



CENTER FOR  
MACHINE PERCEPTION



CZECH TECHNICAL  
UNIVERSITY IN PRAGUE

MASTER'S THESIS

ISSN 1213-2365

# Detector of facial landmarks

Michal Uříčář

[uricar.michal@fel.cvut.cz](mailto:uricar.michal@fel.cvut.cz)

CTU-CMP-2011-05

May 12, 2011

Available at  
[http://cmp.felk.cvut.cz/~uricamic/msc/uricamic\\_mt.pdf](http://cmp.felk.cvut.cz/~uricamic/msc/uricamic_mt.pdf)

**Thesis Advisor: Ing. Vojtěch Franc, Ph.D.**

The authors were supported by EC projects FP7-ICT-247525 HUMAVIPS  
and PERG04-GA-2008-239455 SEMISOL.

**Research Reports of CMP, Czech Technical University in Prague, No. 5, 2011**

Published by

Center for Machine Perception, Department of Cybernetics  
Faculty of Electrical Engineering, Czech Technical University  
Technická 2, 166 27 Prague 6, Czech Republic  
fax +420 2 2435 7385, phone +420 2 2435 7637, www: <http://cmp.felk.cvut.cz>



# **Detector of facial landmarks**

Michal Uříčář

May 12, 2011



České vysoké učení technické v Praze  
Fakulta elektrotechnická

Katedra počítačové grafiky a interakce

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Michal Uříčář**

Studijní program: Otevřená informatika (magisterský)  
Obor: Počítačová grafika a interakce

Název tématu: **Detekce významných bodů na lidské tváři**

Pokyny pro vypracování:

Detektor významných bodů na lidské tváři (např. detektor očí, nosu a úst) je důležitou součástí systémů pro rozpoznávání tváří. Cílem diplomové práce je implementovat detektor významných bodů jehož parametry se učí automaticky z anotované databáze obrázků lidských tváří. Problém odhadu pozice významných bodů formulujte jako problém strukturní klasifikace. Pro učení parametrů strukturního klasifikátoru použijte metodu Structured Output Support Vector Machines. Přesnost naučeného detektoru ověřte experimentálně na reálných datech.

Seznam odborné literatury:


Dodá vedoucí práce

Vedoucí: Ing. Vojtěch Franc, Ph.D.

Platnost zadání: do konce letního semestru 2011/2012

  
prof. Ing. Jiří Žára, CSc.  
vedoucí katedry



  
prof. Ing. Boris Šimák, CSc.  
děkan

V Praze dne 13. 12. 2010

## **Acknowledgements**

I would like to thank to my supervisor Ing. Vojtěch Franc, Ph.D. whose suggestions and overall support helped me a lot in my work and prof. Ing. Václav Hlaváč, Csc. who came up with an offer of the thesis topic. I would also like to thank to my family for their support throughout my studies and to my friends who helped me to relax.

## **Declaration**

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.

I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague on May 12, 2011

.....

## Abstract

In this thesis we develop a detector of facial landmarks based on the Deformable Part Models. We treat the task of landmark detection as an instance of the structured output classification problem. We propose to learn the parameters of the detector from data by the Structured Output Support Vector Machines algorithm. In contrast to previous works, the objective function of the learning algorithm is directly related to the performance of the resulting detector which is controlled by a user-defined loss function. The resulting detector is real-time on a standard PC, simple to implement and it can be easily changed for detection of a different set of landmarks. We evaluate performance of the proposed landmark detector on a challenging “Labeled Faces in the Wild” database. The empirical results demonstrate that the proposed detector is consistently more accurate than two public domain implementations based on the Active Appearance Models and the Deformable Part Models. We provide an open source implementation of the proposed detector as well as the algorithm for supervised learning of its parameters from data.

**Keywords:** Facial Landmark Detection, Support Vector Machines, Structured Output Classification, Deformable Part Models



## Resumé

Tato práce navrhuje detektor významných bodů na lidské tváři založený na Deformable Part Models. Na problém detekce významných bodů pohlížíme jako na úlohu strukturální klasifikace. Parametry detektoru jsou učeny z dat pomocí algoritmu Structured Output Support Vector Machines. Na rozdíl od předchozích prací námi používaný algoritmus učení optimalizuje přímo přesnost výsledného detektoru. Algoritmus učení navíc umožňuje snadno měnit statistiku měřící přesnost detektoru pomocí uživatelem definované ztrátové funkce. Výsledný detektor pracuje v reálném čase na standardním PC, je jednoduchý na implementaci a může být snadno modifikován pro detekci jiné množiny významných bodů. Funkčnost navrhovaného detektoru je vyhodnocena na náročné databázi „Labeled Faces in the Wild“. Získané výsledky demonstrují, že navrhovaný detektor dosahuje konzistentně vyšší přesnosti než dvě testované volně dostupné implementace založené na Active Appearance Models a Deformable Part Models. Součástí práce je i open source implementace navrhovaného detektoru a algoritmus pro učení jeho parametrů z anotovaných dat.

**Klíčová slova:** Support Vector Machines, strukturální klasifikace, Deformable Part Models, detekce významných bodů na lidské tváři

# Contents

<b>Abbreviations</b>	<b>5</b>
<b>Symbols</b>	<b>6</b>
<b>1. Introduction</b>	<b>7</b>
<b>2. Related work</b>	<b>9</b>
2.1. Active Appearance Models . . . . .	9
2.2. Deformable Part Models . . . . .	9
<b>3. Proposed detector</b>	<b>11</b>
3.1. Structured output classifier . . . . .	11
3.1.1. Appearance Model . . . . .	13
Normalized image intensity values . . . . .	13
Derivatives of image intensity values . . . . .	14
Local Binary Patterns histogram . . . . .	14
LBP pyramid . . . . .	15
Histogram of Oriented Gradients . . . . .	15
3.1.2. Deformation Cost . . . . .	15
Table representation . . . . .	15
Displacement representation . . . . .	15
3.2. Learning parameters of the structured output classifier . . . . .	16
3.2.1. Bundle Method for Regularized Risk Minimization . . . . .	16
3.2.2. Stochastic Gradient Descent . . . . .	17
<b>4. Experiments</b>	<b>19</b>
4.1. Database: Labeled Faces in the Wild . . . . .	19
4.2. Evaluation procedure . . . . .	20
4.3. Competing methods . . . . .	20
4.3.1. Independently trained binary SVMs detector . . . . .	21
4.3.2. AAM . . . . .	22
IIM Face database . . . . .	23
4.3.3. Oxford detector . . . . .	23
4.4. Comparison of BMRM and SGD . . . . .	28
4.5. Summary results . . . . .	31
<b>5. Implementation</b>	<b>35</b>
<b>6. Conclusions</b>	<b>37</b>
<b>7. Further extensions</b>	<b>39</b>
<b>A. Experimental tuning of the detector configuration</b>	<b>41</b>
A.1. Structured output SVM with table deformation cost . . . . .	41

A.1.1. Parameters . . . . .	41
A.1.2. Results . . . . .	41
A.2. Structured output SVM with displacement deformation cost . . . . .	44
A.2.1. Parameters . . . . .	44
A.2.2. Results . . . . .	44
A.3. Modification of $s_0$ . . . . .	47
A.3.1. Parameters . . . . .	47
A.3.2. Results . . . . .	47
A.4. Features: Normalized image intensity values . . . . .	51
A.4.1. Parameters . . . . .	51
A.4.2. Results . . . . .	51
A.5. Features: Derivatives of image intensity values . . . . .	54
A.5.1. Parameters . . . . .	54
A.5.2. Results . . . . .	54
A.6. Features: LBP histogram . . . . .	57
A.6.1. Parameters . . . . .	57
A.6.2. Results . . . . .	57
A.7. Features: HOG . . . . .	60
A.7.1. Parameters . . . . .	60
A.7.2. Results . . . . .	60
A.8. Summary of all experiments . . . . .	63
<b>B. CD Contents</b>	<b>65</b>
<b>Bibliography</b>	<b>67</b>

## List of Figures

1.1. Example of use of detector . . . . .	7
3.1. Landmarks & components . . . . .	12
3.2. Admissible positions of each component . . . . .	13
3.3. LBP computation scheme . . . . .	14
4.1. Examples from the LFW face database . . . . .	20
4.2. Error normalization scheme. . . . .	21
4.3. Aquisition of the positive and negative examples for binary SVM . . . . .	22
4.4. Example of detection made by the AAM detector on the LFW database . . . . .	23
4.5. Some annotated examples of the IIM face database . . . . .	26
4.6. Output of the Oxford's detector. . . . .	27
4.7. Comparison of the BMRM and SGD . . . . .	29
4.8. Comparison of the BMRM and SGD . . . . .	30
4.9. Summary — Cumulative histograms . . . . .	33
7.1. Modification of the deformable part model. . . . .	39
A.1. Image results for experiment: structured output SVM with the table deformation cost . . . . .	43
A.2. Cumulative histograms— structured output SVM with the table deformation cost. 43	
A.3. Image results for experiment: structured output SVM with the displacement deformation cost . . . . .	45
A.4. Cumulative histograms— structured output SVM with the displacement deformation cost. . . . .	46
A.5. Definition of the center of the face . . . . .	47
A.6. Image results for experiment: modification of $s_0$ . . . . .	48
A.7. The comparison of image results . . . . .	49
A.8. Cumulative histograms— modification of $s_0$ . . . . .	49
A.9. Image results for experiment: normalized image intensity values features . . . . .	52
A.10. Cumulative histograms— normalized image intensity values features. . . . .	52
A.11. Image results for experiment: derivatives of image intensity values features . . . . .	55
A.12. Cumulative histograms— derivatives of image intensity values features . . . . .	55
A.13. Image results for experiment: LBP histogram features . . . . .	58
A.14. Cumulative histograms— LBP histogram features . . . . .	58
A.15. Image results for experiment: HOG features . . . . .	61
A.16. Cumulative histograms— HOG features . . . . .	61
A.17. Summary — Cumulative histograms . . . . .	63

## List of Tables

4.1.	Partitioning of the LFW database into training, validation and testing set. . . . .	20
4.2.	Overall number of annotated points in both face databases. . . . .	23
4.3.	Comparison of the BMRM and SGD. . . . .	28
4.4.	Detail around 10% of relative error . . . . .	31
4.5.	Summary of mean errors . . . . .	31
4.6.	Summary of maximal errors . . . . .	32
A.1.	Parameters settings for experiment: structured output SVM with the table deformation cost . . . . .	41
A.2.	Results of validation of the experiment: structured output SVM with the table deformation cost . . . . .	42
A.3.	Normalized errors of the experiment: structured output SVM with the table deformation cost . . . . .	42
A.4.	Parameters settings for experiment: structured output SVM with the displacement deformation cost . . . . .	44
A.5.	Results of validation of the experiment: structured output SVM with the displacement deformation cost . . . . .	44
A.6.	Normalized errors of the experiment: structured output SVM with the displacement deformation cost . . . . .	45
A.7.	Parameters settings for experiment: modification of $s_0$ . . . . .	47
A.8.	Results of validation of the experiment: modification of $s_0$ . . . . .	48
A.9.	Normalized errors of the experiment: modification of $s_0$ . . . . .	50
A.10.	Parameters settings for experiment: normalized image intensity values features	51
A.11.	Results of validation of the experiment: normalized image intensity values features . . . . .	51
A.12.	Normalized errors of the experiment: normalized image intensity values features	53
A.13.	Parameters settings for experiment: normalized image intensity values features	54
A.14.	Results of validation of the experiment: derivatives of image intensity values features . . . . .	54
A.15.	Normalized errors of the experiment: derivatives of image intensity values features . . . . .	56
A.16.	Parameters settings for experiment: LBP histogram features . . . . .	57
A.17.	Results of validation of the experiment: LBP histogram features . . . . .	57
A.18.	Normalized errors of the experiment: LBP histogram features . . . . .	59
A.19.	Parameters settings for experiment: HOG features . . . . .	60
A.20.	Results of validation of the experiment: HOG features . . . . .	60
A.21.	Normalized errors of the experiment: HOG features . . . . .	62
A.22.	Detail around 10% of relative error . . . . .	63
A.23.	Summary of mean errors . . . . .	64
A.24.	Summary of maximal errors . . . . .	64

## Abbreviations

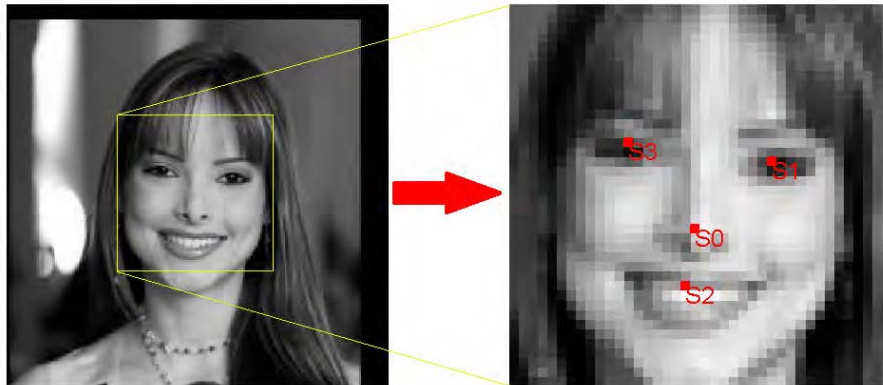
AAM	Active Appearance Models
AB	AdaBoost
BMRM	Bundle Method for Regularized Risk Minimization
DP	Dynamic Programming
DPM	Deformable Part Models
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
LFW	Labeled faces in the wild
LIBOCAS	Library implementing OCAS solver for training linear SVM classifier from large-scale data
OCAS	Optimized Cutting Plane Algorithm for Support Vector Machines
PCA	Principle Component Analysis
SGD	Stochastic Gradient Descent
SO-SVM	Structured Output SVM
SVM	Support Vector Machines
TRN	Training set
TST	Testing set
VAL	Validation set

# Symbols

$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of positive real numbers (without 0)
$\mathbf{w}$	Vector $\mathbf{w}$
$\langle \mathbf{x}, \mathbf{x}' \rangle$	Dot product between $\mathbf{x}$ and $\mathbf{x}'$
$\  \cdot \ $	Euclidean norm, $\ \mathbf{x}\  = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$
$\mu$	Mean value, $\mu = \frac{1}{n} \sum_1^n x_i$
$\sigma$	Standard deviation, $\sigma = \sqrt{\frac{1}{n-1} \sum_1^n (x_i - \mu)^2}$
$R(\cdot)$	Risk function, e.g. $R(\mathbf{w})$ is a risk of joint parameter vector $\mathbf{w}$
$\mathcal{O}(\cdot)$	Big O notation; Asymptotically worst case of running time in algorithms analysis
$I$	Image, i.e. 2D matrix of dimension $H \times W$
$\mathcal{S}_i$	Set of all admissible positions of the $i$ -th landmark within $I$ , $\mathcal{S}_i \in \{1, \dots, H\} \times \{1, \dots, W\}$
$f(I, \mathbf{s})$	Scoring function, $f : \mathcal{I} \times \mathcal{S} \rightarrow \mathbb{R}$
$L(\mathbf{y}, \mathbf{y}')$	Loss (Penalty) function

# 1. Introduction

This master thesis deals with the problem of automatic detection of facial landmarks like centers (or corners) of eyes, nose and mouth. Functionality of the landmark detector is illustrated by Figure 1.1.



**Figure 1.1.** Given a face image along with a rough estimate of the face bounding box (yellow), the landmark detector estimates positions of a set of facial landmarks like centers of eyes, nose and mouth (red points  $S_0, \dots, S_3$ ).

The detection of facial landmarks is an essential part of many face recognition systems. Accuracy and speed of the landmark detection significantly influences final performance of the face recognition system [Beumer and Veldhuis, 2005], [Cristinacce et al., 2004], [Riopka and Boulton, 2003].

The problem of detecting facial landmarks is largely considered to be a solved scientific problem. There exists several successful commercial solutions like the OKAO Vision Facial Feature Extraction API [OMRON, 2011] which is used for example in Picasa™ or Apple iPhoto software. On the other hand, open source implementations of acceptable quality are scarce. The goal of this thesis is to fill this gap by developing high performance open source implementation available for academic use.

In this thesis we develop a landmark detector based on the Deformable Part Models (DPM) [Fischler and Elschlager, 1973]. We treat the landmark detector as an instance of the structured output classifier whose accuracy is measured by a user-defined loss function. We propose to learn parameters of the detector from data by the Structured Output Support Vector Machines algorithm [Tsochantaridis et al., 2005]. In contrast to existing approaches which learn the detector in two independent stages, objective function of our learning algorithm is directly related to the performance of the resulting detector via clearly specified loss function. The novelty of our approach is not in using the deformable part models for the landmark detection but in using a principled approach to learn the parameters of the detector from data. The proposed landmark detector is real-time on a standard PC, simple to implement and it can be easily used to detect different sets of landmarks. We evaluate performance of the proposed detector on a challenging “Labeled Faces in the Wild” database. The experimental results demonstrate



## 1. Introduction

that the proposed landmark detector consistently outperforms two public domain implementations based on the Active Appearance Models [Kroon, 2010] and the Deformable Part Models [Sivic et al., 2009]. We would like to point out that especially the letter landmark detector was a strong competitor which had been previously used in a number of successful face recognition projects [Everingham et al., 2006], [Everingham et al., 2009], [Sivic et al., 2009].

The main contributions of this thesis are as follows:

1. We treat the landmark detection with the Deformable Part Model as an instance of the structured output classification problem whose detection accuracy is measured by a user-defined loss function.
2. We propose to use the Structured Output Support Vector Machines for supervised learning of the parameters of the landmark detector from data.
3. We empirically evaluate accuracy of the proposed landmark detector on a challenging “Labeled Faces in the Wild” database. The results show that the proposed detector consistently outperforms a baseline “unstructured” SVM detector and two public domain landmark detectors based on the Active Appearance Models and the Deformable Part Models.
4. We provide an empirical comparison of two optimization algorithms — the Bundle Method for Regularized Risk Minimization [Teo et al., 2010] and the Stochastic Gradient Descent [Bordes et al., 2009] — which are suitable for solving the convex optimization problem emerging in the Structured Output SVM learning.
5. We provide an open source implementation of the proposed landmark detector as well as the algorithm for supervised learning of its parameters from data.

The text of the thesis is organized as follows:

**Chapter 2 Related work** Gives a brief description of two approaches which are most frequently used for the detection of facial landmarks. Namely, the detectors based on the Active Appearance Models and the Deformable Part Models are outlined.

**Chapter 3 Proposed Detector** Describes the proposed landmark detector based on the Deformable Part Models and the algorithm for supervised learning of its parameters from data.

**Chapter 4 Experiments** Provides experimental evaluation of the proposed landmark detector and its comparison to one baseline approach and two public domain implementations. In addition, two solvers for the Structured Output SVM learning are also compared.

**Chapter 5 Implementation** Gives a brief description of the open source library **flandmark** implementing the proposed detector and the learning algorithm.

**Chapter 6 Conclusions** Gives the conclusions.

**Chapter 7 Further extensions** Provides ideas for further extension on the detector.

**Appendix A Tuning the model configuration** Describes the experiments done in order to tune the optimal configuration of the proposed detector.

**Appendix B CD Contents** Describes the content of the enclosed CD.

## 2. Related work

In this chapter we give a brief description of two approaches which are most frequently used for the detection of facial landmarks. First, in Section 2.1, we describe the Active Appearance Model which we use as one of the competing method in the empirical evaluation. Second, in Section 2.2, we outline the Deformable Part Models on which we build our own landmark detector.

### 2.1. Active Appearance Models

Among the most popular are the detectors based on the Active Appearance Models (AAM) [Cootes et al., 2001]. This method uses joint statistical model of a shape and appearance. Detectors build on AAM provide a dense set of facial features. In turn, whole contours of facial parts like eyes, nose or mouth can be extracted from the response of the AAM detector. On the other AAM have several drawbacks. First, AAM require high resolution images. Second, annotation of training data is very costly. Third, the detection leads to a non-convex optimization problem susceptible to local optima unless a good initial guess of the landmark positions is available.

AAM rely on the statistical model of shape and appearance. The shape is captured by a finite number of points which define contours of the object. The appearance is a texture (i.e. pixel-based pattern of intensities or colors across an image patch) captured by sampling a suitable image warping function. AAM normalize the aligning contours w.r.t. position, orientation and scale using a Procrustes analysis into a “shape-free patch”. The appearance (or texture) is normalized by removing the linear global illumination effects by standardization. Finally, the Principal Component Analysis (PCA) is performed on both shape and texture to achieve a constrained and compact description.

In the test time, the parameters of the AAM are tuned in order to generate a synthetic image from the AAM which best matches the input image. This process leads to a non-convex optimization problem. [Cootes et al., 2001] proposed a scheme to solve this optimization problem efficiently. In brief, the iterative model refinement procedure projects the texture sample into the texture frame and evaluates the error vector which is then used for computation of the predicted displacements. Then the model parameters are updated and the projection error vector computation is repeated until the fit error is less than the current one. In practice the coarse-to-fine approach is used which applies the iterative procedure in different scales.

### 2.2. Deformable Part Models

A straightforward approach to landmark detection is based on using independently trained detectors for each facial landmark. For instance, the AdaBoost based detector and its modifications have been frequently used [Viola and Jones, 2004]. If applied independently, the individual detectors often fail to provide a robust estimate of the landmark positions. The weakness of the local evidence can be compensated by using a prior on the geometrical configuration of the landmarks. The detection is then carried out in two consecutive steps. In the first step the individual detectors are used to find a set of candidate positions for each landmark separately. In the second step the best landmark configuration with the highest support from the

## 2. Related work

geometrical prior is selected. The landmark detectors based on this approach were proposed for example in [Beumer et al., 2006], [Cristinacce and Cootes, 2003], [Erukhimov and Lee, 2008], [Wu and Trivedi, 2005].

The Deformable Part Models (DPM) [Fischler and Elschlager, 1973], [Crandall et al., 2005], [Felzenszwalb and Huttenlocher, 2005], [Felzenszwalb et al., 2009], go one step further by fusing the local appearance model and the geometrical constraint into a single model. The DPM is given by set of parts along with a set of connections between certain pairs of parts arranged in a deformable configuration. A natural way to describe the DPM is an undirected graph with vertices corresponding to the parts and edges representing the pairs of connected parts. The DPM based detector estimates all landmark positions simultaneously by optimizing a single cost function composed of a local appearance model and a deformation cost. The complexity of finding the best landmark configuration depends on the structure of the underlying graph. If the graph does not contain loops, e.g. it has star-like structure with the central node corresponding to the nose, the estimation can be solved efficiently by a variant of the Dynamic Programming.

An instance of finely tuned facial landmark detector based on the DPM has been proposed in [Everingham et al., 2006]. The very same detector was also used in several successful face recognition systems described in [Everingham et al., 2009] and [Sivic et al., 2009]. This landmark detector is publicly available and we use it for comparison with our detector. In this case, the local appearance model is learned by a multiple-instance variant of the AdaBoost algorithm with the Haar-like features used as the weak classifiers. The deformation cost is expressed as a mixture of Gaussian trees. Importantly, learning of the local appearance model and the deformation cost is done in two independent steps which simplifies the problem but may not be the optimal solution. In contrast, we propose to learn the parameters in one step by directly optimizing accuracy of the resulting detector.

### 3. Proposed detector

In this chapter we describe the proposed detector of the facial landmarks and an algorithm for supervised learning of the parameters of the detector from data. The chapter is split into two main sections. First, in Section 3.1, we describe the model of the detector which we treat as an instance of the structured output classifier based on the Deformable Part Models. We also describe several instances of the detector which use different local appearance models and the deformation costs. Second, in Section 3.2, we formulate the problem of learning the parameters of the detector based on the Structured Output SVM algorithm (SO-SVM). We also describe two optimization methods which are suitable for optimization of large-scale instances of the convex problem emerging in the SO-SVM learning.

#### 3.1. Structured output classifier

We assume that the input of our classifier is a still image of a fixed size which contains a single face. We denote this input image as a **normalized image frame**. The normalized image frame is constructed as follows. First, the face bounding box is estimated by a face detector (e.g. we use a commercial implementation of the AdaBoost face detector [Viola and Jones, 2004]<sup>1</sup>). Second, the face box is enlarged by a certain margin to ensure that the whole face is contained. Third, the face image is cropped according to the enlarged face box and its size is normalized.

Let  $I \in \mathcal{I} = \mathcal{X}^{H \times W}$  be an input image and let  $\mathcal{S}_i \subset \{1, \dots, H\} \times \{1, \dots, W\}$  denote a set of all admissible positions of the  $i$ -th landmark within the image  $I$ . The symbol  $\mathcal{X}$  denotes a set of pixel values which in our experiments, dealing with 8bit gray-scale images, equals to  $\{0, \dots, 255\}$ . Each landmark is defined by certain region that surrounds it, i.e. bounding box around the landmark. We refer to this region as the **component** (see Figure 3.1). The set of all configurations of  $M$  landmarks is denoted by  $\mathcal{S} = \mathcal{S}_0 \times \dots \times \mathcal{S}_{M-1}$ . The quality of a landmark configuration  $\mathbf{s} = (s_0, \dots, s_{M-1})$  given an image  $I$  is measured by a scoring function  $f: \mathcal{I} \times \mathcal{S} \rightarrow \mathbb{R}$  defined as

$$f(I, \mathbf{s}) = \sum_{i=0}^{M-1} q_i(I, \mathbf{s}_i) + \sum_{i=1}^{M-1} g_i(\mathbf{s}_0, \mathbf{s}_i) \quad (3.1)$$

The first term in (3.1) corresponds to a local appearance model which evaluates how well landmarks on positions  $\mathbf{s}$  match with the input image  $I$ . The second term in (3.1) corresponds to the deformation cost which evaluates the relative positions of the landmarks with respect to the anchor position  $\mathbf{s}_0$ . In particular, we use the nose as the anchor landmark. We assume that the costs  $q_i: \mathcal{I} \times \mathcal{S}_i \rightarrow \mathbb{R}$ ,  $i = 0, \dots, M-1$  and  $g_i: \mathcal{S}_0 \times \mathcal{S}_i \rightarrow \mathbb{R}$ ,  $i = 0, \dots, M-1$  are linearly parametrized functions

$$q_i(I, \mathbf{s}_i) = \langle \mathbf{w}_i^q, \Psi_i^q(I, \mathbf{s}_i) \rangle \quad (3.2)$$

$$g_i(\mathbf{s}_0, \mathbf{s}_i) = \langle \mathbf{w}_i^g, \Psi_i^g(\mathbf{s}_0, \mathbf{s}_i) \rangle \quad (3.3)$$

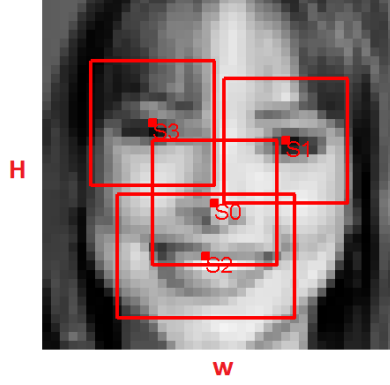
where  $\Psi_i^q: \mathcal{I} \times \mathcal{S}_i \rightarrow \mathbb{R}^{n_{iq}}$ ,  $\Psi_i^g: \mathcal{S}_0 \times \mathcal{S}_i \rightarrow \mathbb{R}^{n_{ig}}$ ,  $i = 0, \dots, M-1$  are predefined maps and  $\mathbf{w}_i^q \in \mathbb{R}^{n_{iq}}$ ,  $\mathbf{w}_i^g \in \mathbb{R}^{n_{ig}}$ ,  $i = 0, \dots, M-1$  are parameter vectors which will be learned

<sup>1</sup>The face detector was provided by courtesy of the Eyedea Recognition s.r.o. (<http://www.eyedea.cz>)

### 3. Proposed detector

from examples. Let us introduce a joint map  $\Psi: \mathcal{I} \times \mathcal{S} \rightarrow \mathbb{R}^n$  and a joint parameter vector  $\mathbf{w} \in \mathbb{R}^n$  defined as a column-wise concatenation of the individual maps  $\Psi_i^q, \Psi_i^g$  and the individual parameter vectors  $\mathbf{w}_i^q, \mathbf{w}_i^g$ , respectively. With these definitions we see that the cost function (3.1) simplifies to

$$f(I, \mathbf{s}) = \langle \mathbf{w}, \Psi(I, \mathbf{s}) \rangle. \quad (3.4)$$



**Figure 3.1.** Our configuration of Landmarks & components depicted in the normalized image frame. Note that this is not the only possible configuration. Both size of the components and the number of landmarks may be modified. Sizes of components were determined experimentally, the number of landmarks corresponds with the annotation of the available face database.

Given an input image  $I$ , the structured output classifier outputs the configurations  $\hat{\mathbf{s}}$  computed by maximizing the cost function  $f(I, \mathbf{s})$ , i.e.,

$$\begin{aligned} \hat{\mathbf{s}} &\in \arg \max_{\mathbf{s} \in \mathcal{S}} f(I, \mathbf{s}) \\ &= \arg \max_{\mathbf{s}_0 \in \mathcal{S}_0} \left[ q_0(I, \mathbf{s}_0) + \sum_{i=1}^{M-1} \max_{\mathbf{s}_i \in \mathcal{S}_i} \left( q_i(I, \mathbf{s}_i) + g_i(\mathbf{s}_0, \mathbf{s}_i) \right) \right] \end{aligned} \quad (3.5)$$

The star like structure of the max-sum problem (3.5) allows to solve the classification problem efficiently by dynamic programming (DP). The way how to organize the DP algorithm is apparent directly from (3.5).

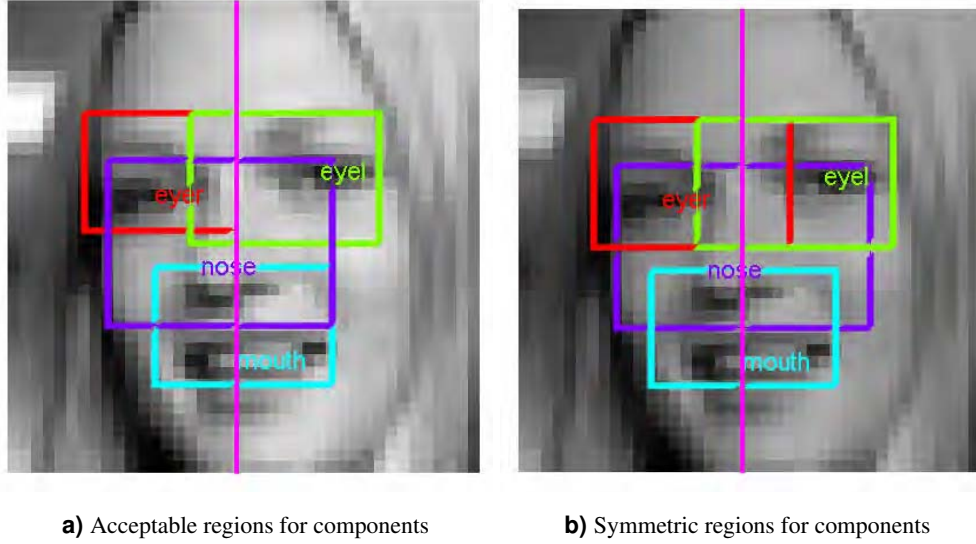
A complete specification of the structured classifier (3.5) requires to define:

- The fixed maps  $\Psi_i^q(I, \mathbf{s}_i), i = 0, \dots, M-1$ , which define a feature description of a rectangle cropped around the position  $\mathbf{s}_i$ , i.e.,  $\Psi_i^q(I, \mathbf{s}_i)$  is the feature description of the  $i$ -th component. The size of the rectangular component and the particular feature descriptor are crucial design options which have to be made carefully. In Section 3.1.1, we describe list of feature descriptors we have considered. Results of the experimental tuning of the best configuration of the components size and the feature descriptor are provided in Appendix A.
- The fixed maps  $\Psi_i^g(\mathbf{s}_0, \mathbf{s}_i), i = 0, \dots, M-1$ , which define parametrization of the deformation cost. Section 3.1.2 describes the parametrization which we considered. The selection of the best parametrization is done experimentally. These experiments are described in Appendix A.
- The set  $\mathbf{S} = (\mathbf{s}_0 \times \dots \times \mathbf{s}_{M-1})$  which defines the search space of the landmark positions. These sets can be interpreted as hard constraints on the admissible configurations of the landmarks, i.e. the landmark positions outside these sets correspond to  $\infty$  value of the deformation cost. The optimal setting of these sets is selected experimentally (more details

are in Section 4). Figures 3.2a and 3.2b visualize the found optimal search spaces for each component.

- The joint parameter vector  $w \in \mathbb{R}^n$  which is learned from training examples by the structured output SVMs described in Section 3.2.

Note, that the set of four landmarks and their star-like structure used in this thesis is only one option. The approach proposed here can be readily applied for different sets of landmarks as well as different structural constraints. Of course, the particular choice must be done carefully in order to keep the inference problem (3.1) efficiently solvable.



**Figure 3.2.** We put hard constraints on admissible positions of each component by restricting the search space for each component to a certain region. Hard constraints are estimated from training examples by computing bounding boxes of all positions of the respective landmark. Finally, the search regions are made vertically symmetric.

### 3.1.1. Appearance Model

We tried several features for the appearance model  $q_i(I, s_i)$  and we summarize them in this section. The best features were found experimentally. The corresponding experiments are summarized in Appendix A.

#### Normalized image intensity values

Among the simplest features are the normalized image intensity values. We generate the feature map  $\Psi_i^q(I, s_i)$  as a concatenation of the normalized image intensity values  $\bar{x}$  and its element wise square  $\bar{x}^2$ . The normalized image intensity values are defined as

$$x_i = \frac{x_i - \mu}{\sigma} \quad (3.6)$$

where  $x_i$  is the  $i$ -th component of  $\bar{x}$ ,  $\mu$  is the mean and  $\sigma$  is the standard deviation of the intensities in  $\bar{x}$ .

Experimental evaluation of this feature is given in Appendix A.4.

### Derivatives of image intensity values

Other simple features, which can be easily used in combination with the normalized image intensity values are derivatives of image intensity values. We compute directional derivatives (column-wise and row-wise) as the difference of consecutive columns (and rows), i.e. we are using the uncentered discrete derivative masks  $([-1, 1])$ .

We generate the feature map  $\Psi_i^q(I, s_i)$  as a concatenation of the normalized image intensity values as defined above (with the square of the normalized image intensity values also) and a concatenation of square of column-wise derivatives  $c$  and square of row-wise derivatives  $r$ . So the final feature vector is defined as

$$\Psi_i^q(I, s_i) = \begin{bmatrix} \bar{x} \\ \bar{x}^2 \\ c^2 \\ r^2 \end{bmatrix} \quad (3.7)$$

where the squares are computed component-wise. The squares of derivatives are used because otherwise the derivatives are a linear combination of the normalized image intensity values.

Experimental evaluation of this feature is given in Appendix A.5.

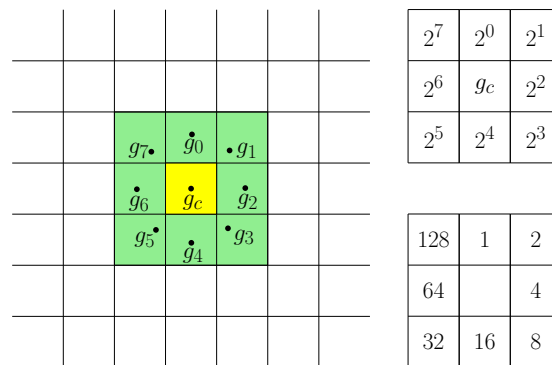
### Local Binary Patterns histogram

The Local Binary Patterns (LBP) have been successfully used in many face recognition problems [Ahonen et al., 2004]. It can be used in a form of the histogram or directly as described in the next section. The LBP number that characterizes the spatial structure of the local image texture [Heikkilä et al., 2009] [Matas et al., 2010, lecture 1–3] is defined as

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(x)2^p, \quad x = g_p - g_c, \text{ where} \quad (3.8)$$

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (3.9)$$

where  $g_c, g_p$  are image intensity values as depicted in Figure 3.3.  $P$  defines neighbourhood and  $R$  is the spatial resolution. We generate the feature map  $\Psi_i^q(I, s_i)$  by computing LBP features of  $i$ -th component (with  $P = 8$  and  $R = 1.0$ ), then the histogram of LBP values is made and the resulting vector is normalized to a unit size.



**Figure 3.3.** LBP computation scheme. The yellow box shows the center pixel, i.e. the pixel for which LBP number is computed. The green boxes are 8-neighbourhood for the center pixel.

Experimental evaluation of this feature is given in Appendix A.6.

## LBP pyramid

Instead of the histogram of the LBP features we can use LBP features directly (i.e. binary encoded LBP features as defined in equation (3.8)) in form of the LBP pyramid. That is, the feature description is a concatenation of binary encoded LBP numbers computed in several scales [Franc and Sonnenburg, 2010]. In particular, we use LBPs computed in 4 scales starting from the original image and consequently downscaling the image 3 times by  $\frac{1}{2}$ . The resulting feature vector is very sparse which is exploited by the learning algorithm as well as during the classification.

Experimental evaluation of this feature is given in Appendix A.1, A.2, A.3 and Section 4.4

## Histogram of Oriented Gradients

Another option for generating the feature map  $\Psi_i^g(I, \mathbf{s}_i)$  is the usage of the Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005]. The computation of HOG features goes as follows. The first step computes image derivatives. Following the original paper we use centered discrete derivative masks ( $[-1, 0, 1]$ ,  $[-1, 0, 1]^T$ ) without previous Gaussian smoothing. The second step is spatial/orientation binning— in this step each pixel calculates a weighted vote for an edge orientation histogram channel. We use 9 bins histograms. The votes are bilinearly interpolated between neighbouring bin centers and accumulated into spatial regions called *cells*. We use rectangular cells of size  $2 \times 2$  pixels. The last step is block normalization and feature descriptor generation— in this step we normalize the histograms accumulated in cells that are contained in *blocks*. Blocks are overlapping so that each cell contributes to several blocks. We use blocks of size covering  $2 \times 2$  cells. The block overlap is set to be half of their size.

Experimental evaluation of this feature is given in Appendix A.7.

### 3.1.2. Deformation Cost

We consider two parametrizations of the deformation cost  $g_i(\mathbf{s}_0, \mathbf{s}_i)$ . Namely, we represent the cost as a table and as a quadratic function of a displacement vector between landmark positions.

#### Table representation

If no prior knowledge is available, the deformation cost  $g_i(\mathbf{s}_0, \mathbf{s}_i)$  can be represented by a table whose elements specify cost for each combination of  $\mathbf{s}_0$  and  $\mathbf{s}_i$  separately. In this case  $\Psi^g(\mathbf{s}_0, \mathbf{s}_i)$  is a sparse vector with all elements zero but the element corresponding to the combinations  $(\mathbf{s}_0, \mathbf{s}_i)$  which is one.

Representation of the deformation cost by a table is the most flexible way (least prior on the configuration) and it is easy to implement. On the other hand, it is given by a large number of parameters which, in turn, requires a large number of the training examples in order to avoid over-fitting. In fact, each combination  $(\mathbf{s}_0, \mathbf{s}_i)$  should be present in training examples at least once in order to make the corresponding cost non-zero.

#### Displacement representation

Another option to define the cost  $g_i(\mathbf{s}_0, \mathbf{s}_i)$  is to consider its value to be a function of a displacement vector  $\mathbf{s}_i - \mathbf{s}_0$ . Following [Felzenszwalb et al., 2009], we define the deformation cost as

$$\Psi_i^g(\mathbf{s}_0, \mathbf{s}_i) = \left. \begin{aligned} (dx, dy, dx^2, dy^2) \\ (dx, dy) = (x_i, y_i) - (x_0, y_0) \end{aligned} \right\} \quad (3.10)$$



### 3. Proposed detector

This representation accounts for the distance and the direction of the  $i$ -th landmark  $\mathbf{s}_i$  with respect to the anchor landmark  $\mathbf{s}_0$ . This representation is given only by four free parameters which substantially reduces the risk of over-fitting.

## 3.2. Learning parameters of the structured output classifier

We learn the joint parameter vector  $\mathbf{w}$  by the Structured Output SVMs (SO-SVM) algorithm proposed in [Tsochantaridis et al., 2005]. The requirements on the classifier are specified by a user-defined loss function  $L: \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$ . The value  $L(\mathbf{s}, \mathbf{s}^*)$  penalizes the classifier estimate  $\mathbf{s}$  provided the actual configuration of the landmarks is  $\mathbf{s}^*$ . The SO-SVM requires loss function to be non-negative and zero only if the estimate is absolutely correct, i.e.  $L(\mathbf{s}, \mathbf{s}') \geq 0, \forall \mathbf{s}, \mathbf{s}' \in \mathcal{S}$ , and  $L(\mathbf{s}, \mathbf{s}') = 0$  iff  $\mathbf{s} = \mathbf{s}'$ . In particular, we use the mean deviation of the estimated and the ground truth positions as the loss function, i.e.,

$$L(\mathbf{s}, \mathbf{s}^*) = \frac{1}{M} \sum_{j=0}^{M-1} \|\mathbf{s}_j - \mathbf{s}_j^*\|. \quad (3.11)$$

However, any other loss function meeting the constraints defined above can be readily used.

Given a set of training examples  $\{(I^1, \mathbf{s}^1), \dots, (I^m, \mathbf{s}^m)\} \in (\mathcal{I} \times \mathcal{S})^m$  the parameter  $\mathbf{w}$  is obtained by solving the following convex minimization problem

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^n} \left[ \frac{\lambda}{2} \|\mathbf{w}\|^2 + R(\mathbf{w}) \right] \quad (3.12)$$

where

$$R(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \max_{\mathbf{s} \in \mathcal{S}} \left( L(\mathbf{s}^i, \mathbf{s}) + \langle \mathbf{w}, \Psi(I^i, \mathbf{s}) \rangle \right) - \frac{1}{m} \sum_{i=1}^m \langle \mathbf{w}, \Psi(I^i, \mathbf{s}^i) \rangle. \quad (3.13)$$

The number  $\lambda \in \mathbb{R}^+$  is a regularization constant whose optimal value is tuned on a validation set.  $R(\mathbf{w})$  is a convex upper bound on the empirical risk which is the average of the loss  $L$  computed over the training examples.

We consider two different optimization algorithms for solving the problem (3.12), namely, the Bundle Method for Regularized Risk Minimization (BMRM) and the Stochastic Gradient Descent (SGD). The algorithms are shortly described in the following two sections. Experiments comparing their performance on learning our landmark detector are presented in Section 4.4.

### 3.2.1. Bundle Method for Regularized Risk Minimization

Bundle Method for Regularized Risk Minimization (BMRM) is a generic method for minimization of regularized convex functions proposed in [Teo et al., 2010]. This method is guaranteed to find  $\epsilon$ -precise solution in  $\mathcal{O}(1/\epsilon)$  iterations. The BMRM requires a procedure which for given  $\mathbf{w}$  returns value of the risk  $R(\mathbf{w})$  and its sub-gradient  $R'(\mathbf{w})$ . In our case, the sub-gradient  $R'(\mathbf{w})$  is given by

$$R'(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \left( \Psi(I^i, \hat{\mathbf{s}}^i) - \Psi(I^i, \mathbf{s}^i) \right) \quad (3.14)$$

where

$$\hat{\mathbf{s}}^i = \arg \max_{\mathbf{s} \in \mathcal{S}} \left[ L(\mathbf{s}^i, \mathbf{s}) + \langle \mathbf{w}, \Psi(I^i, \mathbf{s}) \rangle \right]. \quad (3.15)$$

Note that evaluation of  $R(\mathbf{w})$  and  $R'(\mathbf{w})$  is dominated by computation of the scalar products  $\langle \mathbf{w}, \Psi(I^i, \mathbf{s}) \rangle, i = 1, \dots, m, \mathbf{s} \in \mathcal{S}$ , which, fortunately, can be efficiently parallelized.

### 3.2.2. Stochastic Gradient Descent

Another method that can be used to solve (3.12) is the Stochastic Gradient Descent (SGD). We use the modification proposed in [Bordes et al., 2009] which uses two neat tricks. Starting from an initial guess  $\mathbf{w}_0$ , the SGD algorithm iteratively changes  $\mathbf{w}$  by applying the following update rule:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\lambda^{-1}}{t_0 + t} g_t, \text{ where} \quad (3.16)$$

$$g_t = \lambda \mathbf{w}_t + R'_t(\mathbf{w}) \quad (3.17)$$

$\lambda$  is a regularization constant and  $t_0$  is a constant and  $t$  is the number of the iteration. The SGD implementation proposed in [Bordes et al., 2009] tunes the optimal value of  $t_0$  on a small portion of training examples subsampled from training set. The sub-gradient is computed in almost the same manner as in (3.14), but only for one training image at a time, i.e.,

$$R'_t(\mathbf{w}) = \Psi(I^t, \hat{\mathbf{s}}^t) - \Psi(I^t, \mathbf{s}^t) \quad (3.18)$$

In addition, [Bordes et al., 2009] propose to exploit the sparsity of the data in the update step. The equation (3.16) can be expressed as

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t \mathbf{w}_t - \beta_t \mathbf{h}_t, \text{ where} \quad (3.19)$$

$$\alpha = \frac{1}{t_0 + t}, \quad \beta = \frac{\lambda^{-1}}{t_0 + t} \quad (3.20)$$

$$\mathbf{h}_t = \Psi(I^t, \hat{\mathbf{s}}^t) - \Psi(I^t, \mathbf{s}^t). \quad (3.21)$$

Note that if  $\mathbf{h}_t$  is sparse then subtracting  $\beta_t \mathbf{h}_t$  involves only the nonzero coefficients of  $\mathbf{h}_t$ , but subtracting  $\alpha_t \mathbf{w}_t$  involves all coefficients of  $\mathbf{w}_t$ . In turn, it is beneficial to reformulate the equation (3.19) as

$$\mathbf{w}_{t+1} = (1 - \alpha_t) \mathbf{w}_t - \beta_t \mathbf{h}_t. \quad (3.22)$$

By using this trick, the complexity  $\mathcal{O}(d)$  corresponding to the naive implementation of the update rule (3.16) reduces to the complexity  $\mathcal{O}(d_{\text{non-zero}})$  corresponding to the reformulated rule (3.22), where  $d$  is the dimension of the parameter vector and  $d_{\text{non-zero}}$  is the number of the non-zero elements in  $\mathbf{h}_t$ .

A big advantage of the SGD algorithm is its simplicity. Disadvantage is that the SGD algorithm does not provide any certificate of optimality and thus theoretically grounded stopping condition is not available.



## 4. Experiments

In this chapter we present comprehensive experimental evaluation of the proposed landmark detector using challenging data. This chapter is organized as follows:

In Section 4.1 we describe the “Labeled Faces in the Wild” (LFW) database which were used in the experiments.

In Section 4.2 we describe our evaluation procedure along with the performance statistics used to measure the accuracy of the detectors.

In Section 4.3 we describe three competing methods against which we compare our landmark detector. In particular, we compare against independently trained SVM detector (Section 4.3.1) and two public domain implementations of the facial landmark detectors which are based on the Active Appearance Models (Section 4.3.2) and the Deformable Part Models (Section 4.3.3). We would like to point out that especially the last mentioned landmark detector, which we will refer to as the Oxford detector due to its origin, is a strong competitor that has been used in numerous successful face recognition projects [Everingham et al., 2006], [Everingham et al., 2009], [Sivic et al., 2009].

In Section 4.4 we present comparison of the BMRM and the SGD solvers which were used for solving the SO-SVM learning problem.

In section 4.5 we present summary results of the experimental evaluation of the proposed detector and its comparison against the competing methods. We also provide basic timing statistics of the proposed detector.

Besides the parameters learned by the SO-SVMs, the proposed landmark detector is specified by several design options. In particular, the right combination of the feature descriptor for the local appearance model, the sizes of the components and the parametrization of the deformation cost have to be selected carefully. In order to select the best configuration we performed extensive experimental evaluation which is described in Appendix A. The experiments presented in this chapter use only the best found configuration specified in Appendix A.3.

### 4.1. Database: Labeled Faces in the Wild

We use the Labeled Faces in the Wild (LFW) database [Huang et al., 2007] for evaluation as well as for training our detector. The LFW database contains 13,233 images each of them  $250 \times 250$  pixels in size. The LFW database was augmented by manually annotating positions of 4 landmarks: centers of the left and the right eye, the mouth and the nose<sup>1</sup>. The LFW database contains a great ethnicity variance and the images have challenging background clutter.

Before using in experiments we preprocessed the LFW database as follows. First, we run the face detector on all images in the database. Second, we filtered out the images where i) the face detector missed the face and/or ii) the annotation is incomplete (e.g. in side faces only one eye is visible). Third, we determine the search spaces for individual components. We tune the parameters (size of the base window, margin of the base window and sizes of the components) in order to guarantee that 95% of images fit to the normalized image frame. The images that do not pass this step are discarded from evaluation and training. The preprocessing reduced the number of faces to 11,929.

---

<sup>1</sup>The annotation was provided by courtesy of Eyedea Recognition s.r.o. (<http://www.eyedea.cz/>)

## 4. Experiments

Some examples from the LFW face database are depicted in Figure 4.1. Note the challenging background of images. As you can see, some people also wear glasses or have beard. The LFW database consists mostly of relatively good quality images of famous people.



**Figure 4.1.** Examples from the LFW face database. As you can see, this database contains great ethnicity variance. Also it contains faces wearing glasses/sunglasses or which have beards.

### 4.2. Evaluation procedure

Our evaluation procedure involves three stages:

- i) training stage (estimation of the vector  $w$  from examples).
- ii) validation stage (selection of the optimal regularization parameter  $\lambda$ ).
- iii) testing stage (evaluation of the detector on hold out examples).

Each stage requires statistically independent set of examples. For this reason, we split the LFW database randomly into *training*, *validation* and *testing* sets. Table 4.1 describes the partitioning. The evaluation procedure itself is outlined in Algorithm 1.

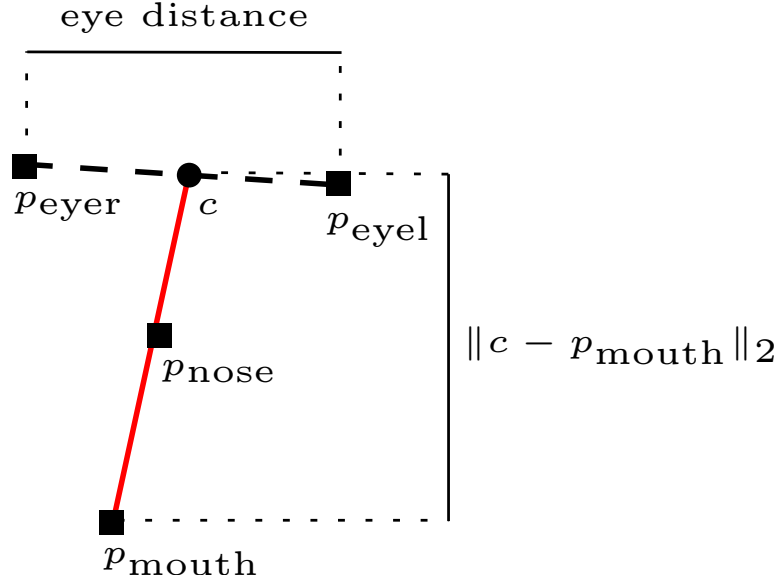
Originally, we used the same loss function for training and testing which is defined by equation (3.11). This loss function is given as an average deviation between the annotated and the estimated landmark positions. Later, we came up with a better loss which normalizes the deviations relatively to the length of the line connecting the center of eyes with mouth (see Figure 4.2). The normalized loss function accounts for a relatively large variance in the size of face boxes estimated by our face detector. As a result, we use a slightly different loss for training (3.11) and testing (4.4). We are aware that ideally we should have used the same normalized loss function also in the training stage. We did not do this due to the time reasons (evaluation of all experiments would take at least a month using our computer cluster). On the other hand, we do not expect large improvement in the accuracy if the normalized loss was used in the training stage.

Data set	Training	Validation	Testing
Percentage	60%	20%	20%
# of examples	7,157	2,386	2,386

**Table 4.1.** Partitioning of the LFW database into training, validation and testing set.

### 4.3. Competing methods

In this section we describe the three detectors used for comparison with the proposed detector.



**Figure 4.2.** Error normalization scheme. All measured deviations are normalized to the length of the line connecting the center of eyes with the mouth. This normalization is needed to make the comparison invariant to the changing scale of the detected faces (the face detector is not perfect).

#### 4.3.1. Independently trained binary SVMs detector

This detector is formed by binary (i.e. standard two-class) SVM classifiers trained independently for each facial landmark. For training we use the SVM solver implemented in LIBO-CAS [Franc and Sonnenburg, 2010]. For each facial landmark we create a different training set containing examples of the positive class and negative class. The positive class contains sub-images cropped around the ground truth position of the respective component. The negative class contains sub-images of the same size as the component which are cropped outside the ground truth region. In concrete, the negative sub-images satisfy the following condition

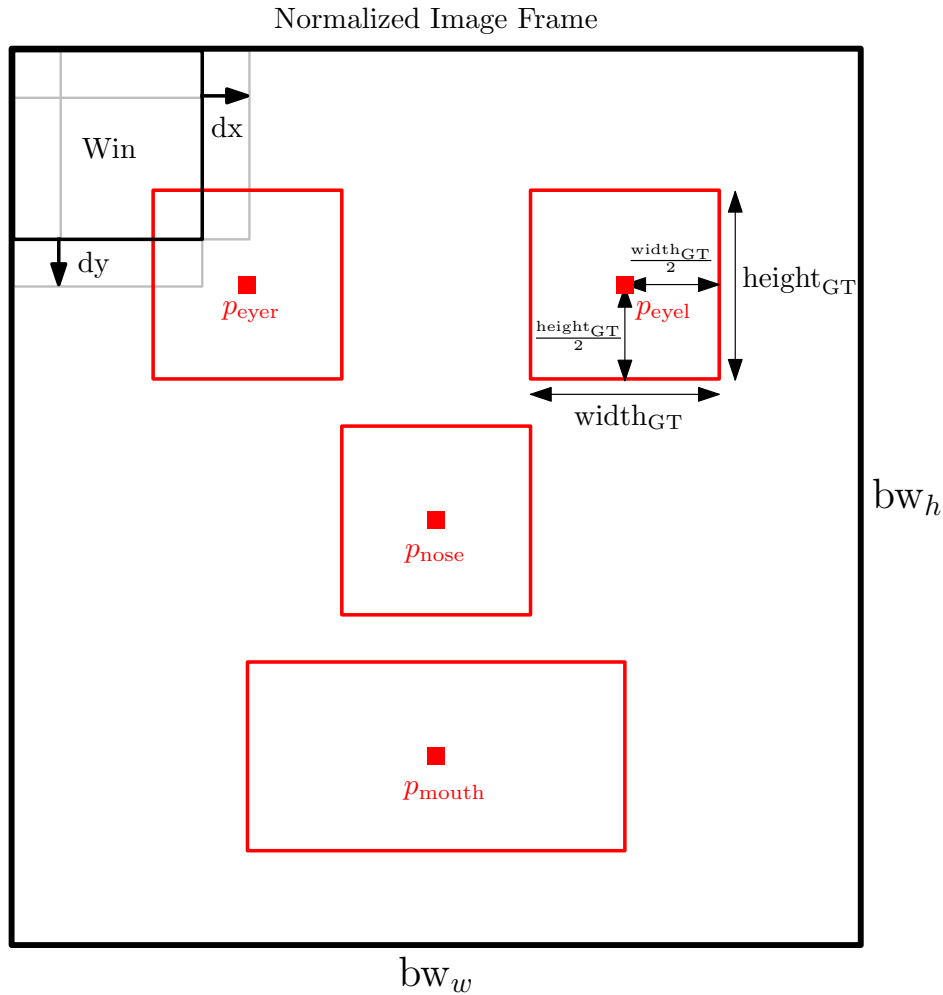
$$|P_-^x - P_{GT}^x| > \frac{1}{2} \text{width}_{GT} \quad (4.10)$$

$$|P_-^y - P_{GT}^y| > \frac{1}{2} \text{height}_{GT} \quad (4.11)$$

where  $P_-^x$  is the  $x$ -coordinate of the negative component and  $P_{GT}^x$  is the  $x$ -coordinate of the ground truth component.  $\text{width}_{GT}$  and  $\text{height}_{GT}$  denote the width and the height of the component. Figure 4.3 illustrates the scheme of the acquisition of the positive and negative examples for training the binary SVMs.

Having the binary SVM classifiers trained for all components, the landmark position is estimated by selecting the place with the maximal response of the classifier score function. The responses are evaluated in search regions defined for each component differently. The size of the search region is exactly the same as we use in the proposed structured SVM detector. We use this baseline detector mainly to show that learning of the deformation cost from data improves the accuracy. Note, that the binary SVM detector is a simple instance of the DPM where the deformation cost  $g_i(\mathbf{s}_0, \mathbf{s}_i)$  is zero for all positions inside the search region and it is  $-\infty$  outside the region.

## 4. Experiments



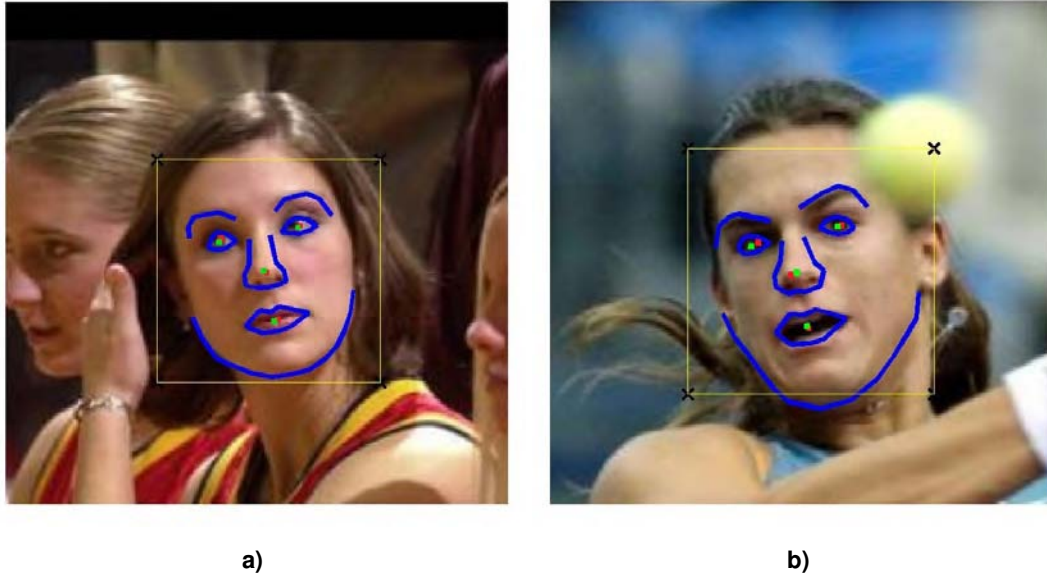
**Figure 4.3.** The acquisition of the positive and negative examples for the binary SVM training. The red rectangles are the positive examples for this image and the annotation. The Black rectangle labeled “Win” is the sliding window, which generates negative examples— window is shifted by  $dx$  or  $dy$  in the  $x$  or  $y$ -axis and cropped window is used as a negative example as long as the conditions (4.10) and (4.11) are fulfilled.

### 4.3.2. AAM

We use a slightly modified version of the publicly available implementation<sup>2</sup> of the AAM [Kroon, 2010]. As the initial guess of the face position required by the AAM, we use the center of the bounding box obtained from our face detector. The AAM estimates a dense set of feature points which are distributed around important face contours like the contour of mouth, eyes, nose and chin. The AAM requires a different training database which contains high resolution images along with annotation of all contour points. The used database is described in the next section.

To compare the AAM based detector with our detector, we have to transform the output of the AAM, i.e. points on contours around important face parts, to the landmark positions returned by our detector. To this end, we use centroids of the contours as the estimate of the corresponding landmark position. Figure 4.4 shows examples of the output of the AAM and the extracted landmark positions.

<sup>2</sup>Can be downloaded from <http://www.mathworks.com/matlabcentral/fileexchange/26706-active-shape-model-asm-and-active-appearance-model-aam>.



**Figure 4.4.** Example of detection made by the AAM detector on the LFW database. The green points are the ground truth positions of facial landmarks. The red points are estimated landmarks from the AAM contours.

### IIM Face database

For training the AAM model we use a publicly available IIM Face database described in [Nordstrøm et al., 2004]. The IIM database consists of 240 annotated images (6 images per person). Each image is  $640 \times 480$  pixel in size and comes with 58 manually annotated points which are distributed along the main face contours. The main disadvantage of this database is a lack of ethnicity variance.

In Figure 4.5 you can see some annotated examples from the IIM face database. Note that creation of the training examples for the AAM puts much higher demands on the annotator, because he/she has to click a large number of points (in our case 58) uniformly distributed on the respective contours. In contrast, our classifier requires only a few well defined points (in particular, 4 points corresponding to center of eyes, nose and mouth). Table 4.2 shows comparison of total number of annotated points for both face databases. It is seen that the labor required to create both databases is similar though the total number of images in the IIM database is much smaller.

Face database	# of annotated points
IIM	13920
LFW	28628

**Table 4.2.** Overall number of annotated points in the LFW and IIM face database.

### 4.3.3. Oxford detector

The last competing detector is the DPM based Oxford detector<sup>3</sup> [Everingham et al., 2008]. This detector was trained on a collection of consumer images which, however, are not available.

<sup>3</sup>Can be downloaded from <http://www.robots.ox.ac.uk/~vgg/research/nface/index.html>.



#### 4. *Experiments*

This detector returns corners of both eyes (2 landmarks for each eye), corners of mouth (2 landmarks) and 3 landmarks on the nose.

To compare the Oxford detector with our detector, we have to transform these landmarks to the landmarks returned by our detector. Similarly as in the AAM detector, we use the centroids of each logical group of landmarks. Figure 4.6 shows output of this detector together with the transformed landmarks.

**Algorithm 1** Evaluation procedure

- 1: **for** each  $\lambda \in \Lambda$  **do**
- 2: Find the parameter vector  $\mathbf{w}(\lambda)$  by solving (3.12) on the TRN set.
- 3: Compute the validation risk on  $p$  examples from the VAL set.

$$R_{\text{VAL}}(\mathbf{w}(\lambda)) = \frac{1}{p} \sum_{i=1}^p L(\mathbf{s}^i, \hat{\mathbf{s}}^i) \quad \text{where } \hat{\mathbf{s}}^i = \arg \max_{\mathbf{s} \in \mathcal{S}} \langle \mathbf{w}(\lambda), \Psi(I^i, \mathbf{s}) \rangle \quad (4.1)$$

- 4: **end for**
- 5: Find the optimal regularization constant

$$\lambda^* = \arg \min_{\lambda \in \Lambda} R_{\text{VAL}}(\mathbf{w}(\lambda)) \quad (4.2)$$

- 6: Compute the test risk on  $q$  examples on the TST set

$$R_{\text{TST}} = \frac{1}{q} \sum_{i=1}^q L'(\mathbf{s}^i, \hat{\mathbf{s}}^i) \quad \text{where } \hat{\mathbf{s}}^i = \arg \max_{\mathbf{s} \in \mathcal{S}} \langle \mathbf{w}(\lambda^*), \Psi(I^i, \mathbf{s}) \rangle \quad (4.3)$$

where the test loss is give by

$$L'(\mathbf{s}^i, \hat{\mathbf{s}}^i) = \kappa^i \frac{1}{M} \sum_{j=0}^{M-1} \|\mathbf{s}_j - \mathbf{s}_j^*\|, \quad (4.4)$$

$$\kappa^i = \frac{1}{\|\mathbf{c}^i - \mathbf{s}_{\text{mouth}}^i\|_2}, \quad (4.5)$$

$$\mathbf{c}^i = \frac{\mathbf{s}_{\text{eyel}}^i + \mathbf{s}_{\text{eyer}}^i}{2} \quad (4.6)$$

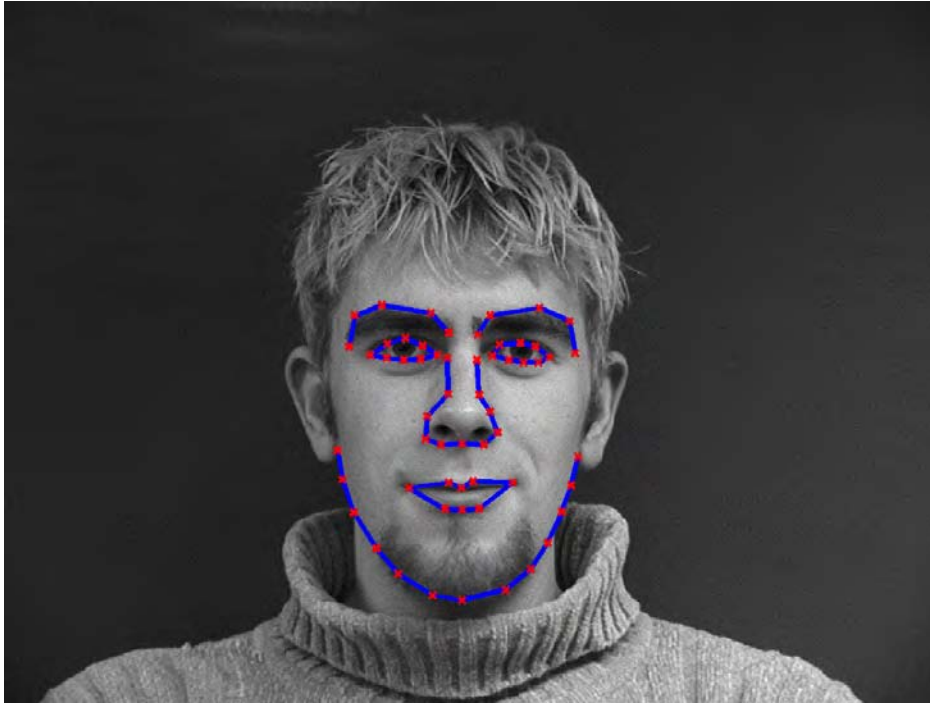
Evaluation of further test statistics:

$$R_{\text{TST}}^j = \frac{1}{q} \sum_{i=1}^q \kappa^i \|\mathbf{s}_j^i - \hat{\mathbf{s}}_j^i\|, \quad j = 0, \dots, M \quad (4.7)$$

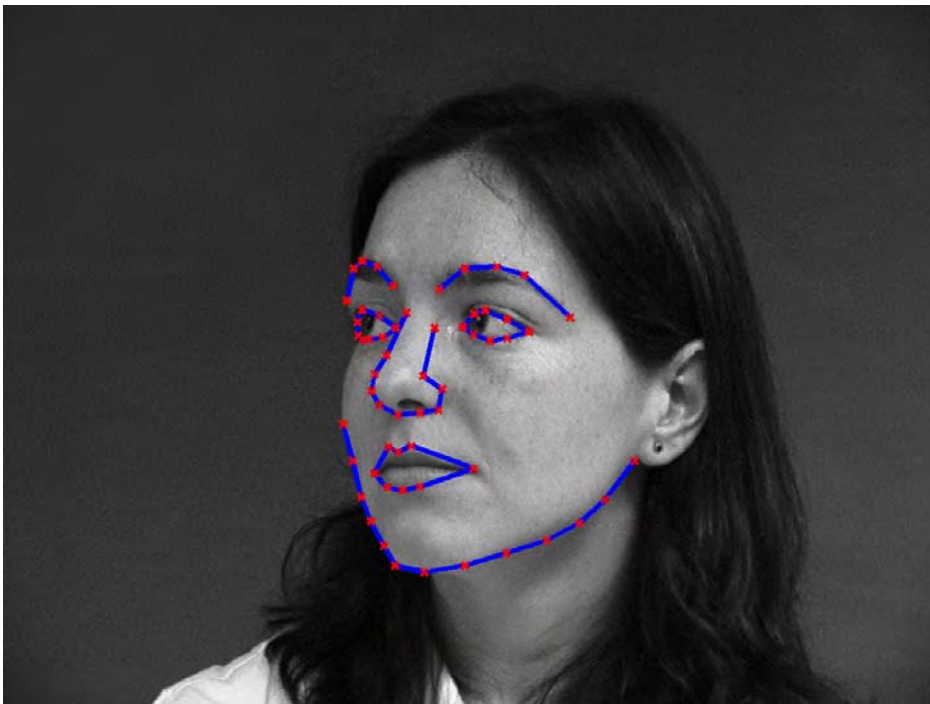
$$R_{\text{TST}}^{\max} = \frac{1}{q} \sum_{i=1}^q \max_{j=0, \dots, 3} \kappa^i \|\mathbf{s}_j^i - \hat{\mathbf{s}}_j^i\| \quad (4.8)$$

$$R_{\text{TST}}^{j_{\max}} = \max_{i=1, \dots, q} \kappa^i \|\mathbf{s}_j^i - \hat{\mathbf{s}}_j^i\| \quad (4.9)$$

#### 4. Experiments

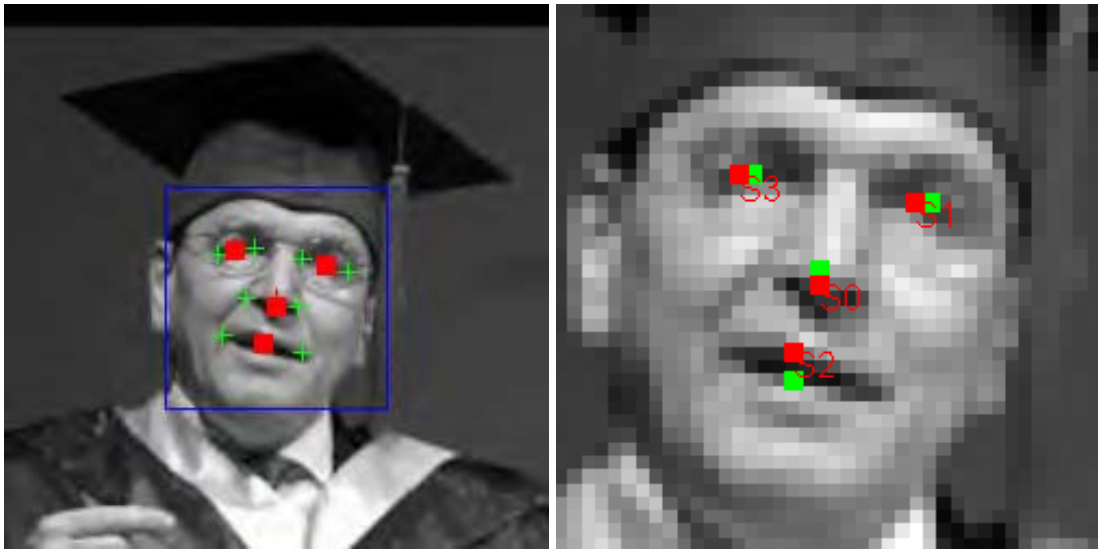


a) Frontal view



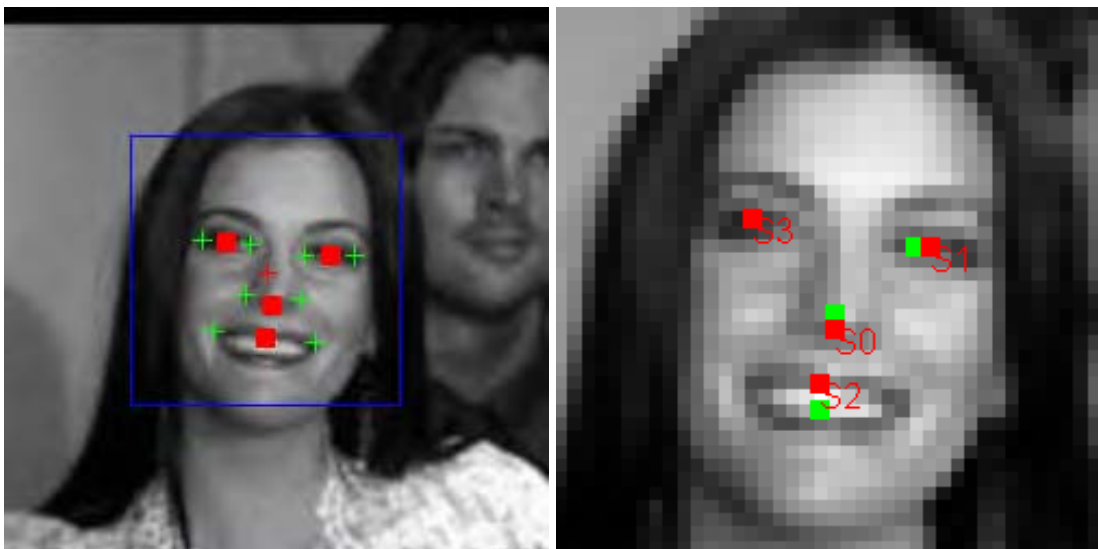
b) Side view

**Figure 4.5.** Some annotated examples of the IIM face database. The red crosses are the landmarks from the annotation. The blue polylines depicts the main contours which are used to estimate landmarks that are consistent with the LFW database annotation (eyes, nose and mouth).



**a)** Original image from the LFW database. The blue rectangle is the bounding box provided by the face detector, the red cross is its center. The green crosses are the detected landmarks. The red squares are transformed landmarks for comparison with our detector.

**b)** Normalized image frame. The red points are estimated positions of facial landmarks. The green points are the ground truth positions from the image annotation.



**c)** Original image from the LFW database. The blue rectangle is the bounding box provided by the face detector, the red cross is its center. The green crosses are the detected landmarks. The red squares are transformed landmarks for comparison with our detector.

**d)** Normalized image frame. The red points are estimated positions of facial landmarks. The green points are the ground truth positions from the image annotation.

**Figure 4.6.** Output of the Oxford's detector together with the transformed landmarks for the comparison with our detector.

#### 4.4. Comparison of BMRM and SGD

This section describes the experiment which compares two solvers for the SO-SVM problem. Namely, the Bundle Method for Regularized risk Minimization (BMRM) (Section 3.2.1) and the Stochastic Gradient Descent (SGD) (Section 3.2.2) are compared on the problem of learning the landmark detector. Parameters for this experiment are set equally to those in the experiments described in Appendix A.2 (see Table A.4).

The task of the solvers is to minimize the following convex objective function

$$F(\mathbf{w}) = \frac{\lambda}{2} \|\mathbf{w}\|^2 + R(\mathbf{w}). \quad (4.12)$$

Besides the objective value  $F(\mathbf{w})$ , we are also interested in the value of the validation risk  $R_{\text{VAL}}(\mathbf{w})$  (defined by equation (4.1)) which is another important criterion characterizing the trained classifier. To make the iterations of the BMRM and SGD comparable we define one iteration of the SGD as a sequence of  $m$  single update steps where  $m$  is the number of training examples. The best SGD parameter  $t_0$  was selected from a set  $\{1, 10, \dots, 10^6\}$  according to the minimum of the objective function  $F(\mathbf{w})$  computed only on 10% training examples after one pass of the SGD algorithm thorough the data. Note that for each value of  $\lambda$  we have to tune the parameter  $t_0$  again. We fixed the total number of iterations of the SGD algorithm to 50.

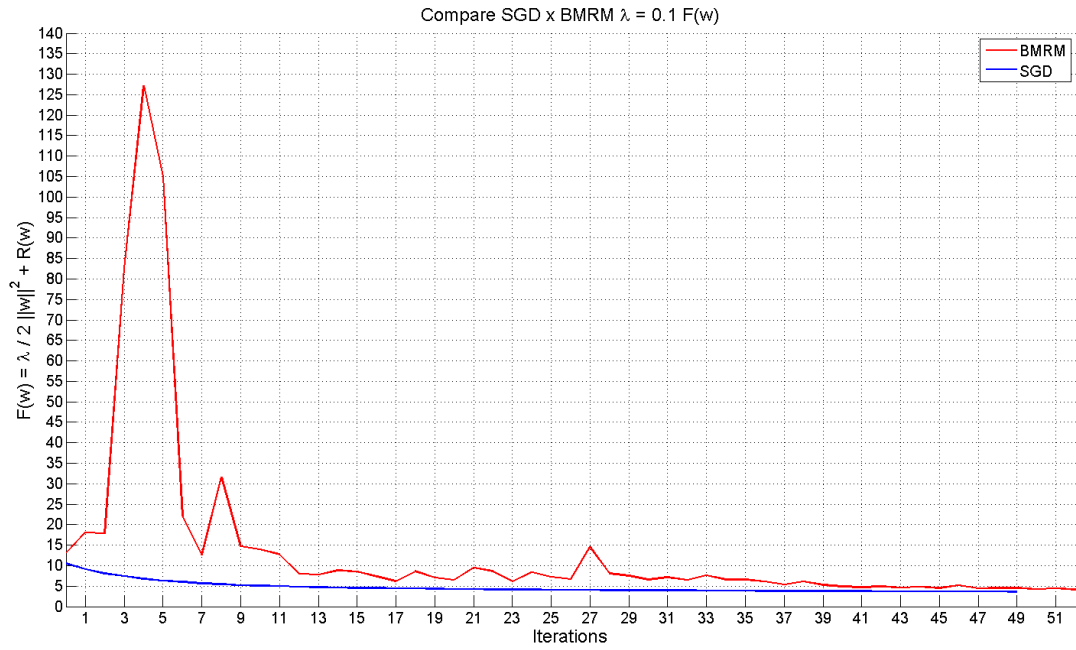
We run both solver on the problem (4.12) with the parameters  $\lambda \in \{10^{-2}, 10^{-1}, \dots, 1\}$  and we recorded both  $F(\mathbf{w})$  and  $R_{\text{VAL}}(\mathbf{w})$ . Results of the experiment are summarized in Table 4.3. Figure 4.7 and 4.8 show convergence curves for  $\lambda = 0.1$  and  $\lambda = 10$ .

It is seen that the SGD converges quickly at the beginning and it stalls as it approaches the minimum of the objective  $F$ . The validation risk achieved by the SGD after 50 iterations is in many cases comparable to the validation risk obtained by the BMRM after much more iterations which are required to achieve solution with guaranteed high precision. The problem is that solution obtained after 50 iterations is in some cases much worse than the precise solution. For example, 50 iterations is enough for  $\lambda = 0.1$  (see Figure 4.7), however it is insufficient for  $\lambda = 10$  (see Figure 4.8). Unfortunately, there is no versatile method to set the correct number of iterations for the SGD algorithm (unless one knows the optimal solution). On the other hand, the BMRM algorithm has a reasonable stopping condition specified by the maximal deviation from the optimal value of the optimized objective function. Hence, we conclude that SGD is useful in cases when using the precise but slower BMRM algorithm is not feasible. In opposite cases the BMRM algorithm returning the solution with guaranteed optimality certificate is preferable. In the remaining experiments we use a parallelized variant of the BMRM algorithm.

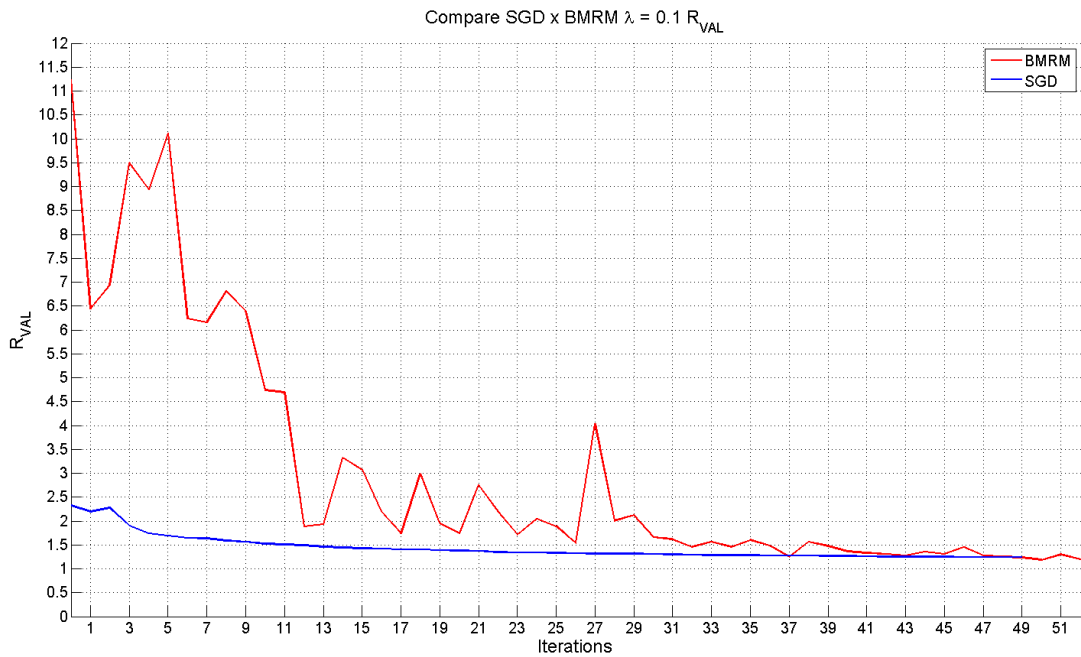
		# of iterations							
		50			29	434	128	52	50
		$10^{-2}$	$10^{-1}$	1	10	$10^{-2}$	$10^{-1}$	1	10
BMRM	$F(\mathbf{w})$	5.201	4.553	7.045	10.920	1.985	3.719	6.923	
	$R_{\text{VAL}}$	1.680	1.240	1.478	3.182	1.078	1.126	1.473	
SGD	$F(\mathbf{w})$	3.316	3.632	7.666	12.160				12.130
	$R_{\text{VAL}}$	1.210	1.243	1.935	5.173				4.850

**Table 4.3.** Comparison of the BMRM and SGD.

#### 4.4. Comparison of BMRM and SGD



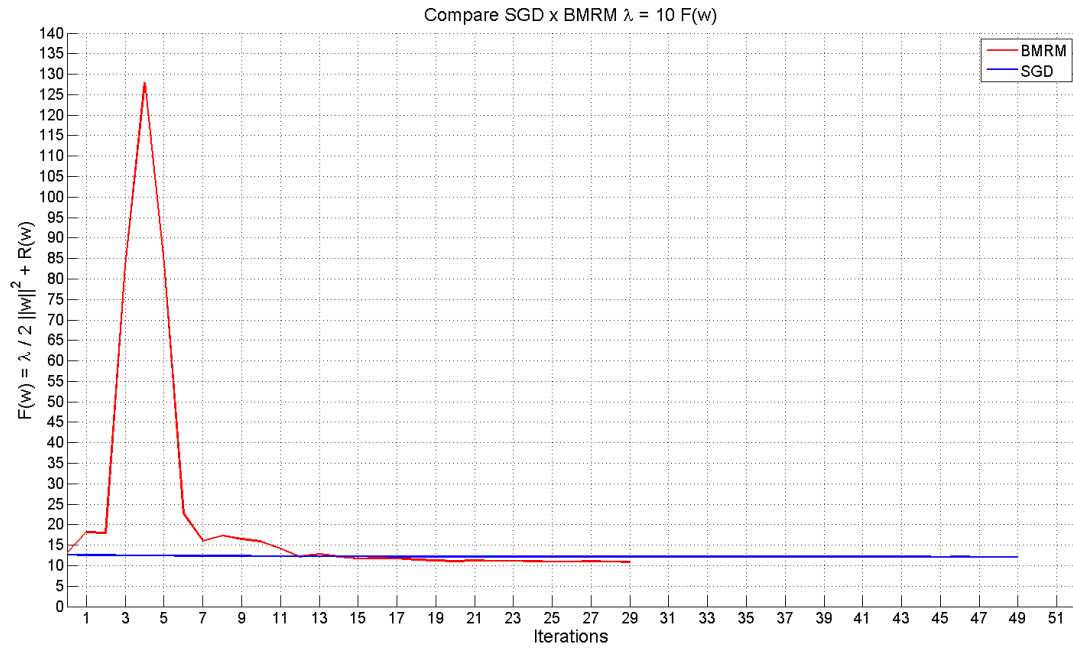
a) Objective function  $F(w)$



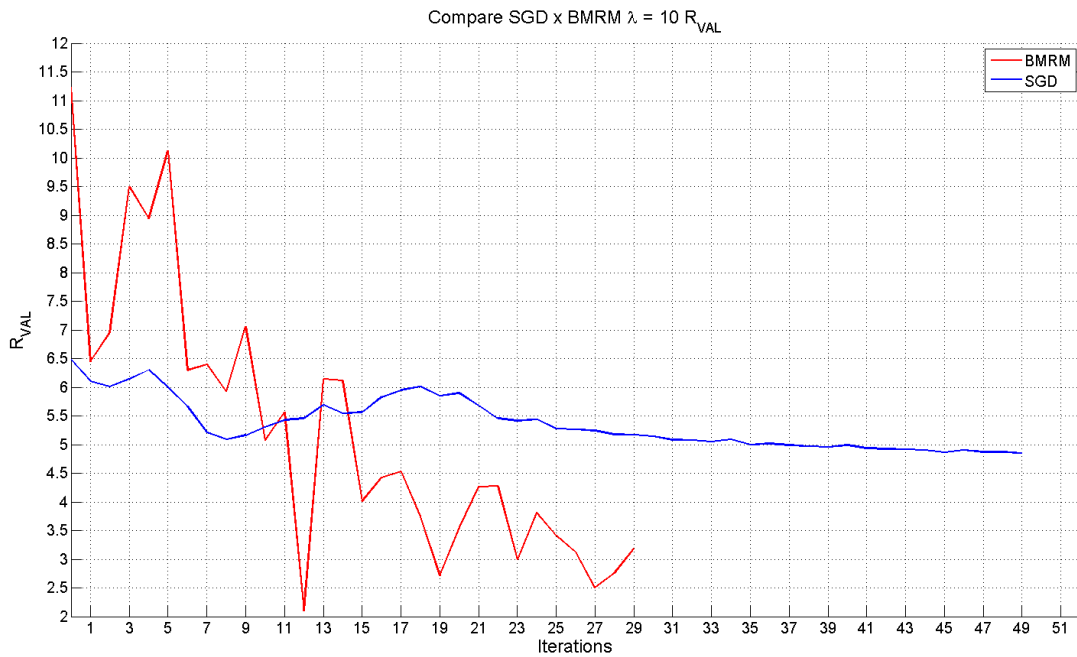
b) Validation risk  $R_{VAL}(w)$

**Figure 4.7.** The comparison of the BMRM and SGD. Cutout of the first 50 iterations from the graphs of (a) the objective function  $F(w)$  for  $\lambda = 0.1$  (b) the validation risk  $R_{VAL}(w)$  for  $\lambda = 0.1$ .

## 4. Experiments



**a)** Objective function  $F(w)$



**b)** Validation risk  $R_{VAL}(w)$

**Figure 4.8.** The comparison of the BMRM and SGD. (a) the graph of the objective function  $F(w)$  for  $\lambda = 0.1$  (b) the graph of the validation risk  $R_{VAL}(w)$  for  $\lambda = 10$ .

## 4.5. Summary results

In this section we compare the proposed landmark detector with the three competing detectors described in Section 4.3 in terms of the accuracy of estimating the landmark positions. To measure the accuracy, we follow the evaluation protocol described in Algorithm 1. In the case of the competing detectors whose models are trained differently (see Section 4.3 for more details), we execute only the last step of the algorithm which evaluates the test statistics.

We measure several accuracy statistics which are defined in the last step of Algorithm 1. The notion of relative error is equivalent to the error normalization relatively to the size of the face as described in Section 4.2. Recall, that the face size is defined as the length of the line connecting the mouth and the point between eyes.

The overall results are summarized in Table 4.5 (mean deviations per component) and Table 4.6 (maximal deviations per component). Figure 4.9 shows the cumulative histograms of the count of occurrences of the relative errors. That is, the cumulative histogram shows the number of test examples which have the relative error less or equal certain value. Table 4.4 shows the detail around 10% of the relative error taken from Figure 4.9. That is, this table shows the percentage of the test examples which have the relative error less or equal to 10% of the face size. As can be seen, the proposed detector clearly outperforms all its competitors in all measured statistics.

We also measured average of the detection time which was below 100 milliseconds per image on a notebook with Intel Core 2 Duo T9300 2.50 GHz. Around 75% of the detection time takes computation of the LBP features. The rest is consumed by solving the max-sum problem (3.5). Note, however, that this time is measured in the MATLAB implementation. Moreover, the code can be further optimized, e.g. by computing the LBP features in parallel. Hence, the 100ms is a conservative estimate of the detection time.

**Detail around 10% taken from Figure 4.9**

	Average mean deviation	Average maximal deviation
AAM	18.57 %	2.831 %
Binary SVMs	91.91 %	62.53 %
Oxford's detector	71.63 %	16.20 %
<b>proposed detector</b>	<b>97.15 %</b>	<b>77.25 %</b>

**Table 4.4.** The values are percentages of test examples with error less or equal to 10 % of the face size.

**Mean deviations per component**

	AAM	Binary SVMs	Oxford	<b>proposed detector</b>
$R_{TST}^{\text{left eye}}$	17.1167	5.3333	6.5028	<b>4.0931</b>
$R_{TST}^{\text{right eye}}$	16.4095	5.2212	5.8537	<b>3.9484</b>
$R_{TST}^{\text{mouth}}$	16.9982	5.9941	12.5138	<b>5.2365</b>
$R_{TST}^{\text{nose}}$	17.1284	7.0347	12.2694	<b>5.7556</b>
$R_{TST}$	16.9132	5.8958	9.2849	<b>4.7584</b>

**Table 4.5.** Summary of mean errors. Average mean deviation for each landmark  $R_{TST}^j$  is computed according to (4.7).  $R_{TST}$  is defined by (4.3). We call the  $s_0$  nose, but in the proposed detector is this component rather the center of the face.

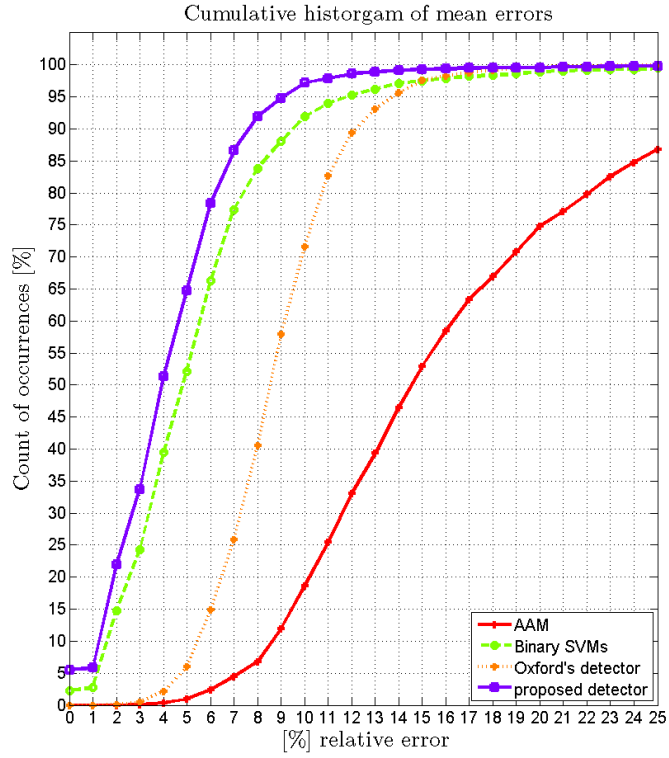


#### 4. Experiments

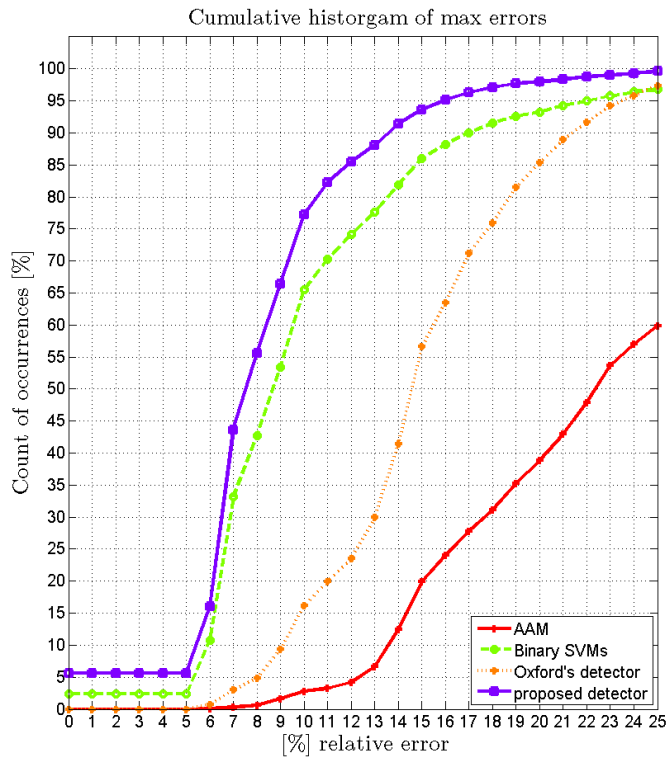
**Maximal deviations per component**

	AAM	Binary SVMs	Oxford	<b>proposed detector</b>
$R_{TST}^{\max, \text{left eye}}$	100.3249	66.6667	44.7214	<b>41.0651</b>
$R_{TST}^{\max, \text{right eye}}$	89.0327	96.5146	52.1536	<b>74.9429</b>
$R_{TST}^{\max, \text{mouth}}$	70.6225	64.4465	37.9987	<b>80.5220</b>
$R_{TST}^{\max, \text{nose}}$	65.4023	77.2270	77.2496	<b>34.4904</b>
$R_{TST}^{\max}$	25.7790	11.6788	15.9857	<b>9.8533</b>

**Table 4.6.** Summary of maximal deviations. Average maximal deviation for each landmark  $R_{TST}^{j, \max}$  is computed according to (4.9).  $R_{TST}^{\max}$  is defined by (4.8). We call the  $s_0$  nose, but in the proposed detector is this component rather the center of the face.



a) Average mean deviation



b) Average maximal deviation

**Figure 4.9.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for all experiments.



## 5. Implementation

We implemented an open source library which contains the proposed landmark detector as well as the SO-SVM algorithm for learning its parameters from annotated images. The homepage of the library **flandmark** is at <http://cmp.felk.cvut.cz/~uricamic/flandmark/>. The library is a collection of MATLAB and C codes. The learning scripts are implemented in Matlab. The time demanding procedures of the learning algorithm like the QP solver or the evaluation of the cost  $q_i(I, s_i)$  are implemented in C and interfaced to Matlab. The landmark detector itself is implemented both in Matlab (this implementation was used in experiments and can be useful for further prototyping) and in C with a simple API for integrating the detector to other applications. A MEX-interface to Matlab for the C implementation of the detector is also provided. The library implements only the best configuration of the landmark detector found in experiments.

The following MATLAB example creates the binary file describing the detector from the structure `model` obtained by the learning script and it calls the mex-function for the facial landmark detection on an image.

```
1 %% Creation of binary file holding the model structure
2 % load structure model
3 load('./data/exp03_detector_model_gdisp.mat'); % contains the ...
   structure "model"
4
5 % save the structure model to a binary file
6 flandmark_load_model(model, './data/model_changeS0.dat');
```

```
1 %% Detection
2 % get normalized frame from image
3 I = rgb2gray(imread('photo.jpg'));
4 bbox = dlmread('photo.dat'); % the detected face box is in ...
   file 'photo.dat'
5 [face_image bbox2] = getNormalizedFrame(I, bbox(1,:), ...
   model.data.options);
6
7 % call the detector
8 detection = flandmark_detector(face_image(:), ...
   './data/model_changeS0.dat');
```



## 6. Conclusions

In this thesis we have developed a detector of facial landmarks based on the Deformable Part Models. We have formulated the problem of landmark detection as an instance of the structured output classification which allows to specify requirements on the detector's accuracy via a user-defined loss function. We use the Structured Output Support Vector Machine algorithm for learning parameters of the detector from annotated images. In contrast to the previous works, we learn the parameters of the detector in one-stage process and the objective function of the learning algorithm is directly related to the performance of the resulting detector.

We have performed extensive experiments in order to find the best configuration of the landmark detector from a large number of design options.

We have evaluated performance of the proposed detector on a challenging database and compared its accuracy against two public domain landmark detectors based on the Active Appearance Models and finely tuned Deformation Part Models. Especially the latter landmark detector was a very strong competitor which had been previously used in many successful face recognition projects. The empirical results demonstrate that the proposed landmark detector clearly outperforms all its competitors in all measured statistics.

We have implemented an open source library **flandmark**

<http://cmp.felk.cvut.cz/~uricamic/flandmark/>

which contains the implemented detector as well as the algorithm for learning its parameters from annotated examples.



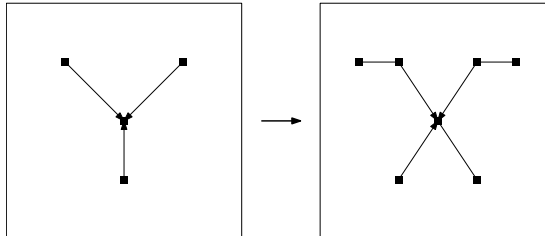
## 7. Further extensions

Although the proposed detector is fully functional there is still a large room for further improvements. We summarize the main ideas of these extensions:

- We used slightly different loss functions in training and testing stages. Of course, using the same loss function in both stages would be better. On the other hand, the difference in the loss functions is minor and we do not expect big boost in accuracy if the same loss is used.
- We used a single loss function which measures the average deviation of the estimated landmark positions. There are clearly other interesting options to try, for instance the maximal deviation over the components

$$L(\mathbf{s}, \mathbf{s}^*) = \max_i \|\mathbf{s}_i - \mathbf{s}_i^*\| \quad (7.1)$$

- We used a simple star-like structure to describe the landmark deformation cost. It would be interesting to experiment with more complex configurations like e.g. using complete graph instead of the star-like structure.
- We used the centers of important facial parts (eyes, nose, mouth) as the components. This option was predetermined by the available annotation of our database. It seems to be a better options to use the corners of the parts because of their more discriminative structure. Using the corners will require a different structure of the deformable part model as shown in Figure 7.1. However, our implementation can easily accommodate this modification.



**Figure 7.1.** Modification of the deformable part model. The current variant versus the new one uses the corners instead of the centers.

- The current code uses single core implementation of the landmark detector. However, both solving the max-sum problem (3.5) and manly computation of features of the local appearance model can be done in parallel. For example, computation of the LBP pyramid can be parallelized very efficiently.





## A. Experimental tuning of the detector configuration

In this appendix we describe the experiments that were made to find out the best configuration of the parameters of the detector which cannot be learned by the SO-SVM algorithm. First two experiments (see Sections A.1 and A.2) concern the deformation cost function  $g_i(s_0, s_i)$  (see Section 3.1.2). In the next experiment (see Section A.3) we focus on the components and we try to change the landmark which represents the nose for the landmark representing the center of the face. The rest of experiments deals with the appearance model  $g_i(I, s_i)$  and follows outline of Section 3.1.1 (see Sections A.4, A.5, A.6 and A.7). In the end of this appendix we provide a comparison of all experiments.

### A.1. Structured output SVM with table deformation cost

In the first experiment we build the proposed model with the deformation cost represented by a table (see Section 3.1.2 for details). As the model of appearance we use the LBP pyramid (see Section 3.1.1). Because of the nature of this deformation cost we can exploit the sparsity of the feature map  $\Psi_i^g(I, s_i)$  (which is made by the composition of LBP pyramid features with height of pyramid equal to 4) and the deformation cost map  $\Psi^g(s_0, s_i)$  (which in this case is the identity matrix).

Learning of the joint parameter vector  $w$  for the optimal  $\lambda = 0.1$  converged to precision  $\epsilon = 0.01$  in 48 iterations. Overall training time (i.e. learning of the joint parameter vector  $w$  for all  $\lambda \in \{10^{-3}, 10^{-2}, \dots, 10\}$ ) took less than a day (about 20 hours) computed parallel in 8 threads. One iteration took less than 4 minutes.

#### A.1.1. Parameters

Table A.1 shows the parameters settings for this experiment. Results of the validation depicts Table A.2. Optimal value of the regularization term  $\lambda$  is denoted in bold.

Structured output SVM with the table deformation cost									
Base window	$[40, 40]^T$ px								
Base window margin	$[20, 20]^T$ %								
Components	<table border="1"> <tr> <td>13</td> <td>13</td> <td>20</td> <td>13</td> </tr> <tr> <td>13</td> <td>13</td> <td>13</td> <td>13</td> </tr> </table> px	13	13	20	13	13	13	13	13
13	13	20	13						
13	13	13	13						

**Table A.1.** Parameters settings for the experiment: structured output SVM with the table deformation cost

#### A.1.2. Results

As we already mentioned the deformation cost represented by a table has many disadvantages. We should ideally provide training database that has each combination of  $(s_0, s_i)$  present at least once. Otherwise the corresponding weight in the joint parameter vector  $w$  is set to zero. Also this kind of the deformation cost allows strange configurations of the estimated landmark

A. Experimental tuning of the detector configuration

**Structured output SVM with the table deformation cost**

$\lambda$	$R_{\text{TRN}}$	$R_{\text{VAL}}$
$10^{-2}$	0.22937	0.77943
$10^{-1}$	<b>0.60101</b>	<b>0.77824</b>
1	0.97624	1.00844
10	1.28770	1.27809

**Table A.2.** The training risk and validation risk as a function of the regularization constant  $\lambda$  measured for the experiment structured output SVM with the table deformation cost. The optimal  $\lambda$  minimizing the validation risk is denoted in bold.

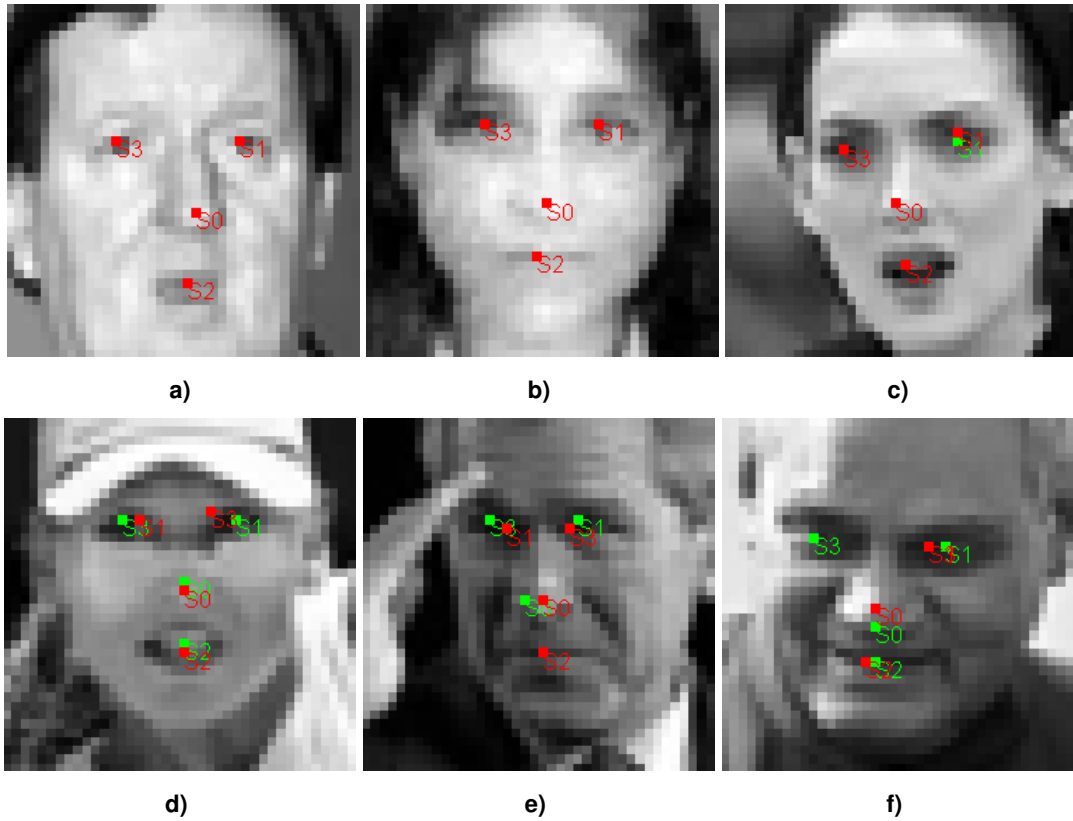
positions (see Figures A.1d, A.1e or A.1f). Figure A.1 depicts some randomly chosen images from the TST set.

Table A.3 shows the normalized errors for each landmark as well as the average mean  $R_{\text{TST}}$  and average maximal  $R_{\text{TST}}^{\max}$  deviation. Figure A.2 shows the cumulative histograms of the average mean and maximal deviation estimated on the test examples for the detector with parameters set as described in this experiment. Section 4.5 summarizes results of all experiments together.

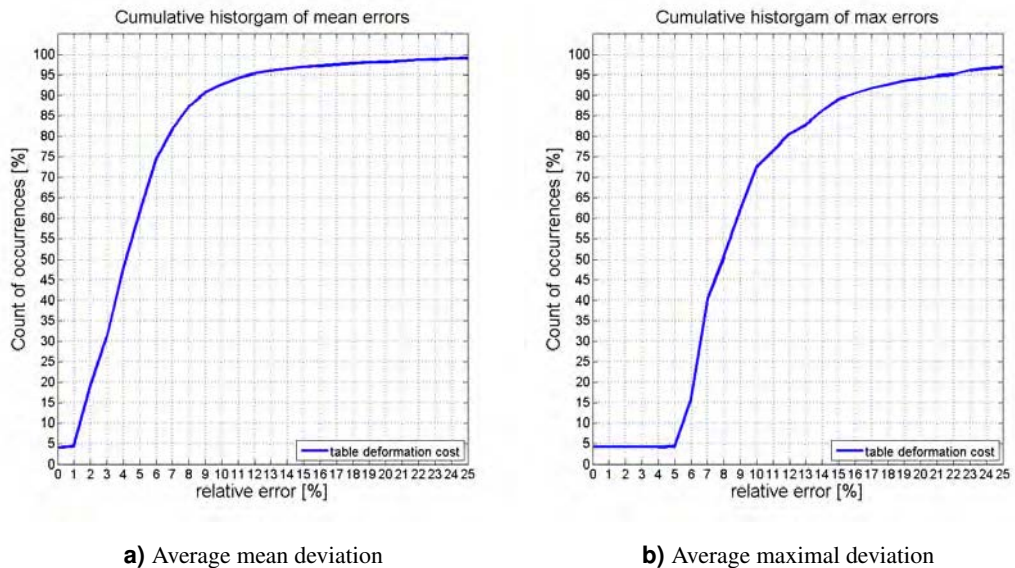
**Structured output SVM with the table deformation cost**

	Left eye $j = 1$	Right eye $j = 3$	Mouth $j = 2$	Nose $j = 0$
$R_{\text{TST}}^j$	4.8684	4.8974	5.3685	6.7003
$R_{\text{TST}}^{j\max}$	100.2596	96.5146	74.2781	115.7292
$R_{\text{TST}}^{\max}$	11.39167			
$R_{\text{TST}}$	5.45866			

**Table A.3.** Normalized errors of the experiment: structured output SVM with the table deformation cost.



**Figure A.1.** Image results for experiment: structured output SVM with the table deformation cost. The red squares are estimated landmarks. The green squares are the ground truth positions. The top row shows some good results of the landmark estimation, the bottom row shows the worst results. Note that in A.1d and A.1e the landmarks for eyes are swapped.



**Figure A.2.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for the experiment structured output SVM with the table deformation cost.

## A.2. Structured output SVM with displacement deformation cost

In the second experiment we build the proposed model with the deformation cost represented by a displacement (see section 3.1.2 for details). As the model of appearance we use the LBP pyramid (see Section 3.1.1). Because of the nature of this deformation cost we can exploit the sparsity of the feature map  $\Psi_i^q(I, s_i)$  (which is made by the composition of LBP pyramid features with height of pyramid equal to 4) and the deformation cost map  $\Psi^g(s_0, s_i)$  (which in this case are only four numbers for each  $(s_0, s_i)$  pair).

Learning of the joint parameter vector  $w$  for the optimal  $\lambda = 0.1$  converged to precision  $\epsilon = 0.01$  in 132 iterations. Overall training time (i.e. learning of the joint parameter vector  $w$  for all  $\lambda \in \{10^{-3}, 10^{-2}, \dots, 10\}$ ) took 6 days and 9 hours computed parallel in 8 threads. One iteration took less than 3 minutes.

### A.2.1. Parameters

Table A.4 shows the parameters settings for this experiment. Results of the validation depicts Table A.5. Optimal value of the regularization term  $\lambda$  is denoted in bold.

Base window	$[40, 40]^T$ px
Base window margin	$[20, 20]^T$ %
Components	$\begin{bmatrix} 13 & 13 & 20 & 13 \\ 13 & 13 & 13 & 13 \end{bmatrix}$ px

**Table A.4.** Parameters settings for the experiment: structured output SVM with the displacement deformation cost

$\lambda$	$R_{\text{TRN}}$	$R_{\text{VAL}}$
$10^{-2}$	0.17265	0.72376
$10^{-1}$	<b>0.52029</b>	<b>0.69892</b>
1	0.84171	0.87090
10	1.44613	1.46576

**Table A.5.** The training risk and validation risk as a function of the regularization constant  $\lambda$  measured for the experiment structured output SVM with the displacement deformation cost. The optimal  $\lambda$  minimizing the validation risk is denoted in bold.

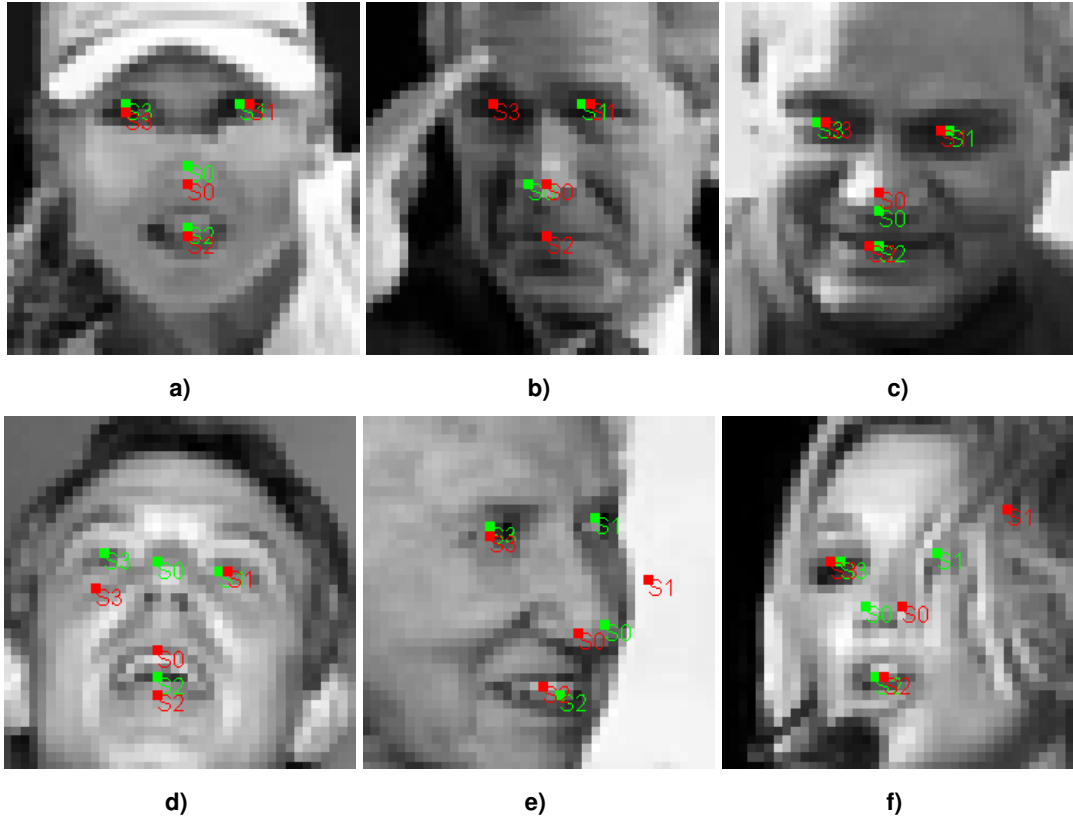
### A.2.2. Results

The deformation cost represented by the displacement instead of the table have quite a big impact on the detector performance. The displacement representation also reduces the dimensionality of the joint parameter vector  $w$ . Figure A.3 depicts some randomly chosen images from TST set with the detected landmarks. Note that Figures A.3a, A.3b and A.3c are the same images as in the previous experiment (see Figure A.1). Figures A.3d, A.3e and A.3f shows the worst results.

Table A.6 shows the normalized errors for each landmark as well as the average mean  $R_{\text{TST}}$  and average maximal  $R_{\text{TST}}^{\max}$  deviation. Figure A.4 shows the cumulative histograms of the

## A.2. Structured output SVM with displacement deformation cost

average mean and maximal deviation estimated on the test examples for the detector with parameters set as described in this experiment. Section 4.5 summarizes results of all experiments together.



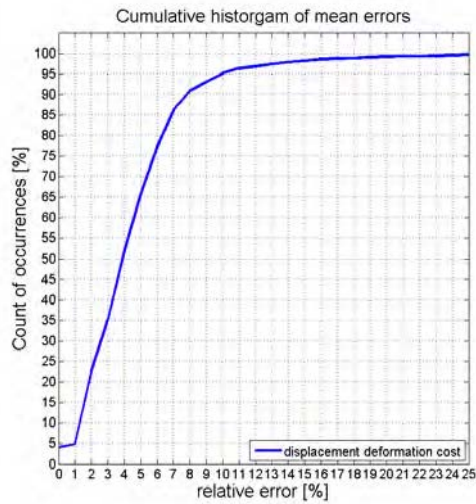
**Figure A.3.** Image results for experiment: structured output SVM with the displacement deformation cost. The red squares are estimated landmarks. The green squares are the ground truth positions. The top row shows the results to compare with the previous experiment (see Section A.1). Note that the displacement deformation cost gives much better results for these images. The bottom row shows the worst results.

**Structured output SVM with the displacement deformation cost**

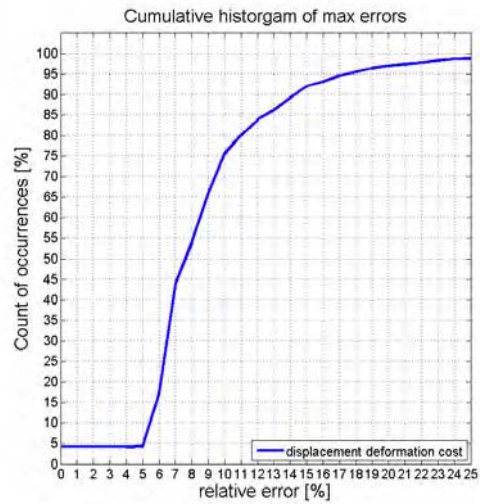
	Left eye $j = 1$	Right eye $j = 3$	Mouth $j = 2$	Nose $j = 0$
$R_{TST}^j$	4.2234	4.5212	4.9546	6.0083
$R_{TST}^{j\max}$	81.3456	80.0000	73.2177	76.8662
$R_{TST}^{\max}$	10.10671			
$R_{TST}$	4.92684			

**Table A.6.** Normalized errors of the experiment: structured output SVM with the displacement deformation cost.

A. Experimental tuning of the detector configuration



a) Average mean deviation



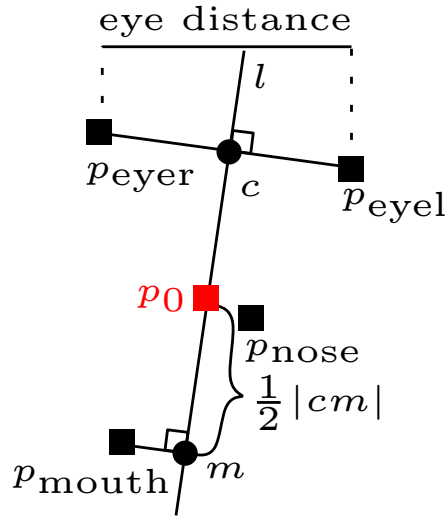
b) Average maximal deviation

**Figure A.4.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for the experiment structured output SVM with the displacement deformation cost.

### A.3. Modification of $s_0$

In this experiment we replace  $s_0$  (i.e. the nose component) with the center of the face and make this component larger. We do this because the proposed model is defined as a star-like structure with the center component. The nose is hard to be defined with only one point (different annotators mark the nose differently). The center of the face is on the other hand not so much dependent on the face rotation and can be computed exactly.

Because the LFW annotation does not have entry for the center of the face, we have to define it. We define the center of the face as the point derived from the annotation as follows: Let  $c$  be the center of both eyes (i.e.  $c = \frac{p_{\text{eyel}} + p_{\text{eyer}}}{2}$ ) and  $l$  be the normal to the line connecting both eyes. Then the center of the face is defined as the midpoint of the line segment  $l_2 = c - m$ , where  $m$  is the orthogonal projection of  $p_{\text{mouth}}$  on the line  $l$ . See Figure A.5 for clarification.



**Figure A.5.** The definition of the center of the face  $p_0$ .

#### A.3.1. Parameters

Table A.7 shows the parameters settings for this experiment. Results of the validation depicts Table A.8. Optimal value of the regularization term  $\lambda$  is denoted in bold.

Base window	$[40, 40]^T$ px
Base window margin	$[20, 20]^T$ %
Components	$\begin{bmatrix} 20 & 13 & 20 & 13 \\ 20 & 13 & 13 & 13 \end{bmatrix}$ px

**Table A.7.** Parameters settings for experiment: modification of  $s_0$ .

#### A.3.2. Results

Modification of the  $s_0$  component appears to be a good choice — it solves a lot of really bad detections from the previous experiments (see Figure A.7 for comparison). Figure A.6 shows some randomly chosen examples from the TST set. The  $s_0$  component can be ignored in image results of this experiments, for it does not correspond to the real facial landmark. It is defined



A. Experimental tuning of the detector configuration

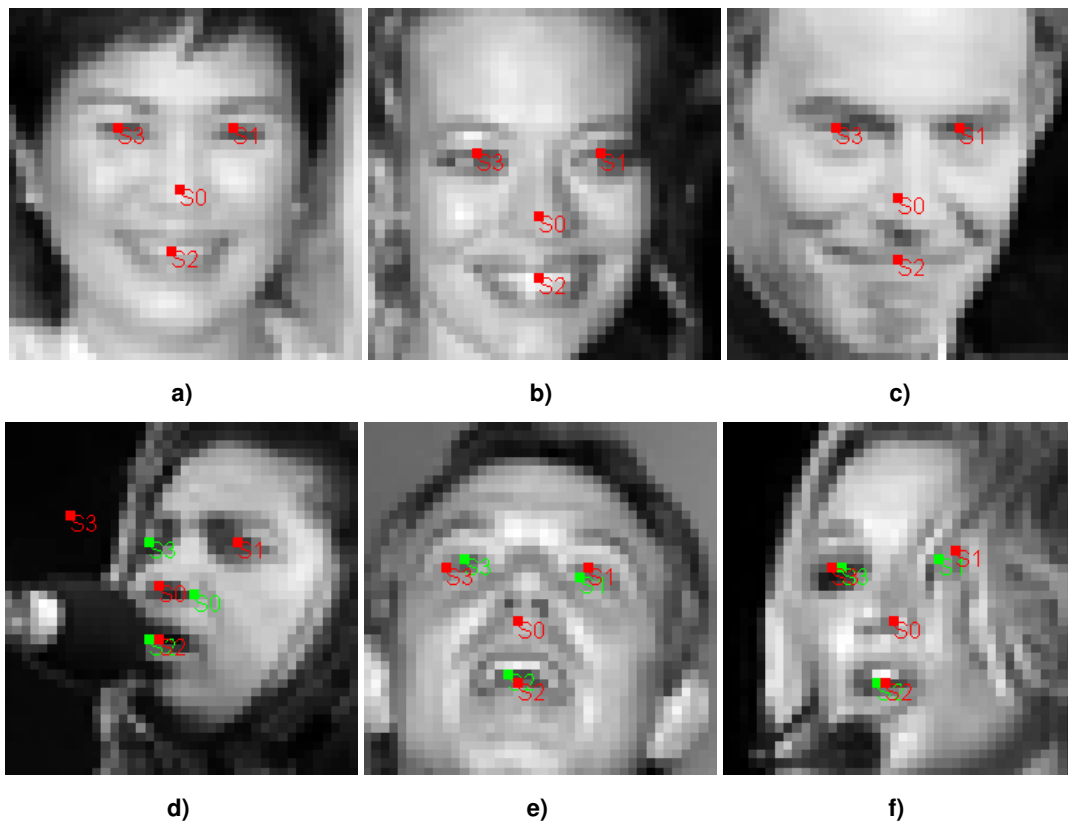
**Modification of  $s_0$**

$\lambda$	$R_{\text{TRN}}$	$R_{\text{VAL}}$
$10^{-2}$	0.06150	0.72250
<b><math>10^{-1}</math></b>	<b>0.36933</b>	<b>0.66465</b>
1	0.69369	0.75019

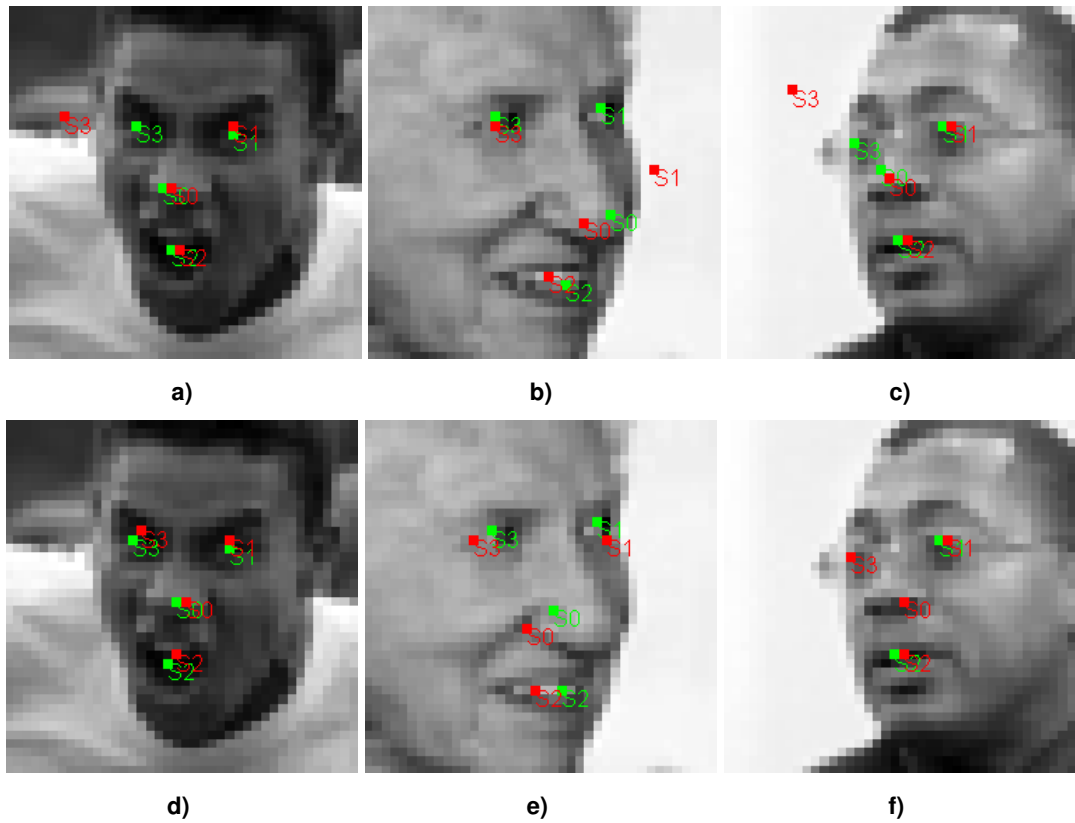
**Table A.8.** The training risk and validation risk as a function of the regularization constant  $\lambda$  measured for the experiment modification of  $s_0$ . The optimal  $\lambda$  minimizing the validation risk is denoted in bold.

mainly for the purpose of the detector functionality. However, we calculate all measurements with this component for completeness.

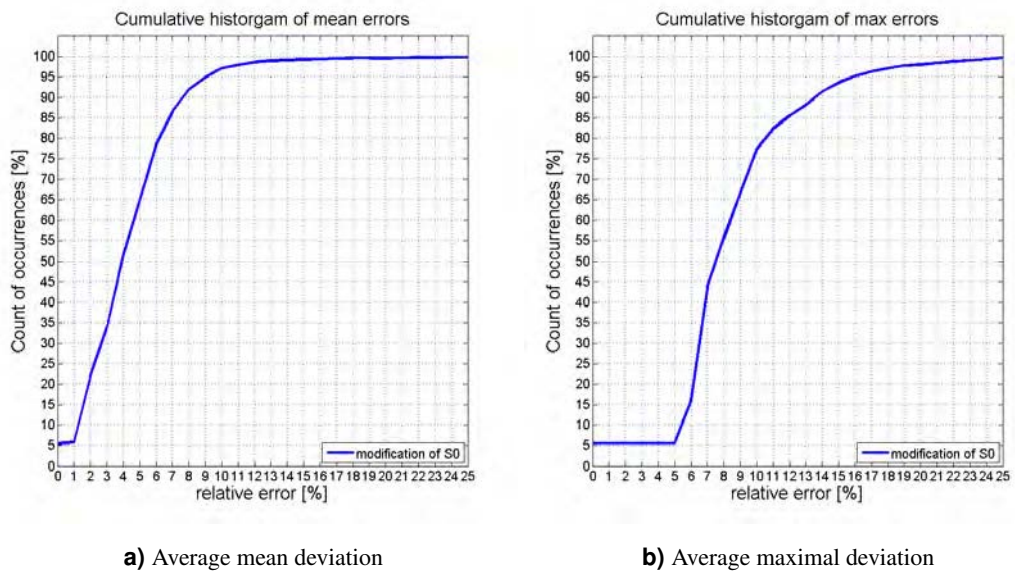
Table A.9 shows the normalized errors for each landmark as well as the average mean  $R_{\text{TST}}$  and average maximal  $R_{\text{TST}}^{\text{max}}$  deviation. Figure A.8 shows the cumulative histograms of the average mean and maximal deviation estimated on the test examples for the detector with parameters set as described in this experiment. Section 4.5 summarizes results of all experiments together.



**Figure A.6.** Image results for experiment: modification of  $s_0$ . The red squares are estimated landmarks. The green squares are the ground truth positions. Figure A.7d is the worst classified example from the TST set. Figures A.7e and A.7f may also serve for comparison with the previous experiments.



**Figure A.7.** The comparison of image results for the experiment structured output SVM with the displacement deformation cost and the experiment modification of  $s_0$ . The top row shows images from the displacement experiment, bottom row shows images from the modification of  $s_0$ .



**Figure A.8.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for the experiment modification of  $s_0$ .

A. Experimental tuning of the detector configuration

<b>Modification of <math>s_0</math></b>				
	Left eye $j = 1$	Right eye $j = 3$	Mouth $j = 2$	Nose $j = 0$
$R_{\text{TST}}^j$	4.0931	3.9484	5.2365	5.7556
$R_{\text{TST}}^{j\text{max}}$	41.0651	74.9429	80.5220	34.4904
$R_{\text{TST}}^{\text{max}}$				9.20465
$R_{\text{TST}}$				4.75839

**Table A.9.** Normalized errors of the experiment: modification of  $s_0$ .

## A.4. Features: Normalized image intensity values

In this experiment we build the proposed model with the deformation cost represented by a displacement (see section 3.1.2 for details). As the model of appearance we use the normalized image intensity values (see Section 3.1.1). We wrote a mex-file very similar to the one used for computation of the LBP pyramid features — which makes the modification of computation of the feature map  $\Psi_i^q(I, s_i)$  very convenient.

Learning of the joint parameter vector  $w$  for the  $\lambda = 0.01$  converged to precision  $\epsilon = 0.01$  in 2493 iterations. Overall training time (i.e. learning of the joint parameter vector  $w$  for all  $\lambda$ -values) took 6 days and 9 hours computed parallel in 8 threads. One iteration took less than 3 minutes. Note that the  $\lambda = 0.01$  may not be optimal, we should try learning also for the  $\lambda = 0.001$ . We omit this step according to the very high number of iterations of the last  $\lambda$ -value used.

### A.4.1. Parameters

Table A.10 shows the parameters settings for this experiment. Results of the validation depicts Table A.11. Optimal value of the regularization term  $\lambda$  was not found.

Base window	$[40, 40]^T$ px
Base window margin	$[20, 20]^T$ %
Components	$\begin{bmatrix} 13 & 13 & 20 & 13 \\ 13 & 13 & 13 & 13 \end{bmatrix}$ px

**Table A.10.** Parameters settings for experiment: normalized image intensity values features.

**Normalized image intensity values features**

$\lambda$	$R_{\text{TRN}}$	$R_{\text{VAL}}$
$10^{-2}$	0.8525	0.8669
$10^{-1}$	0.8713	0.8843
1	0.9655	0.9661
10	1.2198	1.2118

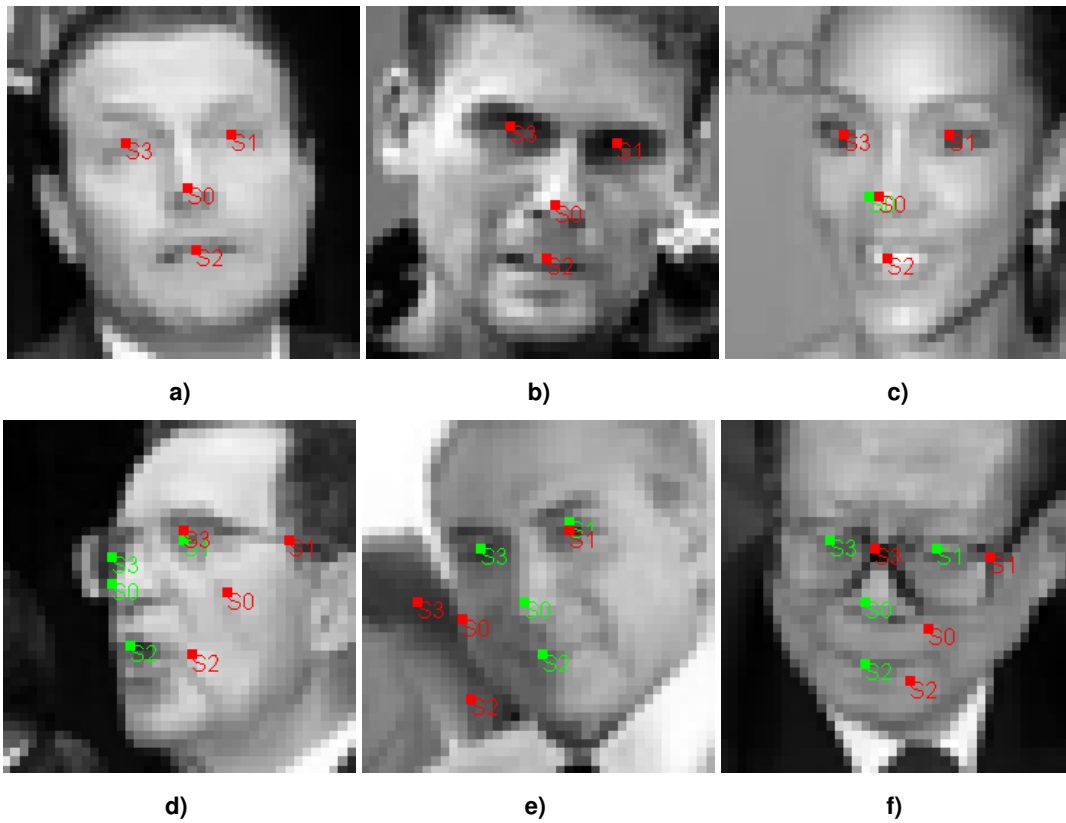
**Table A.11.** The training risk and validation risk as a function of the regularization constant  $\lambda$  measured for the experiment normalized image intensity values features. The optimal  $\lambda$  was not found.

### A.4.2. Results

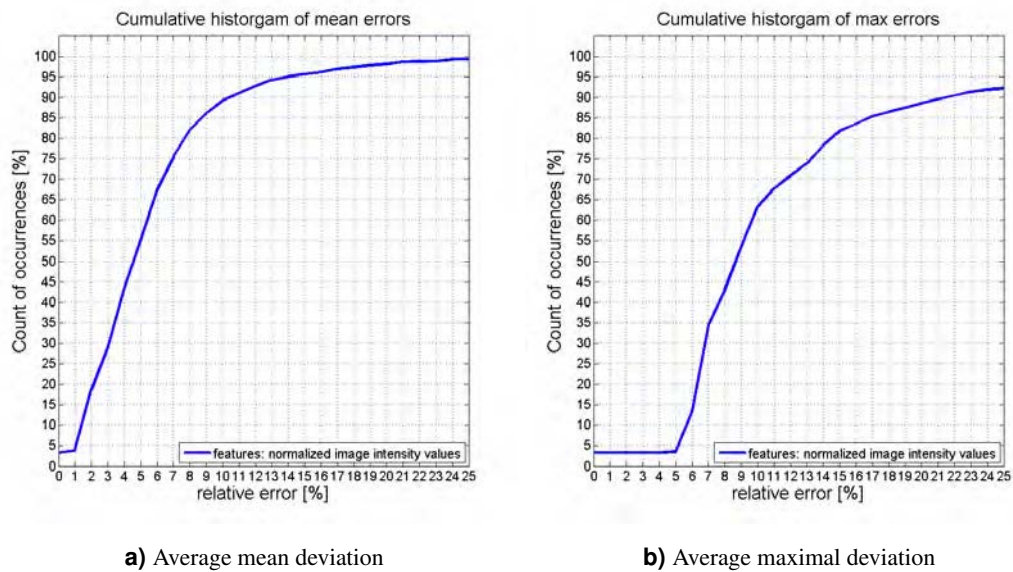
The normalized image intensity values used for computation of the feature map  $\Psi_i^q(I, s_i)$  provides quite good results, even though we do not know if the last used value of  $\lambda$  is optimal. The computation of these features is very fast and can be implemented even faster (e.g. with integral image). The main disadvantage of this features is the very high number of iterations of the BMRM in learning stage. Figure A.9 shows some randomly chosen examples from the test set.

Table A.12 shows the normalized errors for each landmark as well as the average mean  $R_{\text{TST}}$  and average maximal  $R_{\text{TST}}^{\text{max}}$  deviation. Figure A.10 shows the cumulative histograms of the average mean and maximal deviation estimated on the test examples for the detector with parameters set as described in this experiment. Section 4.5 summarizes results of all experiments together.

A. Experimental tuning of the detector configuration



**Figure A.9.** Image results for experiment: normalized image intensity values features. The red squares are estimated landmarks. The green squares are the ground truth positions. The top row of images shows some randomly chosen good results. The bottom row shows the worst results.



**Figure A.10.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for the experiment Normalized image intensity values features.

**Normalized image intensity values features**

	Left eye $j = 1$	Right eye $j = 3$	Mouth $j = 2$	Nose $j = 0$
$R_{TST}^j$	4.7045	4.8463	7.1165	7.4313
$R_{TST}^{j\max}$	108.5531	76.4199	69.1269	116.6190
$R_{TST}^{\max}$				12.14191
$R_{TST}$				6.02465

**Table A.12.** Normalized errors of the experiment: normalized image intensity values features.

## A.5. Features: Derivatives of image intensity values

In this experiment we build the proposed model with the deformation cost represented by a displacement (see section 3.1.2 for details). As the model of appearance we use the derivatives of image intensity values (see Section 3.1.1). Similarly as in the previous experiment we wrote a mex-file for the feature map  $\Psi_i^q(I, s_i)$  computation.

Learning of the joint parameter vector  $w$  for the  $\lambda = 0.01$  converged to precision  $\epsilon = 0.01$  in 6245 iterations. Overall training time (i.e. learning of the joint parameter vector  $w$  for all  $\lambda$ -values) took 19 days and 13 hours computed parallel in 8 threads. One iteration took about 3.5 minutes. Note that the  $\lambda = 0.01$  may not be optimal, we should try learning also for the  $\lambda = 0.001$ . We omit this step according to the very high number of iterations of the last  $\lambda$ -value used. This means that this type of features is not enough discriminative and therefore it is not very appropriate.

### A.5.1. Parameters

Table A.13 shows the parameters settings for this experiment. Results of the validation depicts Table A.14. Optimal value of the regularization term  $\lambda$  was not found.

Base window	$[40, 40]^T$ px
Base window margin	$[20, 20]^T$ %
Components	$\begin{bmatrix} 13 & 13 & 20 & 13 \\ 13 & 13 & 13 & 13 \end{bmatrix}$ px

**Table A.13.** Parameters settings for experiment: derivatives of image intensity values features.

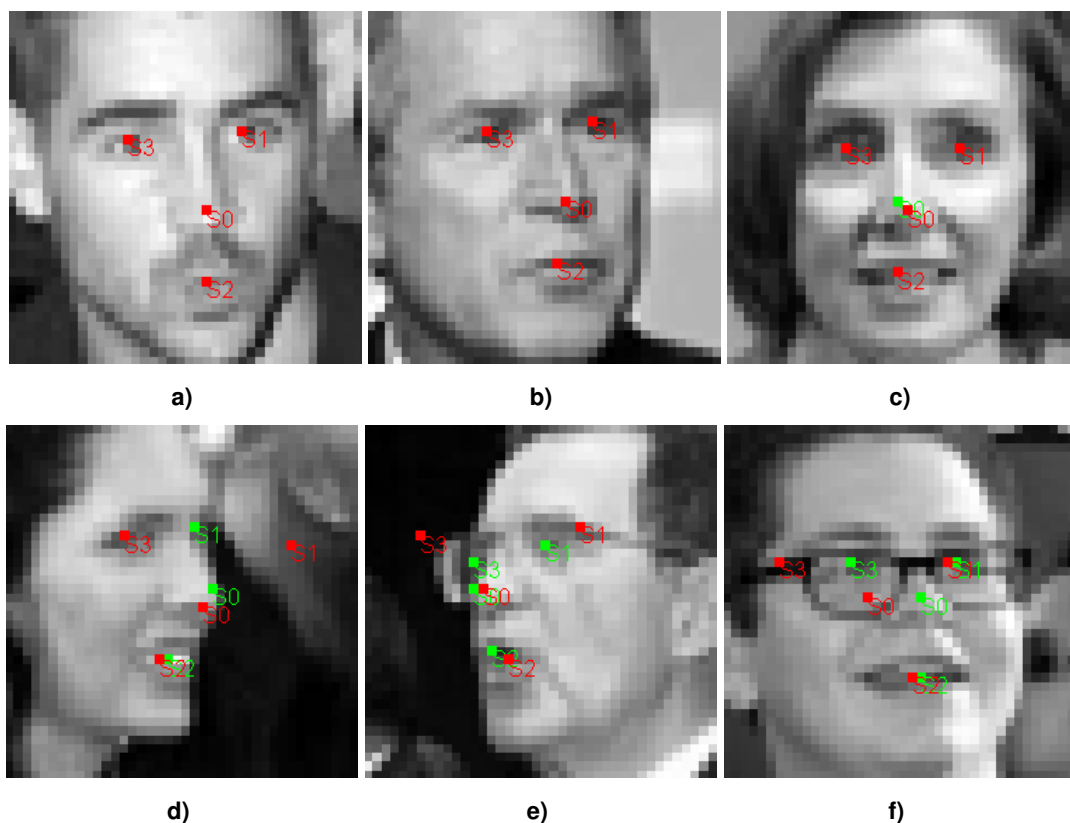
$\lambda$	$R_{\text{TRN}}$	$R_{\text{VAL}}$
$10^{-2}$	0.6508	0.6882
$10^{-1}$	0.6648	0.6947
1	0.7220	0.7290

**Table A.14.** The training risk and validation risk as a function of the regularization constant  $\lambda$  measured for the experiment derivatives of image intensity values features. The optimal  $\lambda$  was not found.

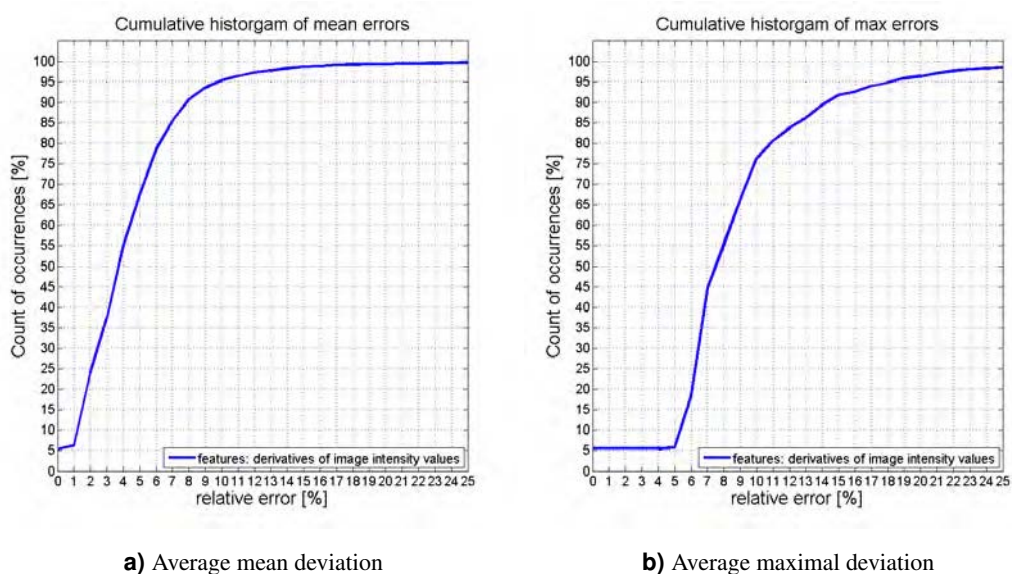
### A.5.2. Results

The derivatives of image intensity values used for computation of the feature map  $\Psi_i^q(I, s_i)$  provides very good results, even though we do not know if the last used value of  $\lambda$  is optimal. These features are fast and very easy to compute. Moreover there is still room for making it faster (more effective computation, parallelization, etc.). The main disadvantage of this features is the very high number of iterations of the BMRM in the learning stage. The Figure A.11 shows some randomly chosen examples from the TST set.

Table A.15 shows the normalized errors for each landmark as well as the average mean  $R_{\text{TST}}$  and average maximal  $R_{\text{TST}}^{\text{max}}$  deviation. Figure A.12 shows the cumulative histograms of the average mean and maximal deviation estimated on the test examples for the detector with parameters set as described in this experiment. Section 4.5 summarizes results of all experiments together.



**Figure A.11.** Image results for experiment: derivatives of image intensity values features. The red squares are estimated landmarks. The green squares are the ground truth positions. The top row of images shows some randomly chosen good results. The bottom row shows the worst results. Note that image A.11e is the same as in the previous experiment A.9d, but this time the detection is much better.



**Figure A.12.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for the experiment derivatives of image intensity values features.



A. Experimental tuning of the detector configuration

**Derivatives of image intensity values features**

	Left eye $j = 1$	Right eye $j = 3$	Mouth $j = 2$	Nose $j = 0$
$R_{TST}^j$	4.0210	3.9130	5.2195	5.9547
$R_{TST}^{j\max}$	76.9231	60.8229	63.2456	61.9715
$R_{TST}^{\max}$				9.85334
$R_{TST}$				4.77704

**Table A.15.** Normalized errors of the experiment: derivatives of image intensity values features.

## A.6. Features: LBP histogram

In this experiment we build the proposed model with the deformation cost represented by a displacement (see section 3.1.2 for details). As the model of appearance we use the LBP histogram (see Section 3.1.1). Similarly as in the previous experiment we wrote a mex-file for the feature map  $\Psi_i^q(I, s_i)$  computation.

Learning of the joint parameter vector  $w$  for the  $\lambda = 10^{-5}$  converged to precision  $\epsilon = 0.01$  in 423 iterations. Training time of the joint parameter vector  $w$  for all  $\lambda \in \{10^{-4}, 10^{-3}, \dots, 1\}$  took 19 hours computed parallel in 8 threads. One iteration took about 2 minutes. Note that the  $\lambda = 10^{-5}$  may not be optimal, we should try learning also for the  $\lambda = 10^{-6}$ . We omit this step according to results for  $\lambda = 10^{-5}$  which indicates poor discriminability of this type of features.

### A.6.1. Parameters

Table A.16 shows the parameters settings for this experiment. Results of the validation depicts Table A.17. Optimal value of the regularization term  $\lambda$  was not found.

Base window	$[40, 40]^T$ px
Base window margin	$[20, 20]^T$ %
Components	$\begin{bmatrix} 13 & 13 & 20 & 13 \\ 13 & 13 & 13 & 13 \end{bmatrix}$ px

**Table A.16.** Parameters settings for experiment: LBP histogram features.

#### LBP histogram features

$\lambda$	$R_{TRN}$	$R_{VAL}$
$10^{-5}$	1.9490	1.9324
$10^{-4}$	2.0885	2.0812
$10^{-3}$	2.5581	2.5991
$10^{-2}$	3.9915	4.0315
$10^{-1}$	4.4252	4.4536
1	5.3939	5.4023

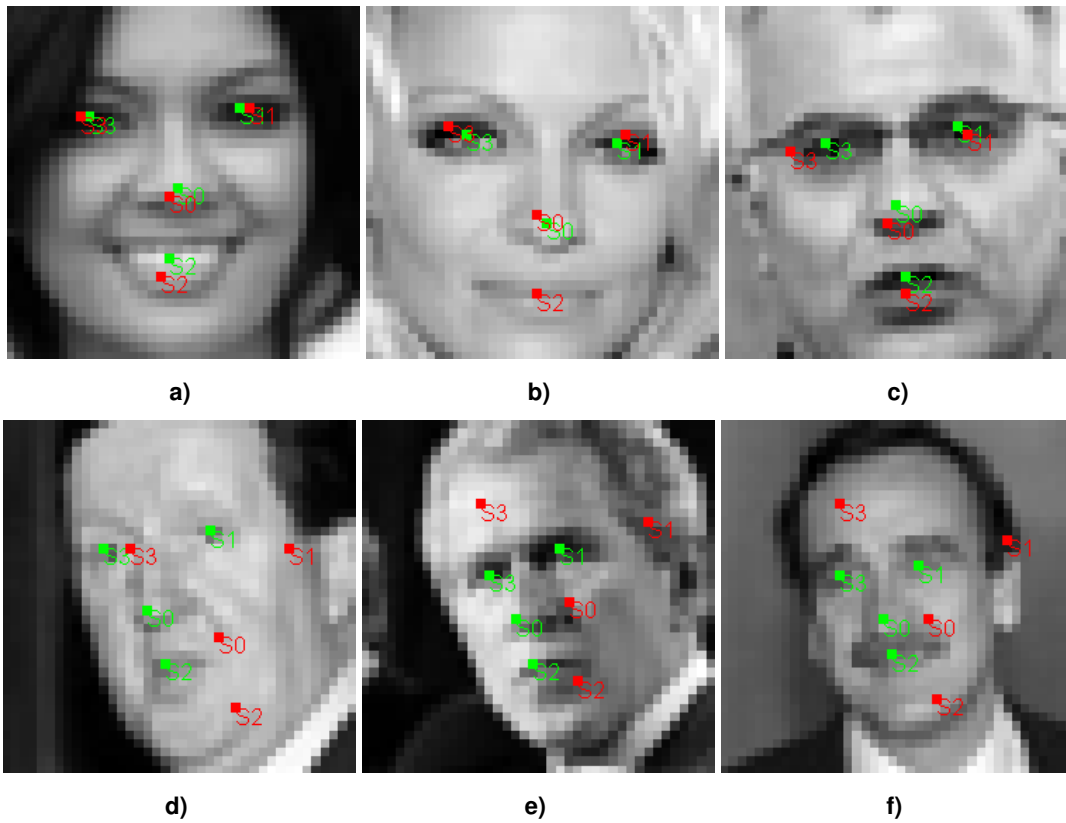
**Table A.17.** The training risk and validation risk as a function of the regularization constant  $\lambda$  measured for the experiment LBP histogram features. The optimal  $\lambda$  was not found.

### A.6.2. Results

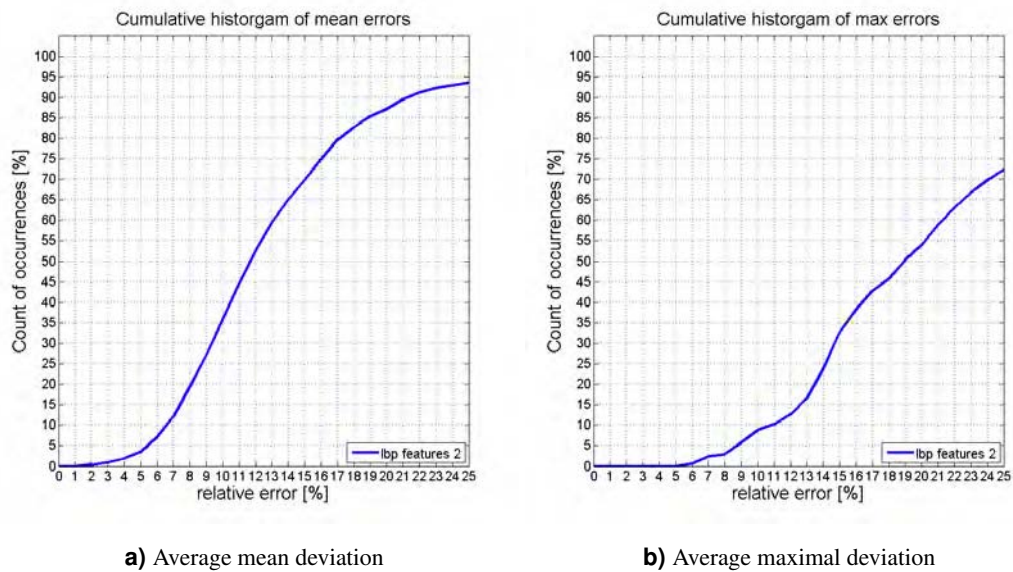
LBP histogram features computed in one scale are not very useful for the purpose of facial landmarks detection. The LBP pyramid features provides much better results. Figure A.13 shows the best (the top row of images) and the worst (the bottom row of images) classified examples from the test set. Note the quite poor quality of the best classified images in the top row.

Table A.18 shows the normalized errors for each landmark as well as the average mean  $R_{TST}$  and average maximal  $R_{TST}^{\max}$  deviation. Figure A.14 shows the cumulative histograms of the average mean and maximal deviation estimated on the test examples for the detector with parameters set as described in this experiment. Section 4.5 summarizes results of all experiments together.

A. Experimental tuning of the detector configuration



**Figure A.13.** Image results for experiment: LBP histogram features. The red squares are estimated landmarks. The green squares are the ground truth positions. The top row of images shows the best classified results. The bottom row shows the worst results.



**Figure A.14.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for the experiment LBP histogram features.

**LBP histogram features**

	Left eye $j = 1$	Right eye $j = 3$	Mouth $j = 2$	Nose $j = 0$
$R_{TST}^j$	13.4445	13.1180	13.7387	15.1231
$R_{TST}^{j\max}$	79.4966	105.1177	80.1623	114.3179
$R_{TST}^{\max}$				22.48238
$R_{TST}$				13.85607

**Table A.18.** Normalized errors of the experiment: LBP histogram features.

## A.7. Features: HOG

In this experiment we build the proposed detector all the same as in previous experiments with modification of the appearance model  $q_i(I, s_i)$ . We now use the HOG features (see 3.1.1) for the appearance model. Similarly as in previous experiments we wrote a mex-file for computation of HOG features.

Learning of the joint parameter vector  $w$  for the  $\lambda = 10^{-3}$  covered to precision  $\epsilon = 0.01$  in 1102 iterations. Training time for all  $\lambda \in \{10^{-3}, \dots, 1\}$  took 5 days and 8.5 hours computed parallel in 8 threads. One iteration took about 5 minutes. Note that  $\lambda = 10^{-3}$  may not be optimal, we should also try learning with  $\lambda = 10^{-4}$ . Since this is the last type of features we used for the appearance model, we were not able to finish the  $\lambda$  tuning of this experiment in time.

### A.7.1. Parameters

Table A.19 shows the parameters settings for this experiment. Results of the validation depicts Table A.20. Optimal value of regularization term  $\lambda$  was not found.

Base window	$[40, 40]^T$ px
Base window margin	$[20, 20]^T$ %
Components	$\begin{bmatrix} 13 & 13 & 20 & 13 \\ 13 & 13 & 13 & 13 \end{bmatrix}$ px

**Table A.19.** Parameters settings for experiment: HOG features.

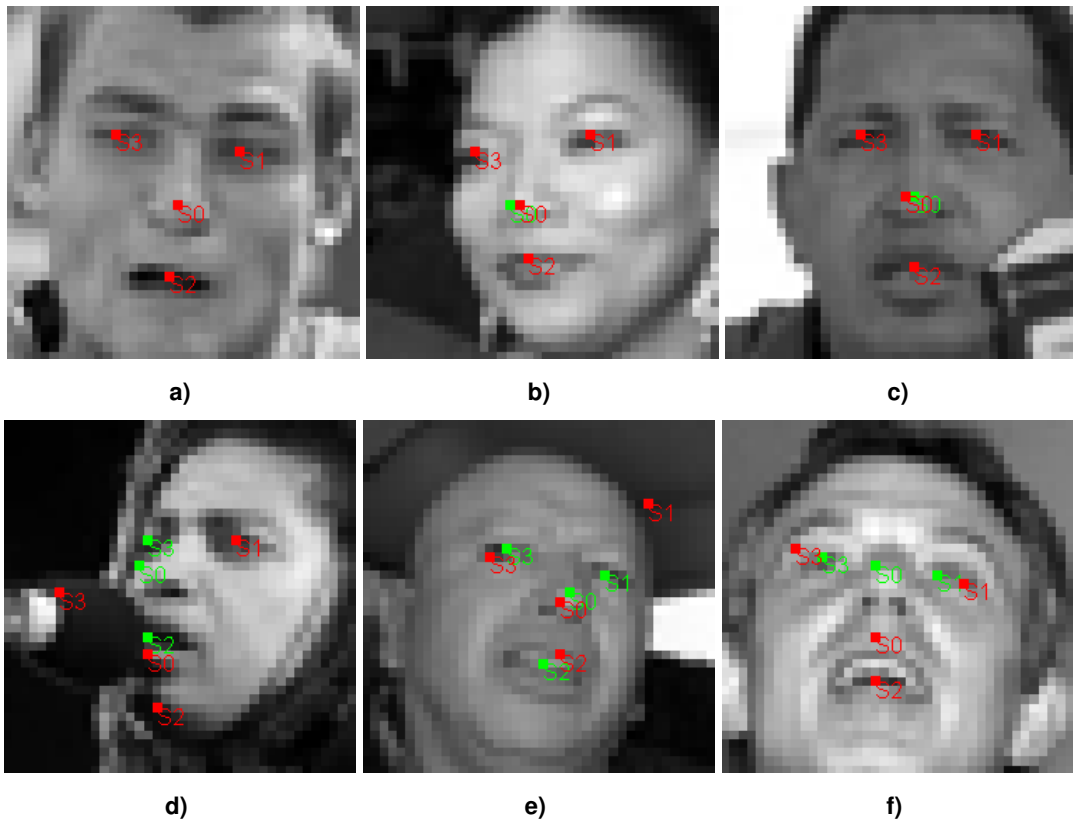
$\lambda$	$R_{\text{TRN}}$	$R_{\text{VAL}}$
$10^{-3}$	0.87671	0.8858
$10^{-2}$	0.92979	0.9370
$10^{-1}$	1.03225	1.0337
1	1.38325	1.3562

**Table A.20.** The training risk and validation risk as a function of the regularization constant  $\lambda$  measured for the experiment HOG features. Optimal value of  $\lambda$  was not found.

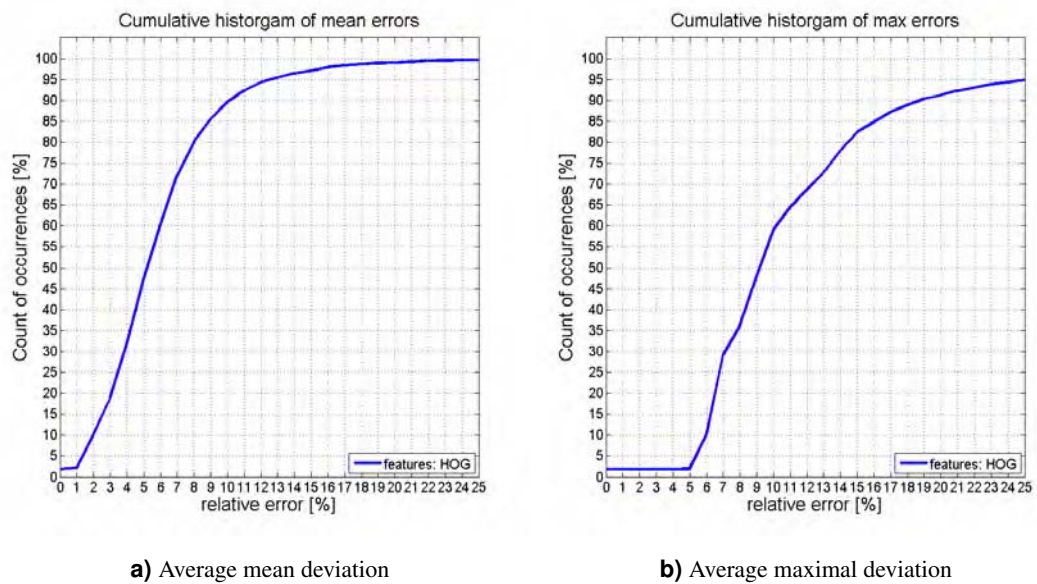
### A.7.2. Results

The HOG features gives very promising results. Figure A.15 shows some randomly chosen image results of detector with parameter settings described in this experiment.

Table A.21 shows the normalized errors for each landmark as well as the average mean  $R_{\text{TST}}$  and average maximal  $R_{\text{TST}}^{\text{max}}$  deviation. Figure A.16 shows the cumulative histograms of the average mean and maximal deviation estimated on the test examples for the detector with parameters set as described in this experiment. Section 4.5 summarizes results of all experiments together.



**Figure A.15.** Image results for experiment: HOG features. The red squares are estimated landmarks. The green squares are the ground truth positions. In the top row are randomly chosen good results, in the bottom row are the worst results.



**a)** Average mean deviation

**b)** Average maximal deviation

**Figure A.16.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for the experiment HOG features.

A. Experimental tuning of the detector configuration

**HOG features**

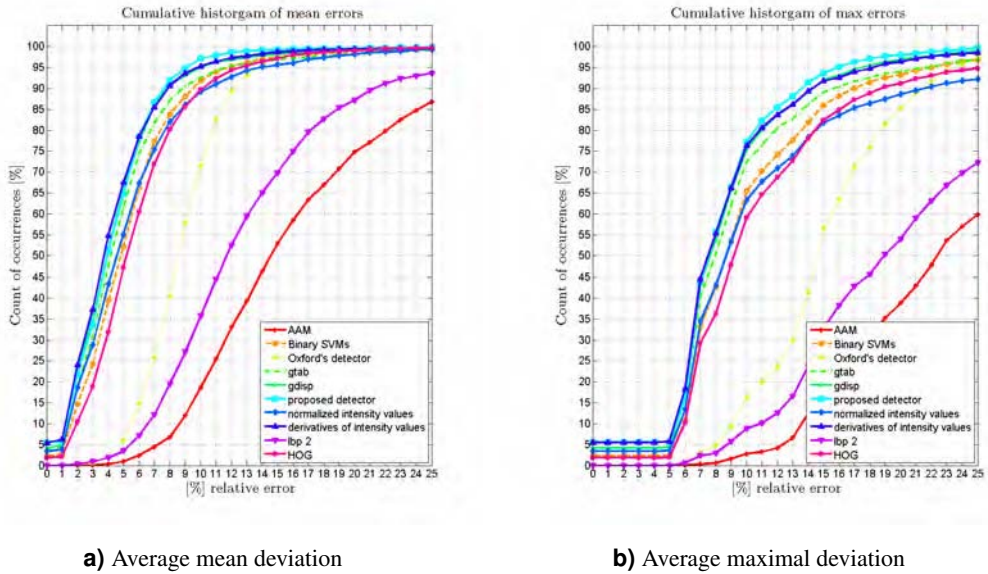
	Left eye $j = 1$	Right eye $j = 3$	Mouth $j = 2$	Nose $j = 0$
$R_{TST}^j$	5.2875	5.7199	6.6347	7.6876
$R_{TST}^{j\max}$	81.3456	96.5146	70.6510	83.1734
$R_{TST}^{\max}$	6.33241			
$R_{TST}$	12.08693			

**Table A.21.** Normalized errors of the experiment: HOG features.

## A.8. Summary of all experiments

In this section we summarize results of all experiments in order to choose the best detector with optimal parameter settings. Tables A.23 and A.24 shows results of all detectors built by instructions of individual experiments.

In Figure A.17 you can see the cumulative histograms for all experiments, including the baselines. For clarity we show Table A.22, where you can find the detail around 10% of relative error of Figure A.17. Experiments results shows that the best performing detector is the detector described in Section A.3.



**Figure A.17.** Cumulative histograms of the average (a) and maximal (b) deviations estimated on the test examples for all experiments.

**Detail around 10% of normalized error of relative error of Figure A.17**

	Average mean deviation	Average maximal deviation
AAM	18.57 %	2.831 %
Binary SVMs	91.91 %	62.53 %
Oxford's detector	71.63 %	16.20 %
structured output SVM - gtab	92.58 %	72.48 %
structured output SVM - gdisp	95.05 %	75.55 %
modification of $s_0$	97.15 %	77.25 %
normailzed intensity values	89.19 %	63.34 %
derivatives of intensity values	95.31 %	76.36 %
LBP 2	35.71 %	8.785 %
HOG	89.69 %	59.07 %

**Table A.22.** Detail around 10% of relative error of Figure A.17. The values are percents of all test examples that have error less or equal to 10 %.



A. Experimental tuning of the detector configuration

Summary of all experiments — mean errors

	AAM	Binary SVMs	Oxford	gtab	gdisp	change of $s_0$	simple features	simple features 2	LBP 2	HOG
$R_{TST}^{\text{left eye}}$	17.1167	5.3333	6.5028	4.8684	4.2234	4.0931	4.7045	4.0210	13.4445	5.2875
$R_{TST}^{\text{right eye}}$	16.4095	5.2212	5.8537	4.8974	4.5212	3.9484	4.8463	3.9130	13.1180	5.7199
$R_{TST}^{\text{mouth}}$	16.9982	5.9941	12.5138	5.3685	4.9546	5.2365	7.1165	5.2195	13.7387	6.6347
$R_{TST}^{\text{nose}}$	17.1284	7.0347	12.2694	6.7003	6.0083	5.7556	7.4313	5.9547	15.1231	7.6876
$R_{TST}$	16.91322	5.89579	9.28491	5.45866	4.92684	4.75839	6.02465	4.77704	13.85607	6.33241

**Table A.23.** Summary of mean errors of all experiments. Average mean deviation for each landmark  $R_{TST}^j$  is computed as defined in (4.7) of Algorithm 1.  $R_{TST}$  is defined in (4.3). We call the  $s_0$  nose, but in the experiment: modification of  $s_0$  is this component rather the center of the face. All values are in percents of error relative to distance between the center of eyes and mouth of the ground truth. The columns labeled gtab, gdisp, simple features, simple features 2 and LBP 2 refer to the structured output SVM with table deformation cost, displacement deformation cost, normalized image intensity values, derivatives of image intensity values and LBP histogram features.

Summary of all experiments — maximal errors

	AAM	Binary SVMs	Oxford	gtab	gdisp	change of $s_0$	simple features	simple features 2	LBP 2	HOG
$R_{TST}^{\text{max left eye}}$	100.3249	66.6667	44.7214	100.2596	81.3456	41.0651	108.5531	76.9231	79.4966	81.3456
$R_{TST}^{\text{max right eye}}$	89.0327	96.5146	52.1536	96.5146	80.0000	74.9429	76.4199	60.8229	105.1177	96.5146
$R_{TST}^{\text{max mouth}}$	70.6225	64.4465	37.9987	74.2781	73.2177	80.5220	69.1269	63.2456	80.1623	70.6510
$R_{TST}^{\text{max nose}}$	65.4023	77.2270	77.2496	115.7292	76.8662	34.4904	116.6190	61.9715	114.3179	83.1734
$R_{TST}^{\text{max}}$	25.77897	11.67883	15.98571	11.39167	10.10671	9.20465	12.14191	9.85334	22.48238	12.08693

**Table A.24.** Summary of maximal errors of all experiments. Average maximal deviation for each landmark  $R_{TST}^{\text{max}}$  is computed as defined in (4.9) of Algorithm 1.  $R_{TST}^{\text{max}}$  is defined in (4.8). We call the  $s_0$  nose, but in the experiment: modification of  $s_0$  is this component rather the center of the face. All values are in percents of error relative to distance between the center of eyes and mouth of the ground truth. The columns labeled gtab, gdisp, simple features, simple features 2 and LBP 2 refer to the structured output SVM with table deformation cost, displacement deformation cost, normalized image intensity values, derivatives of image intensity values and LBP histogram features.

## B. CD Contents

```
|-- Data           Some example images with detected faces
|-- Demo          Image and video demonstration of functionality
                  of our detector

|  |-- Images
|  `-- Video
|-- Doc           This thesis in .pdf format
`-- flandmark    Open source library implementing the facial
                  landmark detection

    |-- cpp      C source files
    |-- data     MAT-files and some example images
    |  |-- Images
    |-- Functions MATLAB functions
    |-- learning MATLAB scripts for learning
    |  |-- gdisp
    |  |-- gtab
    |  `-- mod_S0
    `-- mex      mex-files generated for 64bit Windows and
                  Linux operating systems
```



## Bibliography

- [Ahonen et al., 2004] Ahonen, T., Hadid, A., and Pietikäinen, M. (2004). Face recognition with local binary patterns. In Pajdla, T. and Matas, J., editors, *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 469–481. Springer Berlin / Heidelberg. 14
- [Beumer et al., 2006] Beumer, G., Tao, Q., Bazen, A., and Veldhuis, R. (2006). A landmark paper in face recognition. In *7th International Conference on Automatic Face and Gesture Recognition (FGR-2006)*. IEEE Computer Society Press. 10
- [Beumer and Veldhuis, 2005] Beumer, G. and Veldhuis, R. (2005). On the accuracy of EERs in face recognition and the importance of reliable registration. In *5th IEEE Benelux Signal Processing Symposium (SPS-2005)*, pages 85–88. IEEE Benelux Signal Processing Chapter. 7
- [Bordes et al., 2009] Bordes, A., Bottou, L., and Gallinari, P. (2009). Sgd-qn: Careful quasi-newton stochastic gradient descent. *Journal of Machine Learning Research*, 10:1737–1754. 8, 17
- [Cootes et al., 2001] Cootes, T., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685. 9
- [Crandall et al., 2005] Crandall, D., Felzenszwalb, P., and Huttenlocher, D. (2005). Spatial priors for part-based recognition using statistical models. In *In CVPR*, pages 10–17. 10
- [Cristinacce and Cootes, 2003] Cristinacce, D. and Cootes, T. (2003). Facial feature detection using adaboost with shape constraints. In *14th Proceedings British Machine Vision Conference (BMVC-2003)*, pages 231–240. 10
- [Cristinacce et al., 2004] Cristinacce, D., Cootes, T., and Scott, I. (2004). A multi-stage approach to facial feature detection. In *15th British Machine Vision Conference (BMVC-2004)*, pages 277–286. 7
- [Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893. 15
- [Erukhimov and Lee, 2008] Erukhimov, V. and Lee, K. (2008). A bottom-up framework for robust facial feature detection. In *8th IEEE International Conference on Automatic Face and Gesture Recognition (FG2008)*, pages 1–6. 10
- [Everingham et al., 2006] Everingham, M., Sivic, J., and Zisserman, A. (2006). “Hello! My name is... Buffy” – automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference*. 8, 10, 19
- [Everingham et al., 2008] Everingham, M., Sivic, J., and Zisserman, A. (2008). Willow project, automatic naming of characters in tv video. MATLAB implementation, [www: http://www.robots.ox.ac.uk/~vgg/research/nface/index.html](http://www.robots.ox.ac.uk/~vgg/research/nface/index.html). 23

## Bibliography

- [Everingham et al., 2009] Everingham, M., Sivic, J., and Zisserman, A. (2009). Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 27(5). 8, 10, 19
- [Felzenszwalb et al., 2009] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2009). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(1). 10, 15
- [Felzenszwalb and Huttenlocher, 2005] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79. 10
- [Fischler and Elschlager, 1973] Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1):67–92. 7, 10
- [Franc and Sonnenburg, 2010] Franc, V. and Sonnenburg, S. (2010). Libocas — library implementing ocas solver for training linear svm classifiers from large-scale data. [www: http://cmp.felk.cvut.cz/~xfrancv/ocas/html/index.html](http://cmp.felk.cvut.cz/~xfrancv/ocas/html/index.html). 15, 21
- [Heikkilä et al., 2009] Heikkilä, M., Pietikäinen, M., and Schmid, C. (2009). Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436. 14
- [Huang et al., 2007] Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst. 19
- [Kroon, 2010] Kroon, D.-J. (2010). Active shape model (ASM) and active appearance model (AAM). MATLAB implementation, [www: http://www.mathworks.com/matlabcentral/fileexchange/26706-active-shape-model-asm-and-active-appearance-model-aam](http://www.mathworks.com/matlabcentral/fileexchange/26706-active-shape-model-asm-and-active-appearance-model-aam). 8, 22
- [Matas et al., 2010] Matas, J., Chum, O., and Svoboda, T. (2010). Lectures for course computer vision methods. [www: https://cw.felk.cvut.cz/doku.php/courses/ae4m33mpv/start](http://www: https://cw.felk.cvut.cz/doku.php/courses/ae4m33mpv/start). 14
- [Nordstrøm et al., 2004] Nordstrøm, M. M., Larsen, M., Sierakowski, J., and Stegmann, M. B. (2004). The IMM face database - an annotated dataset of 240 face images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU. 23
- [OMRON, 2011] OMRON, g. (2011). Okao vision. [www: http://www.omron.com/r\\_d/coretech/vision/okao.html](http://www.omron.com/r_d/coretech/vision/okao.html). 7
- [Riopka and Boulton, 2003] Riopka, T. and Boulton, T. (2003). The eyes have it. In *Proceedings of ACM SIGMM Multimedia Biometrics Methods and Applications Workshop*, pages 9–16. 7
- [Sivic et al., 2009] Sivic, J., Everingham, M., and Zisserman, A. (2009). “Who are you?” – learning person specific classifiers from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8, 10, 19
- [Stegmann, 2007] Stegmann, M. B. (2007). Active appearance models. Master’s thesis, IIM, Technical University of Denmark.

- [Teo et al., 2010] Teo, C. H., Vishwanthan, S., Smola, A. J., and Le, Q. V. (2010). Bundle methods for regularized risk minimization. *J. Mach. Learn. Res.*, 11:311–365. 8, 16
- [Tsochantaridis et al., 2005] Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., and Singer, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484. 7, 16
- [Viola and Jones, 2004] Viola, P. and Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154. 9, 11
- [Wu and Trivedi, 2005] Wu, J. and Trivedi, M. (2005). Robust facial landmark detection for intelligent vehicle system. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*. 10

