate research symposium UofL 2013.

Outstanding Teaching Assistant award (selected by students) AinShams Univ. 2008.

Distinction Graduation with Honor AinShams Univ. 2002.

Computer Skills

Programming Languages: Matlab, C/C++, C#.

IDE: Microsoft Visual Studio.

Operating Systems: Microsoft Windows, Linux, Android.

Mathematical Software Tools: Matlab, Maple.

Professional Activities

Reviewer in IEEE Transactions on Information Forensics and Security.

Reviewer in IET Computer Vision.

Grant Writing Academy UofL 2013.

Entrepreneurship Academy UofL 2014.

PLAN workshops UofL 2013/2014.

Student Volunteer at CVPR 2012 and CVPR 2014 conferences.