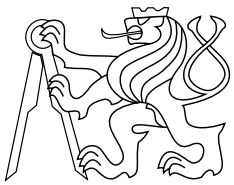
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

PhD THESIS

ISSN 1213-2365

Multi-view Facial Landmark Detection
(PhD Thesis)

Michal Uřičář

Study Programme: Electrical Engineering and Information Technology

Branch of Study: Artificial Intelligence and Biocybernetics

uricamic@cmp.felk.cvut.cz

CTU–CMP–2017–03

April 7, 2017

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/uricar/Uricar-PhD-2016.pdf

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

The author was supported by EC projects FP7-ICT-247525 HUMAVIPS and PERG04-GA-2008-239455 SEMISOL, the Technology Agency of the Czech Republic under the Project TE01020197 Center for Applied Cybernetics, the Czech Science Foundation Project GACR P103/12/G084, and the Grant Agency of the CTU under the project SGS15/201/OHK3/3T/13.

Research Reports of CMP, Czech Technical University in Prague, No. 3, 2017

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

Multi-view Facial Landmark Detection

Michal Uřičář

April 7, 2017

Contents

Abstract

In this thesis, we tackle the problem of designing a multi-view facial landmark detector which is robust and works in real-time on low-end hardware. Our landmark detector is an instance of the structured output classifiers describing the face by a mixture of tree based Deformable Part Models (DPM). We propose to learn parameters of the detector by the Structured Output Support Vector Machine algorithm which, in contrast to existing methods, directly optimizes a loss function closely related to the standard evaluation metrics used in landmark detection. We also propose a novel two-stage approach to learn the multi-view landmark detectors, which provides better localization accuracy and significantly reduces the overall learning time. We propose several speedups that enable to use the globally optimal prediction strategy based on the dynamic programming in real time even for dense landmark sets. The empirical evaluation shows that the proposed detector is competitive with the current state-of-the-art both regarding the accuracy and speed.

We also propose two improvements of the Bundle Method for Regularized Risk Minimization (BMRM) algorithm which is among the most popular batch solvers used in structured output learning. First, we propose to augment the objective function by a quadratic prox-center whose strength is controlled by a novel adaptive strategy preventing zig-zag behavior in the cases when the genuine regularization term is weak. Second, we propose to speed up convergence by using multiple cutting plane models which better approximate the objective function with minimal increase in the computational cost. Experimental evaluation shows that the new BMRM algorithm which uses both improvements speeds up learning up to an order of magnitude on standard computer vision benchmarks, and 3 to 4 times when applied to the learning of the DPM based landmark detector.

Abstrakt

V této tezi se zabýváme návrhem více-pohledového detektoru významných bodů na lidské tváři, který je robustní a funguje v reálném čase, a to i na hardware nižší třídy. Námi navržený detektor významných bodů je instancí strukturálního klasifikátoru, založeného na popisu tváře pomocí směsi "Deformable Part Models" se stromovou strukturou. Parametry detektoru se učí pomocí algoritmu "Structured Output Support Vector Machines", který na rozdíl od existujících metod dokáže optimalizovat ztrátovou funkci přímo související s metrikou používanou pro vyhodnocování přesnosti detektorů významných bodů. V tezi navrhujeme nový dvou-fázový algoritmus pro učení více-pohledových detektorů významných bodů, který dosahuje vyšší přesnosti lokalizace a zároveň významně snižuje celkovou dobu učení. Kromě toho navrhujeme několik urychlení detekčního algoritmu, díky nimž lze predikovat polohu i husté množiny významných bodů v reálném čase za pomoci metod globální optimalizace. Empirické vyhodnocení ukazuje, že navrhovaný detektor je porovnatelný s nejmodernějšími detektory, co se týče přesnosti i rychlosti detekce.

Dále navrhujeme dvě vylepšení algoritmu "Bundle Methods for Regularized Risk Minimization (BMRM)" patřícího mezi nejpopulárnější dávkové metody pro učení strukturálních klasifikátorů. Zaprvé navrhujeme rozšířit účelovou funkci o pomocný kvadratický člen, jehož váha je kontrolována novou adaptivní strategií, která zabraňuje nestabilní konvergenci ("cik-cak" chování) v případě, kdy původní kvadratický regularizační člen je příliš slabý. Zadruhé navrhujeme urychlení konvergence použitím vícenásobného modelu odsekávájících nadrovin, který lépe aproximuje účelovou funkci, a to s minimálním nárůstem výpočetní náročnosti. Experimentální vyhodnocení ukazuje, že nový BMRM algoritmus využívající obě navržená vylepšení zrychluje učení až o jeden řád na standardních srovnávácích sadách a 3–4 krát při aplikaci na učení navrhovaného detektoru významných bodů.

Acknowledgement

I am grateful to my advisor Ing. Vojtěch Franc, Ph.D., and my former advisor prof. Ing. Václav Hlaváč, CSc. It has been a pleasure to work with them and learn from their experience. I would also like to thank my colleagues and friends from the Center of Machine Perception for their support and fruitful discussions. Last but not least, I would like to thank my family for their endless support.

I am grateful I could be the intern at the National Institute of Informatics in Tokyo, Japan, where I worked under the supervision of prof. Akihiro Sugimoto in his lab. It was a great experience and opportunity.

I gratefully acknowledge that my research was supported by EC projects FP7-ICT-247525 HUMAVIPS and PERG04-GA-2008-239455 SEMISOL, the Technology Agency of the Czech Republic under the Project TE01020197 Center for Applied Cybernetics, the Czech Science Foundation Project GACR P103/12/G084, and the Grant Agency of the CTU under the project SGS15/201/OHK3/3T/13.

Authorship

I hereby certify that the results presented in this thesis were achieved during my own research in cooperation with my thesis advisor Ing. Vojtěch Franc, Ph.D. and my former advisor prof. Ing. Václav Hlaváč, CSc.

Abbreviations

AAM	Active Appearance Models ... 11
AFLW	Annotated Facial Landmarks in the Wild ... 58
BMRM	Bundle Method for Regularized Risk Minimization ... 8
C-DPM	Coarse DPM
C2F-DPM	Coarse-to-fine DPM
CLM	Constraint Local Models ... 17
CP	Cutting Plane ... 47
DP	Dynamic Programming ... 20
DPM	Deformable Part Based Models ... 3
DT	Distance Transform ... 21
F-DPM	Fine DPM
HOG	Histogram of Oriented Gradients ... 29
IOD	Inter-ocular Distance
LBP	Local Binary Patterns ... 9
LFW	Labeled Faces in the Wild ... 3
P-SGD	Projected Stochastic Gradient Descent ... 46
PCA	Principal Component Analysis ... 11
S-LBP	Sparse Pyramid of Local Binary Patterns ... 9
SIFT	Scale Invariant Feature Transform ... 19
SO-SVM	Structured Output Support Vector Machine ... 3
SGD	Stochastic Gradient Descent ... 41
SVM	Support Vector Machine ... 56

"It does not matter how slowly you go as long as you do not stop."

– Confucius

1. Introduction

Accurate localization of facial landmarks is an important topic in computer vision, which is increasingly getting more and more attention nowadays. As a crucial pre-processing step, the localization of facial landmarks became an integral part of a facial recognition/processing pipeline. In this work, we focus on the real-time and multi-view facial landmarks detection. While the former is an essential requirement for all time-dependent face processing tasks, which are becoming more frequent thanks to the emerging applications for mobile and embedded systems, the latter is necessary for operation in unconstrained ("in-the-wild") environments. We should also mention, that while the problem of the near-frontal facial landmark detection is covered quite thoroughly in the existing literature, the multi-view scenario is just slowly getting into the attention. This is mainly thanks to the increasing demand for applications working in unconstrained conditions.

The importance of the topic is also apparent in the growing number of related papers presented at the top computer vision conferences. See Figure 1.1 for more details.

There are numerous applications of the facial landmark detection, see Figure 1.2 for illustration. The usage of facial landmark detector is both direct and indirect. Typical representatives of the applications using the output of a facial landmark detector directly are for example: facial expressions analysis [Valstar et al., 2015], where usually the Facial Action Coding System [Ekman and Friesen, 1978] is used to decode the facial expression corresponding to a particular emotion or non-verbal message; marker-less motion capture [Thies et al., 2016], where the facial landmarks are used to aid the Computer Generated Imagery, primarily to transfer the facial expressions from human actors to generated models, in the movie making industry, or simply to create facial expressions for avatars, of the augmented reality in the Human-Computer Interaction. To the category of the indirect applications of facial landmark detection belong all applications where the facial landmarks are used for some pre-processing, for example: face registration [Trigeorgis et al., 2016] is usually performed prior face recognition task, where it often leads to increased accuracy of the follow-up decision problem like the identity recognition [Kemelmacher-Shlizerman et al., 2016]; 3D face reconstruction [Roth et al., 2016], where, for instance, the landmarks are used to aid the structure from motion algorithm; head-pose orientation [Čech et al., 2015] where a 3D face model is fitted to estimated 2D landmark positions; face tracking; other face processing tasks like prediction of gender, age, expression, or other facial attributes [Escalera et al., 2016; Uřičář et al., 2016b].

Detection of facial landmarks in uncontrolled environments is a non-trivial problem for several reasons. The key factor is a large intra-class variability of the input image due to the change of position, scale, and rotation of the face (being itself a 3D object projected to 2D image), lighting conditions, background clutter, facial expression, occlusions, and self-occlusions, hair style, make-up, race, aging, modality (webcam, camera, scanned image) and so on. Despite considerable progress in the last years, the problem in its full generality remains unsolved.

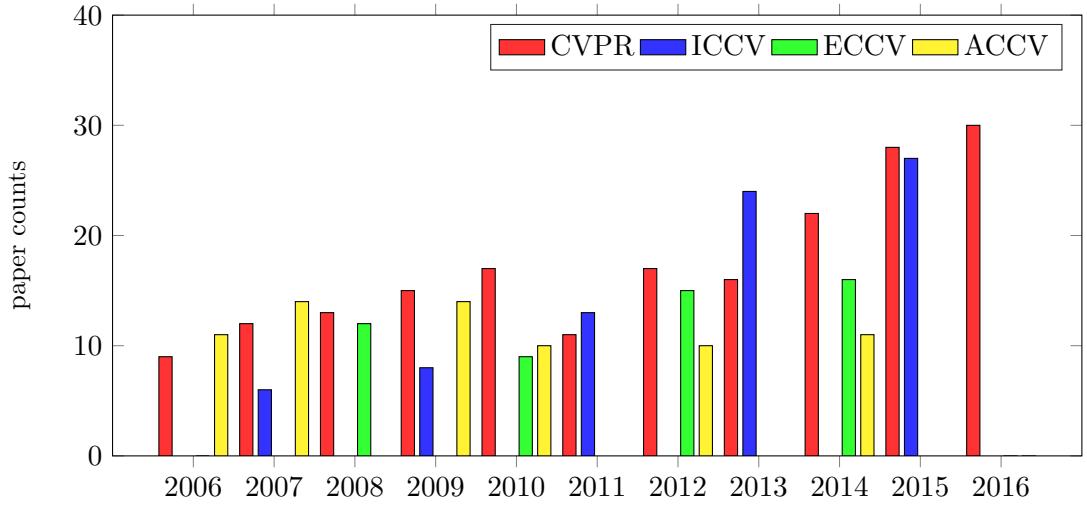


Figure 1.1. Counts of papers relevant to the thesis topic on top computer vision conferences in the last decade. We collected these numbers by manual inspection of the conference proceedings, based on the keywords search, such as "facial", "landmarks", "fiducial points", etc.

1.1. Problem Formulation

In this section, we outline the requirements on a facial landmark detector which we attempt to fulfill. The requirements are motivated by our experience with solving a set of real-life problems. The requirements concern the input of the detector, its output, its processing time, as well as its ability to adapt to a new problem.

We assume, that the landmark detector is provided with a facial image along with an estimate of the face location, scale, and in-plane rotation. Only a rough estimate of the location, scale, and in-plane rotation is required. In other words, an output of a present-day face detectors should be sufficient to generate the input for the landmark detector. The problem of the face detection itself is not tackled in this thesis.

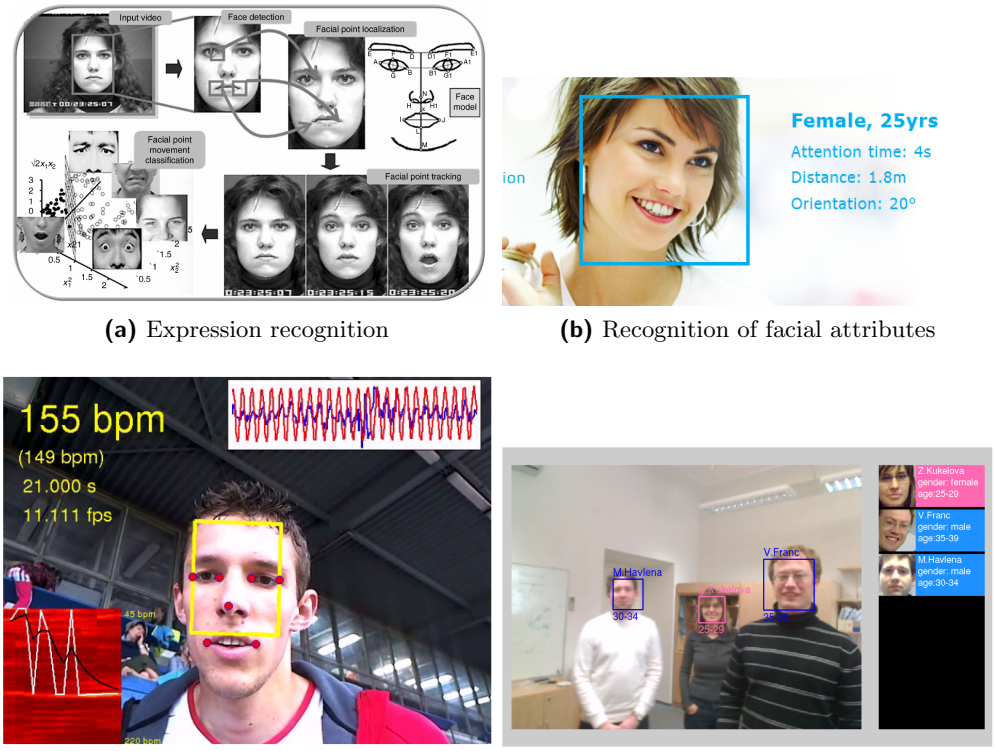
In this thesis, we aim at the multi-view landmark detector. The landmark detector should be able to output a rough estimate of the viewpoint (or the yaw angle of the face in 3D), and a subset of landmarks visible from that viewpoint. For most applications, it is sufficient if the detector is operable in the range of yaw from $-90°$ to $90°$. Also, the landmark detector should be robust against the rotation of face along its lateral axis, or the pitch yaw. Pitch in the range from $-45°$ to $45°$ is sufficient to cover the majority of standard facial images. Figure 1.3, and 1.4 show exemplary inputs and outputs of the required facial landmark detector.

Formally, we treat the multi-view facial landmark detector as an instance of the structured output classifier

$$h\colon \mathcal{I} \to \mathcal{S} \times \Phi\ , \tag{1.1}$$

where $\mathcal{I}$ is a set of possible facial images (as described above), Φ is a set of discretized yaw angles (rough estimate of the viewpoint), and $\mathcal{S}$ is a set of all 2D configurations of facial landmarks visible in all estimated views. In many existing approaches, the evaluation of detector (1.1) leads to an optimization problem for which no global solver exists. On the contrary, in this thesis, we concentrate on a class of detectors allowing to find a global solution to the prediction problem efficiently.

Evaluation time of the detector is a further determining characteristic in many applications. A short decision time is sometimes more important than the accuracy of landmark predictions. In this work we aim at a detector operating in real-time on



(a) Expression recognition **(b)** Recognition of facial attributes

(c) Heart beat rate estimation **(d)** Identity recognition

Figure 1.2. Facial landmarking applications.

standard computers, that is, no special hardware like an up-to-date dedicated Graphics Processing Unit (GPU) is required.

Last but not least, we aim at the framework that is flexible enough to be applicable for different landmark configurations (sparse, or dense landmark sets), and various types of input images. A design of new instances of the landmark detector should require a reasonable effort of a human expert while most of the work should be done automatically by learning from fully annotated examples.

The requirements outlined above lead us to choose tree based Deformable Part Based Models (DPM), learned from fully annotated examples by the Structured Output Support Vector Machine (SO-SVM) as the framework for developing the multi-view detector. Obstacles associated with this choice and our approach to mitigate them is the main subject of this thesis.

1.2. Datasets with Facial Landmarks Annotation

To get the quantitative evaluation of landmark detectors, as well as to train them, the datasets with annotated examples of facial images are needed. Since the acquisition of such datasets, and especially their annotation, is expensive, in the near past there were only a few publicly available datasets. Fortunately, with the increased interest in the facial landmark detection, the number of existing datasets has also increased. Nowadays, there exists a relatively large set of options. We list the most frequently used datasets, and provide their short description below. A summary is given in Table 1.1. Most of the datasets are also used in the experimental evaluation of the proposed landmark detectors.

LFW [Huang et al., 2007] Labeled Faces in the Wild (LFW) is a database of face photographs designed for studying the problem of unconstrained face recognition.

Figure 1.3. Exemplary inputs (top), and outputs (bottom) of desired multi-view facial landmark detector. Red dots denote estimated landmarks. The rough estimate of the yaw angle is shown on the top of each face box. The confidence of the estimate is reported in the bottom part of each face box.

The data set contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1,680 of the people pictured have two or more distinct photos in the data set. The only constraint on these faces is that they were detected by Viola and Jones [2004] face detector.

We use the LFW database enhanced by the manual annotation of 7 landmarks[1], namely the eye canthi, mouth corners and a tip of the nose. The representative images from the database, including the annotation, are depicted in Figure 1.5.

Figure 1.5. Some examples from the LFW dataset with the 7 landmarks annotation.

Multi-PIE [Gross et al., 2010] The CMU Multi-PIE face database contains more than 750,000 images of 337 people recorded in up to four sessions over the span of five months. Subjects were imaged under 15 viewpoints and 19 illumination conditions while displaying a range of facial expressions. The high-resolution frontal images

[1] Courtesy of Eyedea Recognition, Ltd.

Figure 1.4. Near-frontal facial landmark detector with a dense set of landmarks. Input is on the left, the resulting output on the right. The axis-aligned face box (which is the part of the detector's input) is shown in yellow. The red dots denote the detected landmarks. The blue lines represent the graph of landmarks connections, see Section 3.1 for details. This type of detector is suitable for more complex face analysis, such as the expression recognition, etc.

were acquired as well. In total, the database contains more than 305 GB images.

There exist annotation of 68, and 39 facial landmarks set for the near-frontal, and profile face poses, respectively. See Figure 1.6 for a sample. However, this annotation does not come with the database and is available on demand from the authors for academic use only.

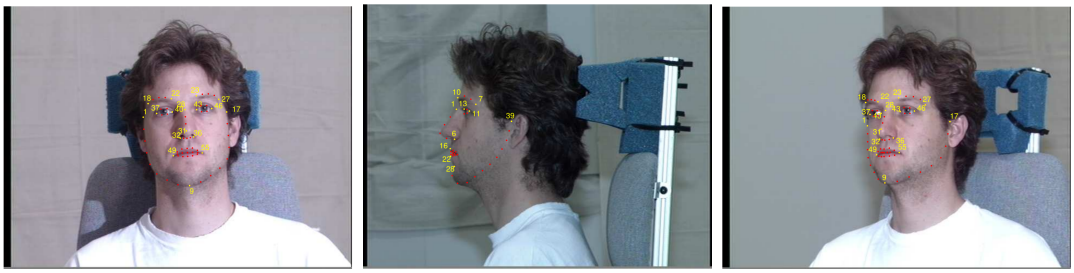


Figure 1.6. Representative Multi-PIE images, and annotations.

LFPW [Belhumeur et al., 2011] "Labeled Face Parts in the Wild" (LFPW) consists of 1,432 faces from images downloaded from the internet using simple text queries on sites such as `www.google.com`, `www.flickr.com`, and `www.yahoo.com`. Each image was labeled by three Amazon Mechanical Turk workers, and the annotation of 29 fiducial points is included in the dataset, see Figure 1.7 for the annotated landmarks configuration.

Figure 1.7. LFPW landmarks annotation configuration.

Due to the copyright issues, the images are available only by their URL addresses, and therefore not all images are still available for download.

AFW [Zhu and Ramanan, 2012] "Annotated Faces in the Wild (AFW) test sets" (AFW) is a collection of 205 Flickr images with 468 faces. Images tend to contain cluttered backgrounds with significant variations in both face viewpoint, and appearance (aging, sunglasses, makeups, skin color, expression, etc.). Each face is labelled with a bounding box, 6 landmarks (centers of eyes, tip of nose, two corners and the center of mouth) and a discretized viewpoint ($-90°$ to $90°$ every $15°$) along pitch and yaw directions (left, center, right) and viewpoints along the roll direction.

PUT [Kasinski et al., 2008] "PUT Face Database" is a collection of $9,971$ images of 100 individuals. Images were taken under controlled conditions, and the database is supplied with additional data including face bounding boxes, eyes, nose and mouth landmarks positions and manually annotated contour models. The database is available for research purposes only.

HELEN [Le et al., 2012] is a collection of $2,000$ training and 330 test images with highly accurate and consistent annotations of the key face components. A large set of candidate photos was gathered using a variety of keyword searches on Flickr. In all cases, the query included the keyword "portrait" and was augmented with different terms such as "family", "outdoor", "studio", "boy", "wedding", etc. An attempt was made to avoid cultural bias by repeating the queries in several different languages. A face detector was run on the resulting candidate set to identify a subset of images that contain faces greater than 500 pixels in width. The subset was further filtered by hand to remove the false positives, profile views, as well as the low-quality images. For each accepted face, a cropped version of the original image that includes the face and a proportional amount of background is generated. In some cases, the face is very close or in contact with the edge of the original image and is consequently not centered in the cropped image. Also, the cropped image can contain other face instances since many photos contain more than one person in proximity.

Finally, the images were hand-annotated using the Amazon Mechanical Turk to locate precisely the eyes, nose, mouth, eyebrows, and jawline. The same annotation convention as in the PUT Face Database [Kasinski et al., 2008] was adopted. To assist the Turk worker in this task, the point locations were initialized by STASM [Milborrow and Nicolls, 2014] algorithm that had been trained on the PUT database. However, since the Helen Dataset is much more diverse than PUT, the automatically initialized points were often far from the correct locations.

XM2VTS [Messer et al., 1999] "The Extended Multi-Modal Verification for Teleservices and Security Applications Database" (XM2VTSDB) contains four recordings of 295 subjects taken over a period of four months. Each record includes a speaking head shot and a rotating head shot. Sets of data taken from this database are available including high-quality color images, 32 KHz 16-bit sound files, video sequences, and a 3D Model.

AFLW [Köstinger et al., 2011] "Annotated Facial Landmarks in the Wild" (AFLW) is a large-scale, multi-view, real-world face database with annotated facial features. The images were gathered from Flickr using a broad range of face relevant tags, e.g. face, mugshot, profile face. All images were manually scanned for images containing faces. The database contains about 25,000 annotated faces from the "in-the-wild" images. 59% of these are tagged as female and 41% as male. Some images contain multiple faces, and there was no rescaling nor cropping applied to them. AFLW provides manual annotation of 21 landmarks, which were annotated upon visibility, i.e. there is no annotation if the landmark is not visible. The database is not limited to the frontal or near-frontal face poses but instead a large range of natural face poses is captured, which is one of the main features of this collection. Face rectangles and ellipses (compatible with the FDDB [Jain and Learned-Miller, 2010] protocol) are also provided along with a coarse head-pose obtained by fitting a mean 3D face with the POSIT algorithm. There is a comprehensive set of tools to work with the annotations and a database backend that enables to import other face collections and annotation types, including the graphical user interface to view and manipulate the annotations.

The database is useful for several tasks, including the facial landmark localization, multi-view face detection or head pose estimation.

We noticed that especially the landmark annotation is sometimes quite poor. To ease this difficulty, we developed an annotation tool, which is compatible with the 21 landmarks set used in AFLW, for fixing the wrong annotations and making it more consistent. We provide the set of 11,384 re-annotated faces (`http://cmp.felk.cvut.cz/~uricamic/clandmark`).

300-W [Sagonas et al., 2013b,a, 2016] "300 Faces in the Wild" (300-V) refer to 2 distinct databases. The first one is a collection of the re-annotated datasets LFPW [Belhumeur et al., 2011], AFW [Zhu and Ramanan, 2012], HELEN [Le et al., 2012], XM2VTS [Messer et al., 1999], and IBUG [Sagonas et al., 2013a] using the same landmark set as is utilized in the Multi-PIE [Gross et al., 2010] database for the near-frontal face poses. The second one is the database which was kept non-public during the both 300-W competitions [Sagonas et al., 2013a, 2016] and was released for public recently. The 300-W dataset is nowadays probably the most important benchmark in landmark detection.

300-VW [Chrysos et al., 2015; Shen et al., 2015; Tzimiropoulos, 2015] "300 Face Videos in the Wild" (300-VW) is a collection of short video sequences with duration around 1 minute (at 25–30 fps). All frames are annotated with the same set of 68 landmarks as in the 300-W database. The database is split into 3 parts based on the difficulty of the landmark localization, based on lighting conditions, occlusions and background clutter.

Table 1.1. List of existing face datasets containing images with annotation of facial landmarks.

Dataset	#images	#landmarks	Conditions	Resolution
LFW [Huang et al., 2007]	13,000	7	in-the-wild	250 × 250 px
Multi-PIE [Gross et al., 2010]	750,000	up to 21/68	lab	640 × 480 px
LFPW [Belhumeur et al., 2011]	1,432	35	in-the-wild	variable
AFW [Zhu and Ramanan, 2012]	468	6	in-the-wild	variable
PUT [Kasinski et al., 2008]	9,971	–	lab	—
HELEN [Le et al., 2012]	2,330	194	in-the-wild	variable
XM2VTS [Messer et al., 1999]	2,360	68	lab	720 × 576
AFLW [Köstinger et al., 2011]	25,000	up to 21	in-the-wild	variable
300-W [Sagonas et al., 2013b,a, 2016]	4,102+600	68	in-the-wild	variable
300-VW [Chrysos et al., 2015; Shen et al., 2015; Tzimiropoulos, 2015]	114 video clips approx. 1 minute (25–30 fps)	68	in-the-wild	variable

1.3. Contributions

In this section, we summarize the main contributions of this thesis. We group the contributions into the three categories.

1. We proposed two novel approaches for learning parameters of the tree based DPM landmark detectors.

   First, we proposed to use the SO-SVM [Tsochantaridis et al., 2005] framework to learn the appearance model and the shape model of the deformable part detector jointly [Uřičář et al., 2012]. Learning is transformed to a single convex optimization problem whose objective function is a surrogate of the actual performance metric that is used to evaluate the landmark detector. The experimental comparison shows that optimizing the actual loss function brings significant improvement in contrast to the existing methods that optimize simpler loss functions, like for example [Zhu and Ramanan, 2012].

   Second, we propose a novel two-stage learning approach which is more suitable for learning the multi-view landmark detector whose performance is evaluated by two different criteria. In the first stage, we learn a set of single-view DPM based detectors independently, by optimizing their ability to precisely localize the position of the landmarks for a corresponding viewing angle. In the second stage, we learn a multi-class classifier that operates on the features found by the independent landmark detectors of the first stage. The multi-class classifier is optimized to predict the viewing angle accurately. In comparison to the single-stage learning approach, the two-stage approach consistently improves the prediction accuracy and, mainly, it significantly reduces the overall learning time.

2. We proposed two algorithmic improvements of the Bundle Method for Regularized Risk Minimization (BMRM) [Teo et al., 2010], which is a standard solver used to optimize the convex problems emerging in the SO-SVM learning. An efficient solver with optimality guarantees turned out to be a necessary component when learning the DPM landmark detector that uses high dimensional features.

   First, we proposed to approximate the SO-SVM objective function by multiple cutting plane models in contrast to the standard BMRM which uses only a single model [Uřičář and Franc, 2012]. We experimentally demonstrate that using multi-

ple cutting plane models leads to a better approximation, which in turn significantly reduces the number of iterations of the BMRM algorithm.

Second, in [Uřičář et al., 2013] we proposed to augment the objective function of BMRM by an additional quadratic proxy term alike to the original bundle methods for the non-smooth convex optimization [Lemaréchal et al., 1995; Lemaréchal, 1978]. The added proxy term enforces the consequent solutions of the iterative process not to be excessively far from each other. This helps to mitigate the "zig-zag" behavior which is often observed when the standard quadratic regularization term present in the BMRM objective has a low influence. We propose a new strategy to automatically tune the strength of the added proxy term to decrease the overall number of iterations.

Experimental evaluation shows that the new BMRM algorithm which uses both proposed improvements speeds up learning up to an order of magnitude on a standard computer vision benchmarks, and 3 to 4 times when applied to the learning of the DPM landmark detector.

3. We developed an efficient implementation of the DPM based landmark detector with several novel speedups. The proposed speedups enable to use the globally optimal prediction strategy of the DPM detector in real-time even for dense landmark configurations.

   First, we propose to describe the landmark appearance by the Sparse Pyramid of Local Binary Patterns (S-LBP) descriptor [Uřičář and Franc, 2012]. The S-LBP is a sparse binary vector composed of standard Local Binary Patterns (LBP) [Ojala et al., 2002] that are evaluated in each pixel of the image patch and under different scales. We propose to use the MIPMAP representation which allows compiling the S-LBP descriptors in each position of the search space from pre-computed base LBP features [Uřičář et al., 2015b]. The MIPMAP representation reduces the total number of computed base LBP features leading to a significant speedup which is almost independent of the number of detected landmarks.

   Second, we propose a coarse-to-fine detection scheme to decrease the evaluation time [Uřičář et al., 2016a]. In the first stage, a coarse detector operating on low-resolution images and detecting a low number of landmarks is used. In the second stage, the landmarks detected by the coarse detector are used to restrict the search space of the fine detector operating on a higher resolution image and detecting the required number of landmarks. The coarse-to-fine strategy helps to make the combinatorial prediction problem tractable in a real-time even for a dense set of 68 landmarks [Uřičář et al., 2015a].

   Third, an efficient implementation of the proposed multi-view landmark detector, as well as the learning algorithms, were released in an open-source library **CLandmark**[2]. The library has been widely used by the community in numerous projects. See Section A.1 for a list of scientifical papers citing CLandmark related papers [Uřičář et al., 2015b,a, 2016a].

1.4. Outline

Chapter 2 presents the state-of-the-art related to the topics studied in the thesis. We put the emphasis on two core categories. Namely, the generative and the discriminative methods and their typical representatives.

[2] `https://cmp.felk.cvut.cz/~uricamic/clandmark`

Chapter 3 describes the proposed DPM based landmark detector including the novel strategies to speed up its prediction stage (contribution 3).

Chapter 4 formulates the problem of learning parameters of the DPM detector from examples as an instance of the SO-SVM framework. In this chapter, we describe the two proposed formulations of the learning problem (contribution 1), as well as, the proposed improvements of the BMRM solver (contribution 2).

Chapter 5 presents the experimental evaluation of the proposed detector and the learning algorithms. The proposed detector is compared to a large number of existing methods in a variety of different settings and benchmark data.

Chapter 6 provides the conclusions of the thesis and describes possible directions of the future research.

2. Related Work

In this chapter, we summarize the state-of-the-art methods for facial landmark detection. We split the relevant work into two main categories: the generative and the discriminative methods. In the following text, we describe typical examples of each category. The aim is to outline main principles rather than to given an exhaustive description.

2.1. Generative Methods

Generative methods build a parametric model of face shape and its appearance. It is most common to use linear models for both. Linear model means that the shape/appearance is given as a linear combination of a set of template shapes/appearances. The template shapes/appearances are learned from examples of facial images with annotated position of landmarks. A generative model allows generating synthetic faces. Fitting the generative model to the image amounts to searching for a synthetic face most similar to the input face. Sum of squared differences of pixel intensities is often chosen as the similarity measure which makes the model fitting stage an instance of the non-linear least squares problem. The desired landmark positions are subsequently extracted from shape parameters of the most similar synthetic image. Although the shape and the appearance are described by linear models, the error function, which is minimized during the model fitting, is highly non-linear in the shape/appearance parameters. Specialized iterative solvers usually solve the resulting nonlinear minimization problem without the guarantee to find the global minimum. Among the most popular generative methods applied to facial landmark detection belong the Active Appearance Models (AAM) [Cootes et al., 2001; Matthews and Baker, 2004], and Morphable models [Blanz and Vetter, 2003].

The AAMs have been extensively studied in the field of computer vision since their introduction by Cootes et al. [2001]. They are used for an alignment of various types of objects spanning from the medical imaging to human faces. The main advantages (✓) and disadvantages (✗) of the AAMs are as follows:

✓ They can generate synthetic faces.

✓ They provide interpretable parameterization of the shape and appearance which is useful for various purposes, e.g. for face recognition.

✓ The model parameters, which involve templates of the shape and appearance, can be learned by simple algorithms like Principal Component Analysis (PCA).

✗ Fitting the model to the image is a highly non-convex problem. The existing solvers return only a local optimum quality of which strongly relies on an initial estimate.

✗ The learning algorithm does not directly optimize the true objective of the landmark detection, i.e. the localization accuracy.

In the remainder of this section, we outline the basic variant of the AAMs and algorithms used for fitting the model to the image. Our exposition is largely based on the seminal paper of [Matthews and Baker, 2004] from which we adopt the notation.

2.1.1. Building blocks of Active Appearance Models

AAMs are defined by three integral building blocks: shape model, appearance model, and deformation model. A short description of each of them follows.

Shape model Shape is represented by a vector $\boldsymbol{s} = (x_1, y_1, \ldots, x_L, y_L)^\top \in \mathbb{R}^{2L}$ which is composed of (x, y) coordinates of L landmark points connected to a triangulated mesh. Let $\mathcal{T} = \{(I^1, \boldsymbol{s}^1), \ldots, (I^m, \boldsymbol{s}^m)\}$ be a training set of facial images I^j and corresponding shapes $\boldsymbol{s}^j$. The training set $\mathcal{T}$ is used to construct shape model as follows. The Procrustes Analysis [Goodall, 1991] is applied to remove the similarity transformations from the original shapes. The affine subspace spanned by the resulting m similarity-free shapes is approximated by the PCA. In return, we get a linear shape model consisting of the mean shape $\boldsymbol{s}_0$ and n template shapes $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$ which correspond to the eigenvectors of the covariance matrix calculated from the similarity-free shapes. A shape $\boldsymbol{s}$ generated by the model is then represented as a linear combination

shape model

$$\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{n} \boldsymbol{s}_i p_i \,, \tag{2.1}$$

where $\boldsymbol{p} = (p_1, \ldots, p_n)^\top \in \mathbb{R}^n$ is a vector of shape parameters. The idea is illustrated in Figure 2.1. Because the eigenvectors of a symmetric real matrix are orthogonal, shape templates are orthogonal as well. Provided the template shapes are not obtained by PCA, they can be orthogonalized by a linear reparametrization. The orthogonality is important because it is exploited by algorithms fitting the model to the image. Since the model is constructed on similarity-free shapes, it does not capture the global transformations like scaling, translation, and rotation. This can be resolved by appending auxiliary shape templates and re-orthogonalization as proposed by Matthews and Baker [2004].

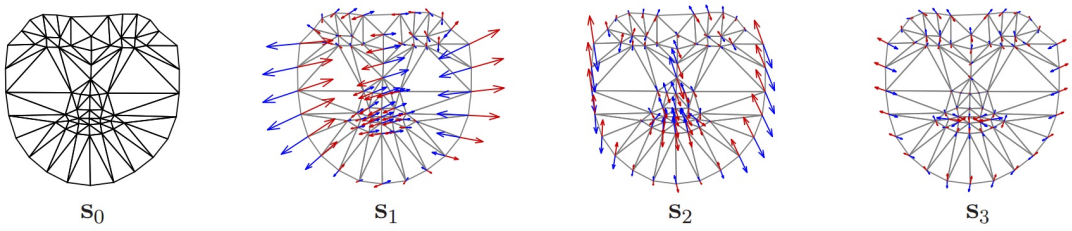


Figure 2.1. The linear shape model used by the AAMs. The model consists of the mean shape $\boldsymbol{s}_0$ and a sequence of the eigen-shapes $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n)$ capturing the variation of the shape. The mesh obtained by triangulating the mean shape $\boldsymbol{s}_0$ defines the piece-wise affine warp. The picture is adopted from [Matthews and Baker, 2004].

Deformation model The deformation model is defined by a function $\boldsymbol{x}' = \mathbf{W}(\boldsymbol{x};\ \boldsymbol{p})$ warping the pixel coordinates $\boldsymbol{x} = (x, y)$ in the template coordinate system to coordinates $\boldsymbol{x}' = (x, y)$ in the image coordinate system, or in the opposite direction using the inverse warp $\boldsymbol{x} = \mathbf{W}^{-1}(\boldsymbol{x}';\ \boldsymbol{p})$. The vector $\boldsymbol{p}$ encapsulates the shape parameters as before. The warping function is usually represented by a piece-wise affine warp which is illustrated in Figure 2.2. A point $\boldsymbol{x} = (x, y)$ from the template coordinate system is transformed as follows. Triangle T containing $\boldsymbol{x}$ is found in the triangulated mesh defined by the mean shape $\boldsymbol{s}_0$. Subsequently, triangle T is transformed to a triangle T' in the image coordinate system whose coordinates are given by (2.1). Finally, the input coordinates $\boldsymbol{x}$ are warped by the affine transform mapping triangle T to triangle T'. Beside the piece-wise affine warp other transformations can be used. For example,

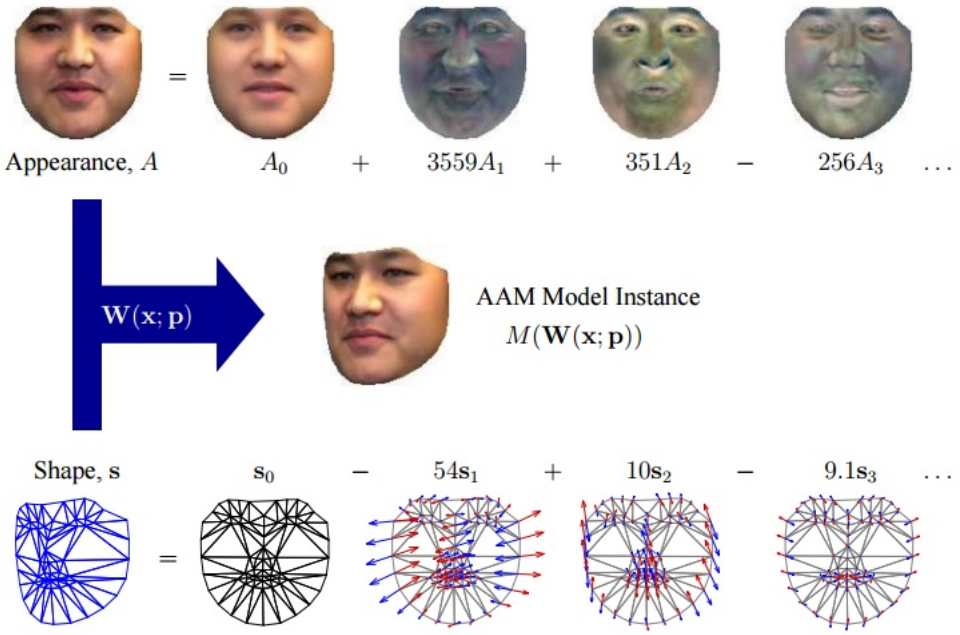


Figure 2.2. Picture illustrates the process of generating a synthetic faces by the AAM. The texture A is generated by (2.2) using appearance parameters $\boldsymbol{\lambda} = (3559, 351, -256, \ldots)$. The shape $\boldsymbol{s}$ is generated by (2.1) using the shape parameters $\boldsymbol{p} = (-54, 10, -9.1, \ldots)$. The synthetic faces $M(\mathbf{W}(\boldsymbol{x};\boldsymbol{p})) = A(\boldsymbol{x})$ is obtained by transforming $A(\boldsymbol{x})$ to image coordinate system via piece-wise affine warp $\mathbf{W}(\boldsymbol{x};\boldsymbol{p})$. The picture is adopted from [Baker and Matthews, 2004].

a simple translation is employed in the part-based AAMs [Tzimiropoulos and Pantic, 2014].

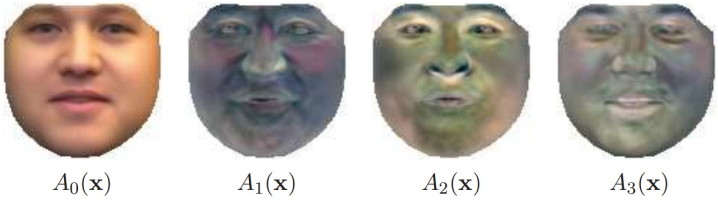


Figure 2.3. The linear model of the face appearance used in the AAMs. The model consists of a mean texture A_0 defined on pixels inside the mean mesh $\boldsymbol{s}_0$ and a set k of eigenfaces $(A_1, \ldots, A_k)$ defined on the same set of pixels. The picture is adopted from [Matthews and Baker, 2004].

Appearance model The appearance model is usually defined within the mesh obtained by triangulation of the mean shape $\boldsymbol{s}_0$. Before building the appearance model the shape variation is removed from the training images in $\mathcal{T}$. This is done by transforming the training images $\{I^1, \ldots, I^m\}$ to the mean shape $\boldsymbol{s}_0$ using the deformation model $\mathbf{W}(\boldsymbol{x};\boldsymbol{p})$. Then, similarly to the shape model, the PCA is applied to the geometrically normalized training images. The result is a mean appearance A_0 and k eigenfaces $\{A_1, \ldots, A_k\}$ as illustrated in Figure 2.3. The admissible shape-free texture $A(\boldsymbol{x})$ is then represented as a linear combination

$$A(\boldsymbol{x}) = A_0(\boldsymbol{x}) + \sum_{i=1}^{k} \lambda_i A_i(\boldsymbol{x}) \,, \qquad \boldsymbol{x} \in \mathcal{S}_0 \,, \tag{2.2}$$

appearance model

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)^\top \in \mathbb{R}^k$ are the appearance parameters and $\mathcal{S}_0$ are pixel coordinates inside the mesh defined by $\boldsymbol{s}_0$. Likewise the shape parameters, the appearance parameters are orthogonal due to using the PCA. Otherwise, the orthogonality can be guaranteed by linear re-parametrization.

In the above formulation of the shape and appearance model have their set of parameters, that is, $\boldsymbol{p}$ and $\boldsymbol{\lambda}$, respectively. This variant is called *independent* AAM [Matthews and Baker, 2004] . . Another option is to merge the parameters, which is usually done by performing an additional PCA on top of the shape and appearance templates resulting in a single set of parameters. The variant is called the *combined* AAM. The combined AAM is more general, in fact, the independent AAM is a special case, and it often needs fewer parameters to represent the same degree of accuracy as the independent AAM. In turn fitting the model is more efficient and accurate [Matthews and Baker, 2004]. On the other hand coupling of the parameters prevents the joint appearance-shape templates to be orthogonal which restricts the choice of the fitting algorithm.

independent AAM

Combined AAM

2.1.2. Fitting AAMs to image

Fitting an AAM, defined by shape model (2.1) and appearance model (2.2), to an input image I is defined as a non-linear least squares problem

fitting task

$$\min_{\boldsymbol{p},\ \boldsymbol{\lambda}} \sum_{\boldsymbol{x} \in \mathcal{S}_0} \left[A_0(\boldsymbol{x}) + \sum_{i=1}^{k} \lambda_i A_i(\boldsymbol{x}) - I(\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p})) \right]^2 . \tag{2.3}$$

In words, we want to find shape parameters $\boldsymbol{p}$ and appearance parameters $\boldsymbol{\lambda}$ such that the model generated texture $A(\boldsymbol{x}) = A_0(\boldsymbol{x}) + \sum_{i=1}^{d} \lambda_i A_i(\boldsymbol{x})$ is most similar to the input image $I(\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p}))$ mapped by the warp $\mathbf{W}(\boldsymbol{x}, \boldsymbol{p})$ to the coordinate system of the texture $A(\boldsymbol{x})$. The similarity is measure by the sum of squared differences of the pixel intensities $A(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{S}_0$, and $I(\mathbf{W}(\boldsymbol{x}; \boldsymbol{p}))$, $\boldsymbol{x} \in \mathcal{S}_0$, respectively.

The standard gradient-based methods are usually slow when applied directly to the problem (2.3). For this reason, various specialized solvers have been proposed. In the remainder of this section, we briefly review some of them. In our exposition we assume that the texture is given only by the mean template $A_0(\boldsymbol{x})$, that is $k = 0$, which leads to

$$\min_{\boldsymbol{p}} \sum_{\boldsymbol{x} \in \mathcal{S}_0} \left[A_0(\boldsymbol{x}) - I(\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p})) \right]^2 . \tag{2.4}$$

The problem (2.4) involves optimization of the shape parameters only, which constitutes the main challenge. It has been shown (see [Matthews and Baker, 2004]) that the original fitting problem (2.3) can be decomposed into two independent sub-problems. First, finding the shape parameters $\boldsymbol{p}$ by solving a slight modification of the problem (2.4). Second, optimizing the texture parameters $\boldsymbol{\lambda}$, which has a closed form solution. In other words, solvers of the problem (2.4), that we are going to be explain next, can be utilized to solve the original problem (2.3).

Lucas-Kanade algorithm

The goal of the Lucas-Kanade (LK) algorithm is to align the template image $A_0(\boldsymbol{x})$ to an input image $I(\boldsymbol{x}')$. The LK algorithm was originally proposed for tracking when the template $A_0(\boldsymbol{x})$ might be represented by a region in the video frame at time t_i and $I(\boldsymbol{x}')$ the video frame at time t_{i+1}. The warp $\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p})$ takes the points $\boldsymbol{x}$ from the coordinate system of the template $A_0(\boldsymbol{x})$ and maps it to the sub-pixel location

$\boldsymbol{x}' = \mathbf{W}(\boldsymbol{x};\ \boldsymbol{p})$ in the coordinate system of the image. Warp $\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p})$ might be, for example, a simple translation:

$$\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} . \tag{2.5}$$

translation warp

in which case the vector of parameters $\boldsymbol{p}$ represents the optical flow. Another example, when tracking a patch moving in 3D, a better choice would be a set of affine warps:

$$\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p}) = \begin{bmatrix} (1+p_1)x & + & p_3 y & + & p_5 \\ p_2 x & + & (1+p_4)y & + & p_6 \end{bmatrix} = \begin{bmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} . \tag{2.6}$$

affine warp

There are other ways how to parametrize the affine warps. In general, however, the number of parameters n might be arbitrarily large and $\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p})$ arbitrarily complex. As already mentioned, the standard AAMs use the piece-wise affine warps.

Given an initial estimate $\boldsymbol{p}$, the LK algorithm solves to the optimization problem (2.4) iteratively by optimizing the increments $\Delta\boldsymbol{p}$:

$$\min_{\Delta\boldsymbol{p}} \sum_{\boldsymbol{x}\in\mathcal{S}_0} \left[A_0(\boldsymbol{x}) - I(\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p}+\Delta\boldsymbol{p}))\right]^2 \tag{2.7}$$

LK fitting task

and consequently updating the parameters in the additive fashion

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta\boldsymbol{p} . \tag{2.8}$$

additive update

The LK algorithm iterates this process until convergence which is typically based on monitoring the norm of the increment and stopping when $\|\Delta\boldsymbol{p}\| \leq \epsilon$ for a prescribed ϵ.

The problem (2.7) is solved by the Gauss-Newton method. This involves replacing $I(\mathbf{W}(\boldsymbol{x};\ \boldsymbol{p}+\Delta\boldsymbol{p}))$ by its first order Taylor expansion around the point $\boldsymbol{p}$ which yileds a linear least squares problem:

$$\min_{\Delta\boldsymbol{p}} \sum_{\boldsymbol{s}\in\mathcal{S}_0} \left[I(\mathbf{W}(\boldsymbol{s};\ \boldsymbol{p})) + \nabla I \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}} \Delta\boldsymbol{p} - A_0(\boldsymbol{s})\right] , \tag{2.9}$$

Taylor expansion

where $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ is the gradient of the image function evaluated at $\mathbf{W}(\boldsymbol{s};\ \boldsymbol{p})$, and $\frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}\boldsymbol{p}}$ is the Jacobian of the deformation model evaluated at $\boldsymbol{p}$. The problem (2.9) has a closed-form solution:

$$\Delta\boldsymbol{p} = \mathbf{H}^{-1} \sum_{\boldsymbol{s}\in\mathcal{S}_0} \left[\nabla I \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}}\right]^\top \left[A_0(\boldsymbol{x}) - I(\mathbf{W}(\boldsymbol{s};\ \boldsymbol{p}))\right] , \tag{2.10}$$

closed form

where

$$\mathbf{H} = \sum_{\boldsymbol{s}\in\mathcal{S}_0} \left[\nabla I \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}}\right]^\top \left[\nabla I \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}}\right] \tag{2.11}$$

LK Hessian

is the Hessian matrix.

The LK algorithm is in the AAM community referred to as *forward-additive*, because it operates in the forward coordinate frame, i.e. the image is warped to the template, and the parameters are updated in an additive fashion. The main disadvantage is the need to recompute the Hessian matrix in every iteration. Note that the Hessian $\mathbf{H}$ is defined by the term $\nabla I \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}}$ depending on the image. Recomputation of the Hessian makes the LK algorithm slow unless the number of template pixels $|\mathcal{S}_0|$ is small. In order to overcome the problem, Baker and Matthews [2004] proposed the inverse compositional algorithm, which we describe next.

Inverse Compositional Algorithm

The main idea of the Inverse Compositional Algorithm (ICA) is to switch the roles of image and template leading to the following optimization task

ICA fitting task

$$\min_{\Delta \boldsymbol{p}} \sum_{\boldsymbol{s} \in \mathcal{S}_0} \left[A_0(\mathbf{W}(\boldsymbol{s};\ \Delta \boldsymbol{p})) - I(\mathbf{W}(\boldsymbol{s};\ \boldsymbol{p})) \right]^2 . \tag{2.12}$$

Having the increment $\Delta \boldsymbol{p}$, the warp is updated by

inverse compositional update

$$\mathbf{W}(\boldsymbol{s};\ \boldsymbol{p}) \leftarrow \mathbf{W}(\boldsymbol{s};\ \boldsymbol{p}) \circ \mathbf{W}(\boldsymbol{s};\ \Delta \boldsymbol{p})^{-1} . \tag{2.13}$$

The inverse compositional approach has been proved to be equivalent to the additive one [Matthews and Baker, 2004]. Note, that in contrast to the LK algorithm (c.f. problem (2.7)), the alignment problem (2.12) fixes the image warp $I(\mathbf{W}(\boldsymbol{s};\ \boldsymbol{p}))$ and changes only the template warp $A_0(\mathbf{W}(\boldsymbol{s};\ \Delta \boldsymbol{p}))$. The problem (2.12) is again solved by the Gauss-Newton method as follows. Replacing $A_0(\mathbf{W}(\boldsymbol{s};\ \Delta \boldsymbol{p}))$ by its first order Taylor expansion in (2.12) yields the following linear least squares problem:

Taylor expansion

$$\min_{\Delta \boldsymbol{p}} \sum_{\boldsymbol{s} \in \mathcal{S}_0} \left[A_0(\mathbf{W}(\boldsymbol{s};\ \mathbf{0})) + \nabla A_0 \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}} \Delta \boldsymbol{p} - I(\mathbf{W}(\boldsymbol{s};\ \boldsymbol{p})) \right]^2 , \tag{2.14}$$

which has a closed form solution:

ICA closed form

$$\Delta \boldsymbol{p} = \mathbf{H}^{-1} \sum_{\boldsymbol{s} \in \mathcal{S}_0} \left[\nabla A_0 \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}} \right]^\top \left[I(\mathbf{W}(\boldsymbol{s};\ \boldsymbol{p})) - A_0(\boldsymbol{s}) \right] \tag{2.15}$$

with the Hessian computed as

ICA Hessian

$$\mathbf{H} = \sum_{\boldsymbol{s} \in \mathcal{S}_0} \left[\nabla A_0 \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}} \right]^\top \left[\nabla A_0 \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}} \right] . \tag{2.16}$$

Note, that image I was replaced by template A_0, compared to the original Hessian definition (2.11). The gradient $\nabla A_0 = (\frac{\partial A_0}{\partial x}, \frac{\partial A_0}{\partial y})$ as well as the Jacobian $\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}\boldsymbol{p}}$ at $\boldsymbol{p} = \mathbf{0}$ are constant. In turn most terms in the formula (2.15) can be precomputed leading to a significant speedup.

2.1.3. Variants

The **Project-Out Inverse Compositional Algorithm** (POIC) [Matthews and Baker, 2004] is an instance of the AAM which decouples fitting of the appearance and the shape model. The shape model is fitted by the algorithm outlined above. The POIC algorithm has a little per-iteration complexity, but it shows a slow convergence especially when the training and testing images are substantially different. The POIC is thus most suitable for the *person-specific* AAM fitting [Gross et al., 2005].

person-specific AAM

A different approach, where the shape and appearance parameters are optimized simultaneously, was proposed by Gross et al. [2005]. Their **Simultaneous Inverse Compositional** (SIC) algorithm is more robust for the *generic* AAM fitting. However, the increased computational cost makes the algorithm prohibitive for many applications. The big computational cost of SIC was addressed by Tzimiropoulos and Pantic [2013] who proposed an efficient algorithm called **Fast-SIC**..

generic AAM

SIC

So far we have described only AAM with the holistic appearance model. Tzimiropoulos and Pantic [2014] proposed a part based model called **Gauss-Newton Deformable Part Model** (GN-DPM). In contrast to the holistic AAM, the part-based AAM describes each landmark by a rectangular part. This allows simplifying

holistic part-based AAM

the complex warping functions to much simpler translation warp (2.5). The authors have shown that the part-based generative models may have the same representational power as AAM but the problem of fitting is much easier. In our experiments, presented in Chapter 5, we use an implementation of the GN-DPM whose performance is on par with other state-of-the-art techniques.

2.2. Discriminative Methods

The discriminative methods learn predictors directly estimating pose, shape or the landmark positions from features computed on the input image. The advantages of the discriminative methods are their conceptual simplicity and a low test time. Another advantage might be seen in the possibility to optimize the true performance measure during the training stage.

Among the most popular discriminative approaches nowadays belong cascades of regressors, which were considered for example in [Kazemi and Sullivan, 2014; Dollár et al., 2010; Ren et al., 2014]. Starting from an initial estimate, each regressor in the cascade refines prediction of the previous one. The prediction in each stage is typically based on simple features extracted from patches located at positions determined by the prediction of the previous stage. Besides 2D landmark positions, Asthana et al. [2014] show that the cascade of regressors can also accurately estimate the pose and shape of a 3D face model. Saragih and Göcke [2007, 2009] proposed to use the regression to estimate parameters of the AAM. Regression methods combined with a probabilistic graphical models were proposed in [Valstar et al., 2010; Martínez et al., 2013]. In their work, the graphical model is used to aggregate the estimates of stochastically sampled local regressors into a single robust prediction.

Another important class of discriminative methods is the DPM, also referred to as Pictorial Structures (PS), which were introduced by Fischler and Elschlager [1973] and popularized later by the work of Felzenszwalb and Huttenlocher [2005]; Felzenszwalb et al. [2010]. The DPM predict landmarks by maximizing a sum of responses of local detectors plus a score evaluating the appropriateness of shape. A common approach to model shape is to describe landmark configurations by a sum of functions defined on selected pairs of landmark positions. The pairs are chosen such that they form a tree graph. The main advantage of using the tree based shape model is that the optimization of score function can be decomposed into simpler problems and, in turn, solved efficiently by dynamic programming. Also, learning of a tree based shape models usually requires fewer shape variations in the training examples compared to the generative methods. A similar approach offer Constraint Local Models (CLM) introduced by Cristinacce and Cootes [2006, 2008]. In a sense, the CLM can be seen as a combination of the discriminative and generative methods. The CLM use a generative linear shape model (c.f. Section 2.1.1) like AAM. In contrast to AAM, instead of using a holistic appearance model, CLM employs a set of local detectors as tree based DPM. The more complex shape model has a higher chance to avoid unreasonable landmark configuration which is however paid off by making the application of global optimization methods for prediction unfeasible.

In the remainder of this section, we describe the main discriminative approaches in more details.

2.2.1. Cascaded Regression

Regression methods gained much attention recently. Cascaded regression using the pose-indexed features and random fern regressors were introduced in [Dollár et al.,

2010]. A similar approach was proposed by [Cao et al., 2012, 2014], where the authors learn a regression function to infer the whole facial shape from the image and explicitly minimize the alignment error over training data. Let $\mathcal{I}$ denote a set of all input images, and $\mathcal{S}$ denote the set of admissible shapes $\boldsymbol{s} = (x_1, y_1, \ldots, x_L, y_L)^\top \in \mathcal{S} \subseteq \mathbb{R}^{2L}$, then the regressor is a function defined as $R\colon \mathcal{I} \times \mathcal{S} \to \mathcal{S} \in \mathcal{S}$. The cascaded regression is used to combine T weak regressors $(R_1, \ldots, R_t, \ldots, R_T)$ in an additive manner. Given an input image I and an initial face shape $\boldsymbol{s}(1)$, each regressor computes a shape increment from image features and updates the face shape in a cascaded (recursive) manner as follows

$$\boldsymbol{s}(t+1) = \boldsymbol{s}(t) + R_t(I, \boldsymbol{s}(t)), \quad t = 1, \ldots, T\,. \tag{2.17}$$

That is, new shape $\boldsymbol{s}(t+1)$ depends on previous estimate $\boldsymbol{s}(t)$ and increment $R_t(I, \boldsymbol{s}(t))$ which is often a linear regressor from features computed on image I at positions $\boldsymbol{s}(t)$. The initial shape $\boldsymbol{s}(1)$ is typically a mean shape roughly aligned by face detector. The cascaded regressor then uses the formula (2.17) to generate a sequence of gradually improving shape estimates $\boldsymbol{s}(1), \ldots, \boldsymbol{s}(T+1)$.

The regressors $\{R_1, \ldots, R_T\}$ are learned from annotated training examples $\mathcal{T} = \{(I^1, \boldsymbol{s}^1), \ldots, (I^m, \boldsymbol{s}^m)\}$ by explicit minimization of the sum of prediction errors:

$$R_t = \arg\min_R \sum_{i=1}^{m} \left\| \boldsymbol{s}^i(t) + R(I^i, \boldsymbol{s}^i(t)) - \boldsymbol{s}^i \right\|_2^2, \quad t = 1, \ldots, T\,. \tag{2.18}$$

If the class of linear regressors is used, being often the case, the learning problem (2.18) has a closed form solution that can be computed very efficiently.

The main advantages and disadvantages of the regression methods are:

✓ The prediction step is simple and very fast.

✓ The learning algorithm is simple and fast.

✗ There is no prior on the shape which means that a significant variation of face shapes must be collected to the training set.

✗ Existing learning algorithms are greedy and hence sub-optimal, minimizers of the prediction error.

Linear regression

A common choice is to use a linear regression function R_t. For example, the cascade of linear regressions was proposed in Ren et al. [2014] who present a method operating at impressive $3,000$ frames per second. Their method is similar to works like e.g. [Cao et al., 2012, 2014; Dollár et al., 2010]. However, the key difference lies in the usage of sparse local binary features that can be evaluated very quickly. Different features are learned for each landmark by using an ensemble of regression trees. Learning of regression trees serves as a feature selection method providing very sparse and computationally efficient descriptors. The learned features are subsequently used to learn a cascade of global linear regressors $R_t(I, \boldsymbol{s}) = \mathbf{W}_t \boldsymbol{\Psi}_t(I, \boldsymbol{s})$, where $\boldsymbol{\Psi}_t(I, \boldsymbol{s}) \in \mathbb{R}^d$ denotes concatenation of all binary features computed around landmark positions $\boldsymbol{s}$ and $\mathbf{W} \in \mathbb{R}^{2L \times d}$ parameter matrix. Given an initial estimate $\boldsymbol{s}_{(1)}$, face shape is obtained by recursively applying the linear regression:

$$\boldsymbol{s}(t+1) = \boldsymbol{s}(t) + \mathbf{W}_t \boldsymbol{\Psi}_t(I, \boldsymbol{s}(t)), \quad t = 1, \ldots, T\,. \tag{2.19}$$

The project matrices $\mathbf{W}_1, \ldots, \mathbf{W}_T$ are learned by solving T regularized linear least squares problems

$$\mathbf{W}_t = \underset{\mathbf{W} \in \mathbb{R}^{2L \times d}}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| \boldsymbol{s}^i(t) + \mathbf{W}\boldsymbol{\Psi}_t(I^i, \boldsymbol{s}^i(t)) - \boldsymbol{s}^i \right\|_2^2 + \lambda \|\mathbf{W}\|_F^2, \quad t = 1, \ldots, T \ . \tag{2.20}$$

The first term in (2.20) represents the training prediction error and the second term, the Frobenius norm of the parameters, is a regularization term introduced to prevent over-fitting. The strength of regularization is controlled by a hyper-parameter λ on a validation set.

Supervised Descent Method

Xiong and la Torre [2013] provided a theoretical link between cascaded regression and Newton method. The IntraFace package, implementing Supervised Descent Method (SDM), is one of the best landmark detectors nowadays as can also be seen from our experiments presented in Chapter 5.

SDM is a generic method for solving non-linear least squares minimization problems. It is applied to the problem of landmark detection as follows. The prediction of landmark positions $\boldsymbol{s}$ is defined as the minimization problem:

$$f(\boldsymbol{s}) = \|\boldsymbol{\Psi}(I, \boldsymbol{s}) - \boldsymbol{\Psi}(I, \boldsymbol{s}^*)\|_2^2 \tag{2.21}$$

where $\boldsymbol{\Psi}(I, \boldsymbol{s}) \in \mathbb{R}^n$ denotes a concatenation of features extracted from the input image I around landmark positions $\boldsymbol{s}$ and $\boldsymbol{s}^*$ is the ground-truth position of landmarks. For example, Xiong and la Torre [2013] construct $\boldsymbol{\Psi}(I, \boldsymbol{s}) \in \mathbb{R}^{128L}$ as a concatenation of Scale Invariant Feature Transform (SIFT) features computed on patches cropped from I around positions $\boldsymbol{s}$. According to (2.21), SDM tries to minimize L_2 distance between features computed at the ground truth position $\boldsymbol{s}^*$ and features computed around the estimated position $\boldsymbol{s}$.

Let us consider we want to solve the minimization problem (2.21) iteratively by Newton method. Newton method approximates the objective $f(\boldsymbol{s})$ around a current estimate $\boldsymbol{s}(t)$ by the second-order Taylor expansion

$$\hat{f}(\boldsymbol{s}) = f(\boldsymbol{s}(t)) + \mathbf{J}_f(\boldsymbol{s}(t))^\top (\boldsymbol{s} - \boldsymbol{s}(t)) + \frac{1}{2}(\boldsymbol{s} - \boldsymbol{s}(t))^\top \mathbf{H}(\boldsymbol{s}(t))(\boldsymbol{s} - \boldsymbol{s}(t)) \ , \tag{2.22}$$

where $\mathbf{J}_f(\boldsymbol{s}(t))$ and $\mathbf{H}(\boldsymbol{s}(t))$ are Jacobian and Hessian matrices of f evaluated at $\boldsymbol{s}(t)$, respectively. New iterate $\boldsymbol{s}(t+1)$ is computed as the minimum of $\hat{f}(\boldsymbol{s})$ which has a closed form solution

$$\begin{aligned} \boldsymbol{s}(t+1) &= \boldsymbol{s}(t) - \mathbf{H}(\boldsymbol{s}(t))^{-1} \mathbf{J}_f(\boldsymbol{s}(t)) \\ &= \boldsymbol{s}(t) - 2\mathbf{H}(\boldsymbol{s}(t))^{-1} \mathbf{J}_\Psi(\boldsymbol{s}(t))^T (\boldsymbol{\Psi}(I, \boldsymbol{s}(t)) - \boldsymbol{\Psi}(I, \boldsymbol{s}^*)) \end{aligned} \tag{2.23}$$

where $\boldsymbol{J}_\Psi(\boldsymbol{s}(t))$ is Jacobian of $\boldsymbol{\Psi}(I, \boldsymbol{s})$ evaluated at $\boldsymbol{s}(t)$. By introducing shortcuts $\mathbf{W}_t = -2\mathbf{H}(\boldsymbol{s}(t))^{-1}\mathbf{J}_\Psi(\boldsymbol{s}(t))^T$ and $\boldsymbol{b}_t = \mathbf{W}_t \boldsymbol{\Psi}(I, \boldsymbol{s}^*)$ we can simplify the Newton update (2.23) to

$$\boldsymbol{s}(t+1) = \boldsymbol{s}(t) + \mathbf{W}_t \boldsymbol{\Psi}(I, \boldsymbol{s}(t)) + \boldsymbol{b}_t \ . \tag{2.24}$$

The standard Newton method would involve computation of projection matrices $\mathbf{W}_t$ and bias terms $\boldsymbol{b}_t$ at each iteration. This can be demanding as it involves evaluation of Hessian of $f(\boldsymbol{s})$ and Jacobian of $\boldsymbol{\Psi}(I, \boldsymbol{s})$. Moreover, term $\boldsymbol{b}_t$ is defined by the ground-truth landmark position $\boldsymbol{s}^*$, which is unknown at the test time. SDM instead learns projection parameters $(\mathbf{W}_t, \boldsymbol{b}_t)$, $t = 1, \ldots, T$, from training examples by minimizing L_2 prediction error of each iterate. SDM is thus equivalent to the linear cascaded regression described in Section 2.2.1.

2.2.2. Tree-based Deformable Part Models

Tree-based DPM [Fischler and Elschlager, 1973; Felzenszwalb and Huttenlocher, 2005; Felzenszwalb et al., 2010] predict landmark position by maximizing a sum of responses of local detectors plus a score evaluating the shape. The shape is modeled by a graph $G = (V, E)$ whose vertices V represent landmarks and edges E pairs of landmarks with mutually dependent positions. Let $\boldsymbol{s} = (\boldsymbol{s}_1, \dots, \boldsymbol{s}_L) \in \mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_L$ be a landmark configuration (or face shape) which is defined by coordinates $\boldsymbol{s}_i = (x_i, y_i)$, $i \in \mathcal{V}$, of L landmarks [1]. The sets of possible positions of individual landmarks $\mathcal{S}_i$, $i \in \mathcal{V}$ are finite in contrast to methods discussed so far. Prediction of the best landmark configuration is formulated as a single energy maximization problem

$$\boldsymbol{s}^* \in \arg\max_{\boldsymbol{s} \in \mathcal{S}} \left(\underbrace{\sum_{i \in V} q_i(\boldsymbol{s}_i, I)}_{\text{appearance model}} + \underbrace{\sum_{(i,j) \in E} g_{ij}(\boldsymbol{s}_i, \boldsymbol{s}_j)}_{\text{deformation cost}} \right). \tag{2.25}$$

Note that unlike previous methods, in the case of DPMs the prediction leads to a discrete optimization problem because the set of feasible solutions $\mathcal{S}$ is finite. The objective function of (2.25) consists of two parts. The first one, an appearance model, is defined on the set of vertices V of the underlying graph and the second one, a deformation cost, is defined on its edges E. The appearance model evaluates a match between the image and distinctive landmarks. The value of $q_i(\boldsymbol{s}_i, I)$ corresponds to a likelihood that the i-th landmark is at position $\boldsymbol{s}_i$ in image I. The deformation cost $g_{ij}(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is a function scoring a relative positions $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ of the connected landmarks $(i, j) \in E$. Using the pair-wise costs makes the value of objective function invariant to global similarity transformations. The deformation model can be thought of as a set of springs connecting pairs of dependent landmarks. Felzenszwalb and Huttenlocher [2005] propose to limit the underlying graph (V, E) to a tree which allows solving the discrete optimization problem (2.25) by Dynamic Programming (DP). A disadvantage of the tree structure is that the shape model is relatively weak which may result in an anthropologically implausible landmark configurations. On the other hand, weak shape model requires less shape variation in the training examples and, most importantly, it allows to find the globally optimal configuration efficiently. The global optimization makes DPM independent on an initial estimate in contrast to all other methods discussed so far. Moreover, a mixture of DPM, which will be described later, allows modeling a large range of the viewing angles in a principled way. The model parameters are typically trained from a set of training annotated examples by discriminative approaches [Felzenszwalb and Zabih, 2010]. However, generative methods can be used as well [Everingham et al., 2006].

The main advantages and disadvantages of existing tree-based DPMs are:

✓ The prediction problem can be solved globally in a time which is polynomial in the size of the image and the number of landmarks.

✗ The shape model is too simplistic which may result in anthropologically implausible configurations of the landmarks.

[1] So far we have used the subscript to index vectors containing concatenation of all landmark coordinates. For example, we used $\{\boldsymbol{s}_1, \dots, \boldsymbol{s}_n\}$ to denote n shape templates in AAMs. Because the DPM optimize each landmark position independently, it is advantageous to use the subscript to denote coordinates of individual landmarks $\boldsymbol{s}_i = (x_i, y_i)$, $i \in \mathcal{V}$ and vector $\boldsymbol{s} = (\boldsymbol{s}_1, \dots, \boldsymbol{s}_L)$ (without the subscript), to denote their concatenation. This notation is going to be used consistently from now on untill the end of the thesis.

✗ Solving the prediction problem for a large number of landmarks or high-resolution images is computationally demanding.

✗ Learning algorithms of the existing implementations do not optimize the real aim, i.e. the landmark localization error.

In this thesis, we address mainly the last two disadvantages, that is, how to learn parameters of the landmark detector efficiently and how to speed up a solution to the prediction problem.

Below we describe two instances of the DPM based landmark detectors that have been widely used by the community.

The detector of Everingham et al.

Everingham et al. [2006] combine a discriminative approach for training the appearance model with a generative approach to training shape model. Appearance model is learned by a variation of AdaBoost with Haar-like features. The shape is modeled by a mixture of Gaussian trees used to score landmark positions. The covariance matrix of each mixture component is restricted to form a tree structure with each random variable (landmark position) dependent on a single "parent" variable. It is an extension of [Felzenszwalb and Huttenlocher, 2005] which improves the ability to capture larger pose variations yet allowing an efficient search using Distance Transform (DT) [Felzenszwalb and Huttenlocher, 2005]. The follow-up work published in Sivic et al. [2009] extends the detector to work on profile images as well.

The detector of Zhu & Ramanan

Zhu and Ramanan [2012] couple the task of face detection, pose estimation and landmark localization. Their model is based on a mixture of trees with a shared pool of parts that correspond to facial landmarks. A mixture of trees is used to capture the topological changes due to the viewpoint. Each discretized viewpoint $\phi \in \Phi$ (corresponding to a yaw angle of the face orientation) is described by a dedicated tree (V_ϕ, E_ϕ) which is associated with a different set of landmarks V_ϕ with positions $(\boldsymbol{s}_i \mid i \in V_\phi) \in \mathcal{S}_\phi$. Given a viewpoint $\phi \in \Phi$, the match between image I and landmark positions $\boldsymbol{s} \in \mathcal{S}_\phi$ is evaluated by a score function $f_\phi \colon \mathcal{I} \times \mathcal{S}_\phi \to \mathbb{R}$ defined as:

$$f_\phi(I, \boldsymbol{s}; \boldsymbol{w}^\phi) = \sum_{i \in V_\phi} \left\langle \boldsymbol{w}_i^\phi, \boldsymbol{\Psi}(I, \boldsymbol{s}_i) \right\rangle + \sum_{ij \in E_\phi} \left[a_{ij}^\phi \delta x_{ij}^2 + b_{ij}^\phi \delta x_{ij} + c_{ij}^\phi \delta y_{ij}^2 + d_{ij}^\phi \delta y_{ij} \right] + \alpha^\phi \tag{2.26}$$

where $\delta x_{ij} = x_i - x_j$ and $\delta y_{ij} = y_i - y_j$ are differences of the x/y-coordinates of i-th and j-th landmarks. The vector $\boldsymbol{\Psi}(I, \boldsymbol{s}_i)$ denotes HOG features [Dalal and Triggs, 2005] computed from a patch captured from image I around position $\boldsymbol{s}_i$. The score function $f_\phi(I, \boldsymbol{s}; \boldsymbol{w}^\phi)$ is designed to be linear in appearance parameters $\{\boldsymbol{w}_i^\phi \mid i \in V_\phi, \phi \in \Phi\}$, deformation parameters $\{a_{ij}^\phi, b_{ij}^\phi, c_{ij}^\phi, d_{ij}^\phi \mid (i,j) \in E_\phi, \phi \in \Phi\}$ as well as the bias terms $\{\alpha^\phi \mid \phi \in \Phi\}$. Vector $\boldsymbol{w}^\phi$ contains a concatenation of all parameters defining the score function for the viewpoint $\phi \in \Phi$. Landmark configuration is predicted by solving a discrete maximization problem:

$$\hat{\boldsymbol{s}} \in \operatorname{Arg} \max_{\phi \in \Phi, \boldsymbol{s} \in \mathcal{S}_\phi} f_\phi(I, \boldsymbol{s}; \boldsymbol{w}^\phi) = \operatorname{Arg} \max_{\phi \in \Phi, \boldsymbol{s} \in \mathcal{S}_\phi} \langle \boldsymbol{w}, \boldsymbol{\Psi}(I, \boldsymbol{s}, \phi) \rangle \,, \tag{2.27}$$

where $\boldsymbol{w}$ is a concatenation of all parameters $\boldsymbol{w}^\phi$, $\phi \in \Phi$, and $\boldsymbol{\Psi}(I, \boldsymbol{s}, \phi)$ is appropriately defined map from $(I, \boldsymbol{s}, \phi)$ to the parameter space. Representing the score function as a dot product shows that the detector (2.27) is a special instance of general linear

classifier. The inner maximization w.r.t $\boldsymbol{s} \in \mathcal{S}_\phi$ can be solved efficiently by DP thanks to the tree structure of the graphs (V_ϕ, E_ϕ), $\phi \in \Phi$. Moreover, deformation cost separable in x and y coordinates allows to speed up the calculations by using the DT.

Neighborhood structures of tree graphs (V_ϕ, E_ϕ), $\phi \in \Phi$, are learned in the maximum likelihood fashion by the Chow-Liu algorithm [Chow and Liu, 2006]. This algorithm finds a tree structure best explaining the variation of landmark locations that are assumed to be normally distributed. Maximum likelihood problem is equivalent to computing a minimum spanning tree of a complete undirected graph where weight e_{ij} of each edge (i, j) is the mutual information between the location of the i-th and j-th landmark defined as

$$e_{ij} = \frac{1}{2}\left(\log|\Sigma_{\boldsymbol{s}_i}| + \log|\Sigma_{\boldsymbol{s}_j}| - \log|\Sigma_{\boldsymbol{s}_i,\boldsymbol{s}_j}|\right) . \tag{2.28}$$

Symbol $|\Sigma_{\boldsymbol{s}_i}|$ denotes determinant of a covariance matrix at position $\boldsymbol{s}_i$, and $|\Sigma_{\boldsymbol{s}_i,\boldsymbol{s}_j}|$ is determinant of a covariance matrix at positions $\boldsymbol{s}_i$, and $\boldsymbol{s}_j$. The covariance matrices are estimated from the training examples.

Having the graph structure (V_ϕ, E_ϕ), $\phi \in \Phi$, model parameters are learned in a fully supervised manner using the max-margin framework. The training set consists of positive examples $\mathcal{T}_+ = \{(I_+^1, \boldsymbol{s}_+^1, \phi_+^1), \ldots, (I_+^{m_+}, \boldsymbol{s}_+^{m_+}, \phi_+^{m_+})\}$, and negative examples $\mathcal{T}_- = \{I_-^1, \ldots, I_-^{m_-}\}$. A positive example $(I_+^i, \boldsymbol{s}_+^i, \phi_+^i)$ is a facial image with annotated landmark positions, and the viewpoint. A negative example I_-^i is just a non-facial image. Learning of the parameters $\boldsymbol{w}$ is formulated as a convex quadratic program:

$$(\boldsymbol{w}^*, \boldsymbol{\xi}^*) = \underset{\boldsymbol{w}\in\mathbb{R}^n, \boldsymbol{\xi}\in\mathbb{R}_+^m}{\operatorname{argmin}} \left[\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m_+}\xi_i^+ + C\sum_{i=1}^{m_-}\xi_i^-\right]$$

$$\begin{aligned} \text{s.t.} \quad \langle \boldsymbol{w}, \boldsymbol{\Psi}(I_+^i, \boldsymbol{s}_+^i, \phi_+^i)\rangle &\geq 1 - \xi_i^+ , && \forall i \in \{1, \ldots, m_+\} \\ \langle \boldsymbol{w}, \boldsymbol{\Psi}(I_-^i, \boldsymbol{s}, \phi)\rangle &\leq -1 + \xi_i^- , && \forall i \in \{1, \ldots, m_-\}, \forall \phi \in \Phi, \forall \boldsymbol{s} \in \mathcal{S}_\phi \\ w_k &\leq 0 , && \forall k \in K . \end{aligned} \tag{2.29}$$

The learning task (2.29) states that on positive examples the score of the correct landmark location and viewpoint should not be less than 1 otherwise a proportional linear penalty is added to the objective. Analogously, the score on the negative examples images evaluated for all configurations of landmark positions and viewpoints should not be higher than -1. Otherwise, it is penalized. The symbol K denotes a subset of indices of the quadratic spring terms $(a_{ij}^\phi, c_{ij}^\phi)$ in the joint parameter vector $\boldsymbol{w}$. Negativity constraints imposed on spring parameters ensure that the prediction task (2.27) can be solved by DT. Zhu and Ramanan [2012] propose solving the learning task (2.29) by a dual coordinate ascent algorithm [Yang and Ramanan, 2011].

The work of Zhu and Ramanan [2012] is closest to the method presented in this thesis. A conceptual difference in definition of the learning problem, compared to our approach, is that the landmark localization accuracy of the detector is not directly optimized when solving the task (2.29). The objective function of (2.29) is a sum of quadratic regularizer, and an upper bound on the number of mistakes provided the rule (2.27) is used for face detection. In other words, the objective is a convex proxy of the empirical risk when the predictions of landmark detector (2.27) are evaluated by the 0/1-loss function. In contrast, the objective function of our learning problem is directly related to landmark localization and viewpoint estimation error. Another important difference is that we mitigate the computationally difficult prediction problem by using fast, flexible features and by using a coarse-to-fine search strategy.

2.2.3. Constrained Local Models

The CLM [Cristinacce and Cootes, 2006, 2008], are similar to the tree based DPM described in the previous section. The main difference is that CLM use more complicated shape model that does not decompose over a tree. For example, a linear shape model (2.1) like in the case of AAM can also be used in CLM. Similarly to DPM, the best configuration of landmark positions is found by maximizing a score that combines responses of discriminatively trained local detectors and the shape score. More complex shape model used in CLM has a higher chance to avoid predicting physically implausible landmark configurations. However, this is paid off by ruling out the use of global optimization methods. The prediction problem can be described mathematically as follows

$$\hat{\boldsymbol{p}} \in \operatorname*{argmax}_{\boldsymbol{p}} \sum_{i=1}^{L} q_i(\mathbf{W}(\overline{\boldsymbol{s}}_i; \boldsymbol{p}), I) \;, \tag{2.30}$$

where $\mathbf{W}(\overline{\boldsymbol{s}}_i; \boldsymbol{p})$ is a location of i-th landmark projected to the image coordinates, $\boldsymbol{p} \in \mathbb{R}^n$ are the shape parameters, $q_i(\boldsymbol{s}, I)$ is the response of a local detector of i-th landmark evaluated at position $\boldsymbol{s} = (x, y)$ of image I, and $\{\overline{\boldsymbol{s}}_1, \ldots, \overline{\boldsymbol{s}}_L\}$ are the canonical positions of the landmarks.

A relation between the three based DPM and CLM can be directly seen after rewriting (2.30) as an equivalent problem:

$$(\hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_L) \in \operatorname*{argmin}_{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L} \left[\sum_{i=1}^{L} q_i(\boldsymbol{s}_i, I) + g(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L) \right] \tag{2.31}$$

where

$$g(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L) = \begin{cases} 0, & \text{if } \exists \boldsymbol{p} \text{ such that } \mathbf{W}(\overline{\boldsymbol{s}}_i; \boldsymbol{p}) = \boldsymbol{s}_i, \forall i \in \{1, \ldots, L\} \;, \\ \infty, & \text{otherwise} \;. \end{cases} \tag{2.32}$$

The shape score $g(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L)$ allows only those landmark positions which can be generated by the shape model. In case of the tree based DPM the shape score $g(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L)$ is replaced by a sum of simpler functions $\sum_{i,j \in E} g_{ij}(\boldsymbol{s}_i, \boldsymbol{s}_j)$. In contrast to CLM, the tree based DPM thus use soft constraints on the shape, and they replace the continuous landmark positions by discrete ones, which makes the prediction problem globally solvable by dynamic programming.

The advantages and disadvantages of CLMs are:

✓ Learning local landmark detectors seems to be an easier problem, compared to learning of holistic face appearance as done in AAM.

✓ The shape model is more appropriate than the one used in three-based DPM.

✗ Prediction problem is a highly non-convex task. The existing solvers return only a locally optimal solution quality of which depends on the initial estimate.

✗ The existing implementations do not optimize in the learning stage the true objective like the landmark localization error of the entire detector.

3. Detector

In this chapter, we describe the proposed detector in detail. We begin with the formal definition of single-view DPM based detector in Section 3.1, which is then extended to a multi-view scenario in Section 3.2. Sections 3.3, and 3.4 describe the internal representation of unary and pairwise potentials, representing the appearance and the shape model, respectively. Section 3.5 discusses the inference calculation. Finally, Section 3.6 suggests a speed-up strategy, based on the coarse-to-fine search, for a DPM based detector.

3.1. Single-view DPM Detector

We follow the tree based DPM approach [Fischler and Elschlager, 1973; Felzenszwalb et al., 2010; Felzenszwalb and Huttenlocher, 2005] translating estimation of landmarks configuration (i.e. the face shape) into maximization of a score which evaluates the match between the input image and optimized positions of landmarks. Shape model is represented by an undirected graph $G = (V, E)$, where V is a finite set of vertices representing landmarks and $E \subset \binom{V}{2}$ is a set of edges between pairs of landmarks, whose positions are related. Examples of particular graphs used in the proposed detector are shown in Figure 3.1. Let $\boldsymbol{s} = (\boldsymbol{s}_i \in \{1, \dots, W\} \times \{1, \dots, H\} \mid i \in V)$ be a configuration of landmark (pixel) positions in image $I \in \mathcal{I}^{H \times W}$ (so called *normalized frame*). Landmark configurations are *a priory* restricted to be from a predefined area, $\boldsymbol{s} \in \mathcal{S} = \mathcal{S}_0 \times \cdots \times \mathcal{S}_{|V-1|}$, where $\mathcal{S}_i \subseteq \{1, \dots, H\} \times \{1, \dots, W\}$ denotes a *search space* of i-th landmark. Let $\boldsymbol{w}$ denote vector of parameters composed of parameters $\boldsymbol{w}_i \in \mathbb{R}^{n_i}$ and $\boldsymbol{w}_{ij} \in \mathbb{R}^{n_{ij}}$ (n_i, where n_{ij} denote the number of parameters) associated with the unary, and pair-wise potentials, respectively. Then, the scoring function, and detector $h \colon \mathcal{I}^{H \times W} \to \mathcal{S}$, are defined as follows:

If G is a tree $\Rightarrow$ inference with optimality guarantees (Section 3.5).

$$
\begin{aligned}
f(I, \boldsymbol{s};\ \boldsymbol{w}) &= \sum_{i \in V} q_i(\boldsymbol{s}_i, I;\ \boldsymbol{w}_i) + \sum_{(i,j) \in E} g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{w}_{ij}) \\
h(I;\ \boldsymbol{w}) &= \arg\max_{\boldsymbol{s} \in \mathcal{S}} f(I, \boldsymbol{s};\ \boldsymbol{w})\ .
\end{aligned} \tag{3.1}
$$

"max-sum" problem

First part of the scoring function represents an *appearance model* and it is composed of the unary potentials $q_i(\boldsymbol{s}_i, I;\ \boldsymbol{w}_i)$, measuring the quality of a fit of individual landmark positions $\boldsymbol{s}_i$, $i \in V$, to image I. The second part represents a *deformation cost* and it is composed of the pair-wise potentials $g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{w}_{ij})$, measuring the likelihood of mutual positions of connected pairs of landmarks. In our model the pair-wise functions $g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{w}_{ij})$ have the same form (and thus n_{ij} is a constant $\forall (i, j) \in E$) but every edge $(i, j) \in E$ has different parameters $\boldsymbol{w}_{ij}$. The unary potentials $q_i(\boldsymbol{s}_i, I;\ \boldsymbol{w}_i)$, $i \in \mathcal{V}$, have different parameters $\boldsymbol{w}_i \in \mathbb{R}^{n_i}$ as well as dimensions n_i which results from describing individual landmarks by patches of different size.

The normalized frame $I \in \mathcal{I}^{H \times W}$, serving as an input of the detector, is constructed from the response of a face detector. The face detector provides an estimate of the position, scale, and in-plane rotation of the face. To compensate the imprecision of the face detector, we extend the face box by a multiple of its size. Finally, we apply a similarity transformation to obtain a normalized frame of a fixed size. The process of normalized frame acquisition is illustrated in Figure 3.2.

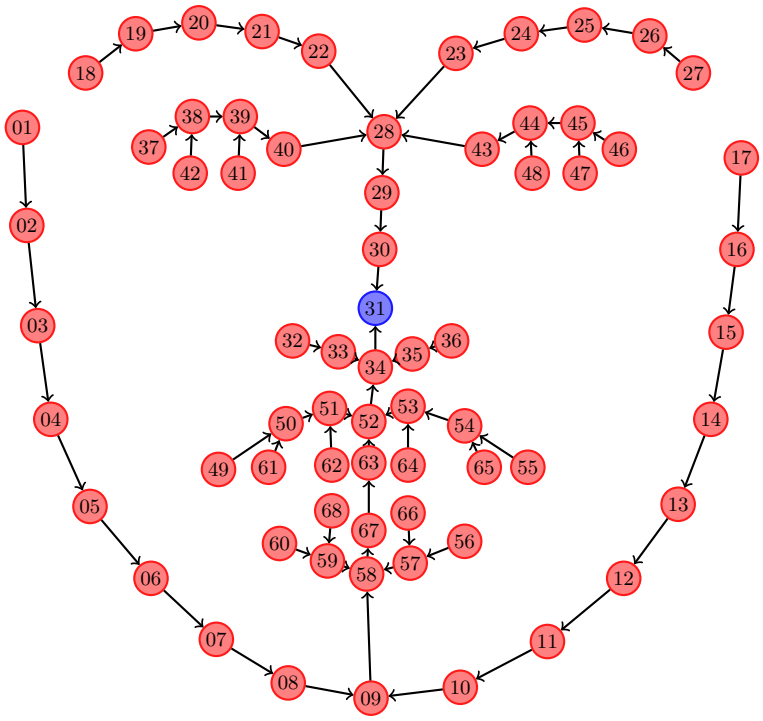


Figure 3.1. The graph structure of the 68 landmarks configuration. Note, that the graph forms a tree rooted at the landmark emphasized by a blue circle.

3.2. Multi-view DPM detector

Let $\phi \in \Phi$ be a discretized viewpoint (or, equivalently, the yaw angle of a head pose, see Figure 3.4 for illustration). Then, the multi-view facial landmark detector is defined as $h_{\text{mv}} \colon \mathcal{I}^{H \times W} \to \Phi \times \mathcal{S}$. That is, the detector is supposed to estimate the viewpoint simultaneously with landmarks positions. Predicted landmarks configuration $\boldsymbol{s}$ might consist of a different number of landmarks for each discretized viewpoint ϕ, hence $\mathcal{S} = \cup_{\phi \in \Phi} \mathcal{S}^{\phi}$, where $\mathcal{S}^{\phi}$ denotes the search space for viewpoint ϕ.

Table 3.1. Definition of the discretized viewpoint that corresponds to an angle from which we observe the face.

Viewpoint ($\phi \in \Phi$)				
$-$profile	$-$half-profile	frontal	half-profile	profile
Range of angles				
$(-110°, -60° >$	$(-60°, -15° >$	$(-15°, 15°)$	$< 15°, 60°)$	$< 60°, 110°)$
Number of landmarks detected in ϕ **(i.e.** $\lvert V^{\phi} \rvert$**)**				
13	19	21	19	13

The set Φ may be arbitrary. However, we use a following discretization $\Phi = \{-\text{profile}, -\text{half-profile}, \text{frontal}, \text{profile}, \text{half-profile}\}$. The precise ranges, as well as the number of landmarks, are summarized in Table 3.1. Graphs representing landmark connections are depicted in Figure 3.3.

In next sections, we describe two distinct approaches formulating joint multi-view facial landmark detector.

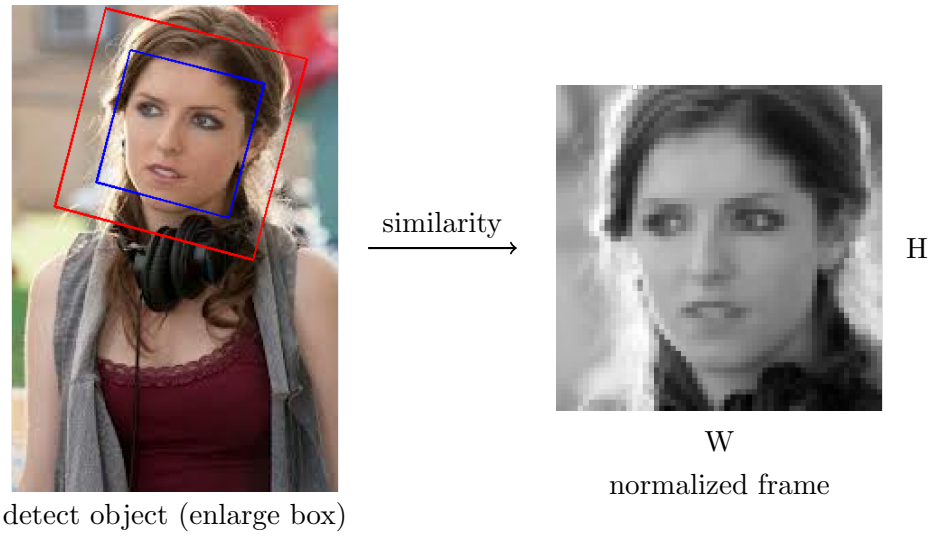


Figure 3.2. The acquisition of "normalized frame". The blue box represents the output of the face detector; red box is the face box enlarged by a pre-defined margin. The similarity transformation (removing the possible in-plane rotation and scaling the image to a fixed size) is applied to the red box, and the normalized frame is obtained.

3.2.1. Single-stage approach

Single-stage approach of multi-view DPM detector is a straightforward extension of (3.1) obtained by including the viewpoint $\phi \in \Phi$ among the hidden parameters to be estimated in a joint formulation. Shape model is described for each ϕ independently, by an undirected graph $G^\phi = (V^\phi, E^\phi)$. Scoring function $f_\phi(I, \boldsymbol{s};\ \boldsymbol{w}^\phi)$ serves two purposes: i) it ranks the configuration of landmarks, and ii) it ranks the viewpoint. Formulae describing the multi-view detector are as follows:

joint multi-view detector

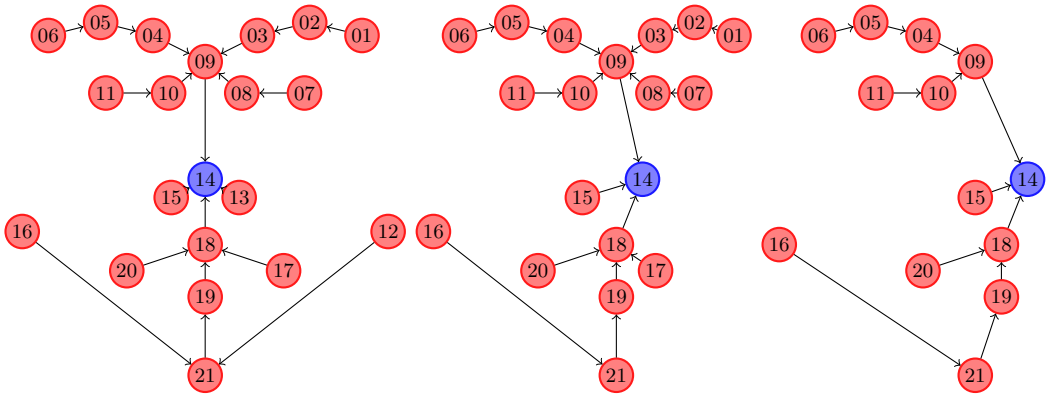
$$\begin{aligned} f_\phi(I, \boldsymbol{s};\ \boldsymbol{w}^\phi) &= \sum_{i \in V^\phi} q_i(\boldsymbol{s}_i, I;\ \boldsymbol{w}_i^\phi) + \sum_{(i,j) \in E^\phi} g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{w}_{ij}^\phi) + b^\phi \\ h_{\text{mv}}(I;\ \boldsymbol{w}) &= \arg \max_{\phi \in \Phi,\ \boldsymbol{s} \in \mathcal{S}^\phi} f_\phi(I, \boldsymbol{s};\ \boldsymbol{w}^\phi)\ , \end{aligned} \tag{3.2}$$

where detector $h_{\text{mv}} \colon \mathcal{I} \to \Phi \times \mathcal{S}$ predicts both viewpoint, and landmarks positions. $q_i(\boldsymbol{s}_i, I;\ \boldsymbol{w}_i^\phi)$ and $g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{w}_{ij}^\phi)$ are the viewpoint specific appearance models and deformation costs, respectivel. b^ϕ is a bias term compensating different number of landmarks in each view. Note, that the problem (3.2) can be decomposed into $|\Phi|$ independent single-view detection problems and a subsequent selection of the viewpoint with the highest score function.

In formulation (3.2), a single set of score functions is used for both the prediction of landmarks' position, and prediction of the viewpoint. A landmark detector based on the same detection strategy has been used in [Zhu and Ramanan, 2012]. In the next section, we propose a different approach which splits the prediction into two stages.

3.2.2. Two-stage approach

The single-stage multi-view detector is formulated clearly and concisely. However, its drawback lies in a challenging learning procedure (c.f. Chapter 4) which, if based on using SO-SVM, involves solving a single but complex optimization problem. To ease the problem, we propose a two-stage detection strategy. The main idea is to train a single-view detector independently for each view $\phi \in \Phi$ in the first stage, then



(a) frontal **(b)** half-profile **(c)** profile

Figure 3.3. Graph structure of several view-specific detectors shown for positive viewpoints (the negative ones are mirrored). Note, that all graphs form a tree rooted at the vertex emphasized by a blue circle.

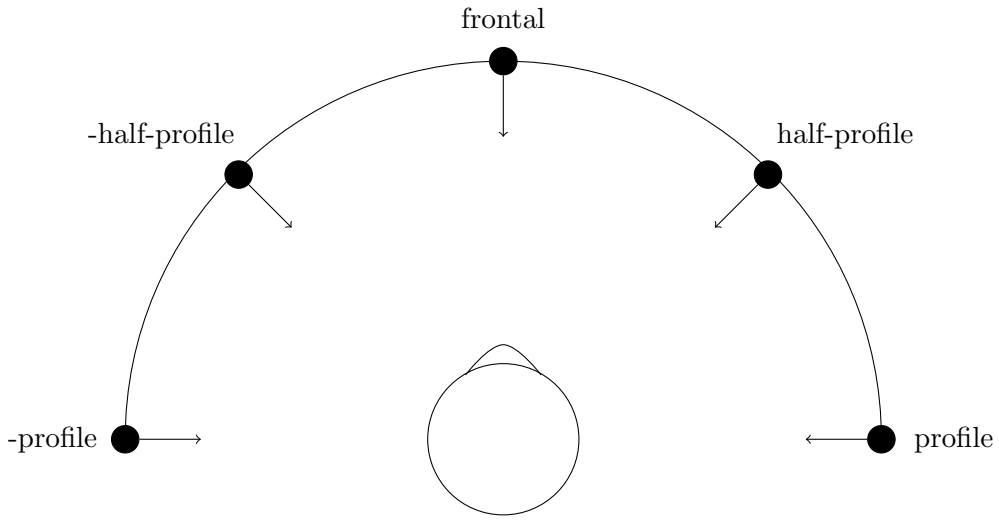


Figure 3.4. Viewpoint definition. The multi-view detector can be viewed as a detector of the viewpoint from which we observe the human head. Alternatively, as the discretized yaw angle detector assuming the camera is fixed, and the head rotates.

consider the landmark localization as fixed, and consequently extract features from fixed landmark positions for learning a simple multi-class viewing angle classifier in the second stage.

Let us denote a set of parameters to be used in the first stage by $\boldsymbol{w}_1^\phi$, and a different set of parameters to be used in the second stage by $\boldsymbol{w}_2^\phi$. Landmarks found for the particular viewpoint ϕ are denoted as $\boldsymbol{s}_\phi$, and after the first stage, we obtain them for all viewpoints $\phi \in \Phi$. The goal of the second stage classifier is just to detect the correct viewpoint ϕ which best describes the given image using parameters $\boldsymbol{w}_2$. More precisely, the two-stage detector is defined as follows:

$$\begin{aligned} \hat{\boldsymbol{s}}_\phi &\in \operatorname*{Argmax}_{\boldsymbol{s}\in\mathcal{S}^\phi} f_\phi(I, \boldsymbol{s}, \boldsymbol{w}_1^\phi), \quad \phi \in \Phi\,, \\ \hat{\phi} &\in \operatorname*{Arg\,max}_{\phi\in\Phi} f_\phi(I, \hat{\boldsymbol{s}}_\phi, \boldsymbol{w}_2^\phi)\,, \\ h(I, \boldsymbol{s}, \boldsymbol{w}) &= (\hat{\boldsymbol{s}}_\phi, \hat{\phi})\,, \end{aligned} \tag{3.3}$$

where the score function $f_\phi(I, \boldsymbol{s}, \boldsymbol{w})$ is the same as defined in (3.2) and $\boldsymbol{w} = (\boldsymbol{w}_k^\phi \mid k \in$

$\{1, 2\}$, $\phi \in \Phi$) is a joint parameter vector composed of parameters for the two stages. Two-stage detector (3.2.2) has some advantages over the single-stage one (3.2):

1. Learning of the two-stage detector decomposes into $|\Phi|+1$ independent optimization problems which are significantly simpler than solving a single complex problem required when learning the single-stage detector. Moreover, defining loss functions for the two-stage approach is straightforward in contrast to the single-stage approach where a single loss has to penalize two types of prediction errors simultaneously (i.e. landmark localization error, and viewpoint prediction error).

2. Evaluation of the two-stage detector has the same computational complexity as the single-stage detector, yet the former is more flexible. It is seen, that in the special case when $\boldsymbol{w}_1^\phi = \boldsymbol{w}_2^\phi$, $\phi \in \Phi$, detectors (3.2), and (3.2.2) are equivalent. However, the freedom to learn specialized weights for landmark localization $\boldsymbol{w}_1^\phi$, $\phi \in \Phi$, and viewpoint prediction $\boldsymbol{w}_2^\phi$, $\phi \in \Phi$, separately, promises an improvement in the accuracy as will be confirmed experimentally in Chapter 5.

3.3. Appearance Model

The appearance model is a linearly parameterized function

$$q_i(\boldsymbol{s}_i, I;\ \boldsymbol{w}_i^\phi) = \langle \boldsymbol{w}_i^\phi, \boldsymbol{\Psi}_i^\phi(I, \boldsymbol{s}_i) \rangle\ , \tag{3.4}$$

unary potentials

where $\boldsymbol{\Psi}_i^\phi(I, \boldsymbol{s}_i) \colon \mathcal{I} \times \mathcal{S}_i^\phi \to \mathbb{R}^{n_i^\phi}$ denotes a feature descriptor of a patch cropped from image I around position $\boldsymbol{s}_i$. Our approach allows using arbitrary feature descriptor. We have experimented with several descriptors including the normalized intensity values, their derivatives, Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005], and the multi-scale pyramid of S-LBP [Sonnenburg and Franc, 2010; Uřičář et al., 2012]. The S-LBP bring a favorable trade-off between the localization accuracy and speed. Weight vectors $\boldsymbol{w}_i^\phi \in \mathbb{R}^{n_i^\phi}$, $i \in V_\phi$, are learned from examples as will be described in Chapter 4. In Sections 3.3.1, we describe proposed S-LBP descriptor and, in Section 3.3.2 a method to accelerate its computation via MIPMAP. The HOG and SIFT descriptors, most commonly used options in the landmark detection, are outlined in Section 3.3.3 and 3.3.4, respectively.

3.3.1. Sparse Pyramid of Local Binary Patterns

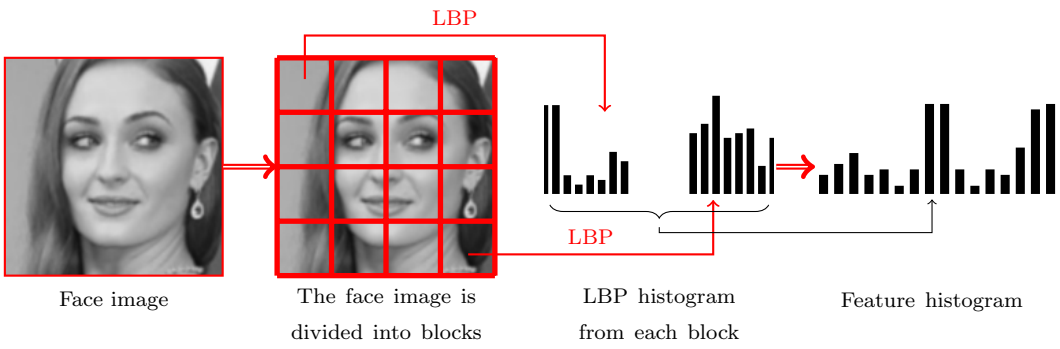


Figure 3.5. The LBP features are standardly computed from a fixed-size grid.

The S-LBP is a based on the LBP [Ojala and Pietikäinen, 1994] histogram (standardly used in computer vision tasks, especially for representing the texture [Ojala and

Pietikäinen, 1994], or facial features [Hadid et al., 2008]). The standard approach is to compute weighted histograms of LBP obtained from a fixed grid, see Figure 3.5 for illustration.

Similarly as in the case of LBP histogram, the S-LBP descriptor evaluates the standard 3×3 LBP [Ojala et al., 2002] in each position of the original patch. Each 8bit LBP code is represented by a binary vector composed of all zeros and a single one, whose position is determined by the LBP code. Then the patch is downscaled by a factor of two, and the LBP are computed again in all positions. This process is repeated until the resolution of the downscaled patch is below 3×3 pixels. The resulting sequence of binary vectors is concatenated to a column vector, forming the final descriptor. The resulting sparse high-dimensional S-LBP descriptor can be best represented by the indices of its components equal to one. To give an example of its dimensionality, consider a patch of size 15×15 pixels. The number of all 3×3 px sub-windows in all levels of the scale pyramid is $(13 \times 13) + (5 \times 5) + (1 \times 1) = 195$. Since each LBP is represented by a 256-dimensional binary vector, the resulting descriptor has $n_i^\phi = 195 \cdot 256 = 49,920$ components, with only 195 non-zero entries.

Proposed S-LBP descriptor has several advantages compared to the standard histogram of LBP. The linear score function (3.4) which uses the standard histogram of LBPs is defined as a weighted sum of relative occurrences of LBP codes computed in a grid defined on the patch. In contrast, when the S-LBP descriptor is used, each LBP in the patch has its weight. This implies that the detectors using S-LBP descriptor subsume the detectors with the LBP histogram as a special case. Moreover, using the S-LBP descriptor does not require the manual definition of the grid, but everything is learned from the examples. The advantages are paid off by a higher number of weights to be learned and, consequently, an increased risk of over-fitting.

3.3.2. Acceleration of S-LBP via Using MIPMAP

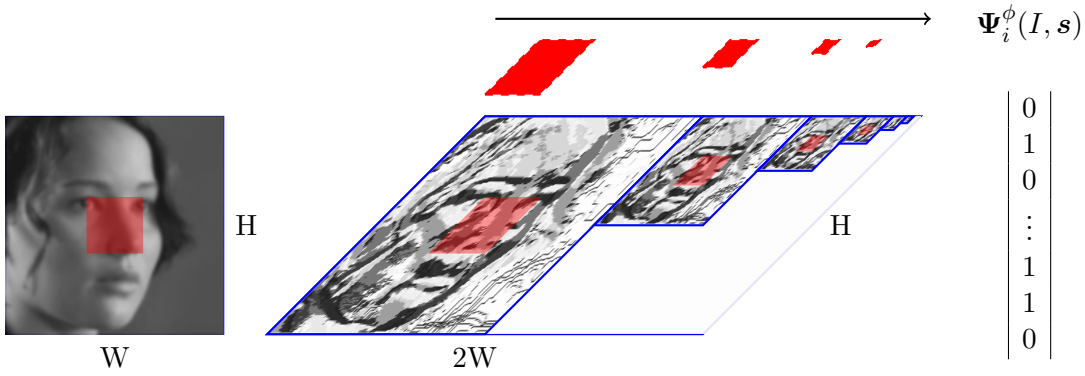


Figure 3.6. Features are pre-computed in all positions and scales of the normalized frame and stored in the form of MIPMAP. The final S-LBP descriptor $\boldsymbol{\Psi}_i^\phi(I, \boldsymbol{s})$ is compiled from the MIPMAP on the fly by stacking the corresponding features.

Finding the optimal landmark locations requires computation of S-LBP features in patches centered at all positions where we search for landmarks. A naïve implementation results in a significant number of repetitive evaluations of the base LBP feature descriptor, since the search patches are highly overlapped. We propose to pre-compute the base LBP in all scales of the entire normalized frame. The resulting LBP codes are represented in the form of a MIPMAP [Williams, 1983], which allows efficient indexing of the corresponding features in different scales. The final S-LBP descriptor is then

compiled from the MIPMAP on the fly (see Figure 3.6 for illustration).

This approach makes the feature computation independent of the number of sought landmarks (assuming that the computational demand of feature compilation is negligible), leading to about 40% speedup compared to the naïve implementation. More importantly, this approach allows us to share the pre-computed features among different views making the final structured classifier only sub-linearly slower, compared to the naïve strategy evaluating the individual DPM detectors from scratch. Note, that the feature descriptor evaluated via the MIPMAP representation is not the same as the original S-LBP descriptor. Using the MIPMAP representation leads to skipping some base LBP features computed in lower scales. However, we found that it has no impact on the accuracy of the detector.

3.3.3. Histogram of Oriented Gradients

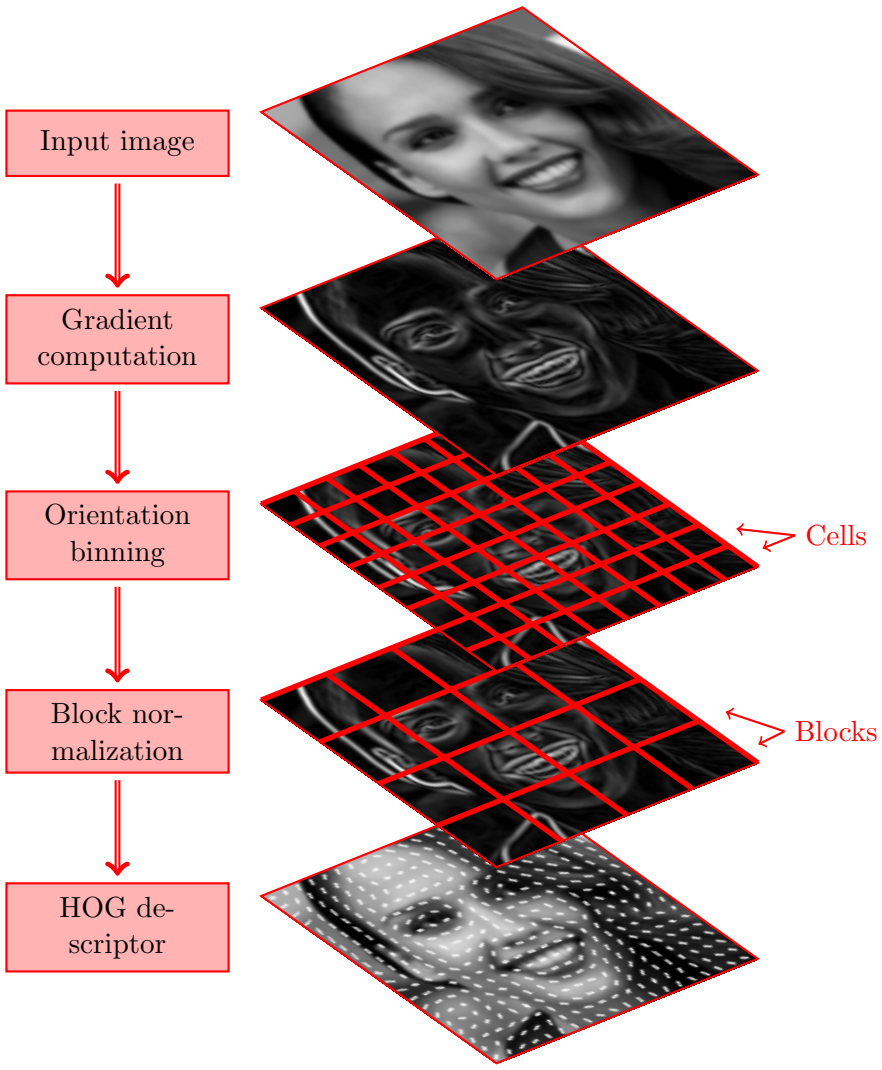


Figure 3.7. The process of HOG features calculation. First, the gradient image is computed. Then, each pixel contributes by a weighted vote to the orientation histograms; these are grouped into cells. The blocks are constructed on top of the cells. Finally, the feature descriptor is obtained by normalizing over the overlapping blocks.

The computation of HOG features [Dalal and Triggs, 2005] goes as follows. The first step computes image derivatives by a convolution with centered discrete deriva-

tive masks ($[-1, 0, 1]^\top$, $[-1, 0, 1]$) without Gaussian smoothing. The second step is a spatial/orientation binning. Each pixel contributes by a weighted vote to 9 bins histograms. Votes are bilinearly interpolated between neighboring bin centers and accumulated into spatial regions called cells. The last step is a block normalization, where histograms accumulated in cells are normalized among the overlapping cells called blocks. Then, the feature descriptor is generated. The whole process is illustrated in Figure 3.7.

The HOG features were successfully used e.g. by Zhu and Ramanan [2012]. However, we found the S-LBP features more suitable for our needs, mainly due to a much faster calculation. See Section 5.5 for the comparison of different feature descriptors timing.

3.3.4. Scale Invariant Feature Transform

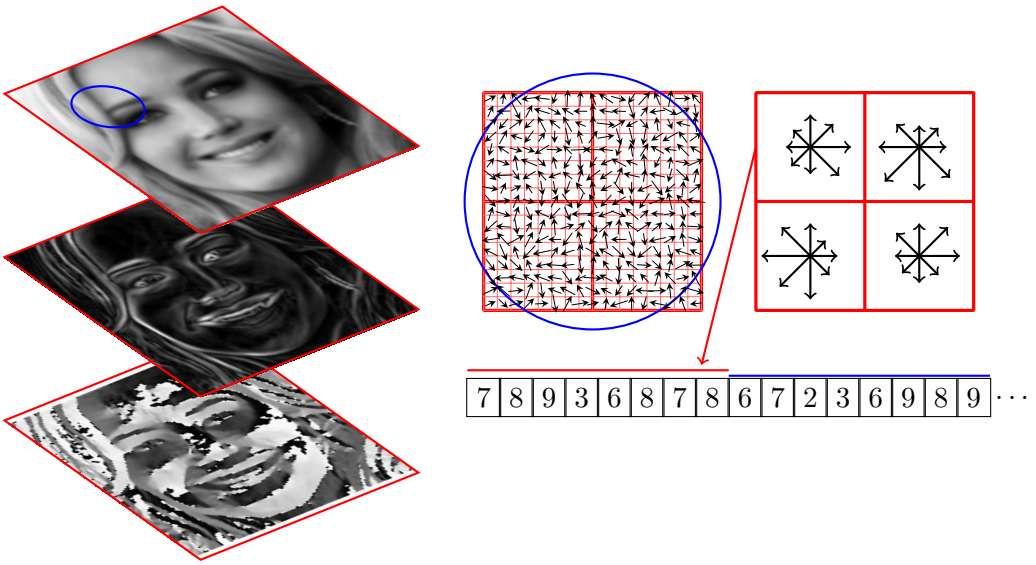


Figure 3.8. The process of SIFT features calculation. First, the gradient image is computed. Then, each pixel contributes by a weighted vote (by a Gaussian window) to the orientation bins, which are subsequently accumulated into orientation histograms, representing the 4×4 sub-regions. The feature descriptor is usually represented by a 128 dimensional vector.

The SIFT descriptor, introduced by Lowe [2004], is created by computing the gradient magnitude and orientation at each image sample point weighted by a Gaussian window. Afterward, the samples are accumulated into orientation histograms representing the 4×4 sub-regions. Usually, the 128 dimensional vector is used. Figure 3.8 depicts the SIFT features calculation.

Xiong and la Torre [2013] used SIFT features with a great success. They show that it can be used for real-time applications, and achieve impressive localization accuracy. However, we should point out, that in their case, the SIFT features are not evaluated in each possible position of the search space. See Section 5.5 for the comparison of different feature descriptors timing.

3.4. Deformation Costs

As the deformation cost $g\colon \mathbb{N}^2 \to \mathbb{R}$ we use quadratic function of coordinate displacements defined as

par-wise potentials

$$g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z}) = z_1\delta x + z_2\delta y + z_3\delta x^2 + z_4\delta y^2\ , \tag{3.5}$$

where δx, δy denote coordinates of the displacement vector

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \boldsymbol{s}_i - \boldsymbol{s}_j = \begin{bmatrix} x_i - x_j \\ y_i - y_j \end{bmatrix} \tag{3.6}$$

and $\boldsymbol{z} = [z_1, z_2, z_3, z_4]^T \in \mathbb{R}^4$, is a placeholder of parameter vectors $\boldsymbol{w}_{ij} \in \mathbb{R}^{n_{ij}}$, $n_{ij} = 4$, $(i,j) \in E$, or $\boldsymbol{w}_{ij}^{\phi} \in \mathbb{R}^4$, $(i,j) \in E$,$\phi \in \Phi$, respectively. It is seen, that the deformation cost (3.5) is linear in its parameters and hence it can be written as a dot product

$$g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{\Psi}_E(\boldsymbol{s}_i, \boldsymbol{s}_j) \rangle \ , \tag{3.7}$$

where the feature map $\boldsymbol{\Psi}_E \colon \mathbb{N}^2 \to \mathbb{R}^4$ reads

$$\boldsymbol{\Psi}_E(\boldsymbol{s}_i, \boldsymbol{s}_j) = \begin{bmatrix} \delta x \\ \delta y \\ \delta x^2 \\ \delta y^2 \end{bmatrix} \ . \tag{3.8}$$

The same quadratic deformation cost has been used by [Felzenszwalb et al., 2010]. The choice of the quadratic cost is motivated by its relation to the logarithm of the Normal distribution defined on the displacement vector $\boldsymbol{s}_i - \boldsymbol{s}_j$. To see this, we can reformulate (3.5) as follows:

$$g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z}) = \underbrace{\begin{bmatrix} x_i \\ y_i \\ x_j \\ y_j \end{bmatrix}^{\top}}_{\boldsymbol{x}^{\top}} \underbrace{\begin{bmatrix} z_3 & 0 & -z_3 & 0 \\ 0 & z_4 & 0 & -z_4 \\ -z_3 & 0 & z_3 & 0 \\ 0 & -z_4 & 0 & z_4 \end{bmatrix}}_{\Sigma_{ij}^{-1}} \underbrace{\begin{bmatrix} x_i \\ y_i \\ x_j \\ y_j \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ -z_1 \\ -z_2 \end{bmatrix}^{\top}}_{\boldsymbol{b}^{\top}} \underbrace{\begin{bmatrix} x_i \\ y_i \\ x_j \\ y_j \end{bmatrix}}_{\boldsymbol{x}} \ . \tag{3.9}$$

Which can be rewritten (3.9) as

$$g_{ij}(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z}) = -(\boldsymbol{x} - \boldsymbol{\mu}_{ij})^{\top} \Sigma_{ij}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_{ij}) + \text{const.} \ , \tag{3.10}$$

from which, we can see the relation of the parameters $\boldsymbol{z}$ with $\boldsymbol{\mu}_{ij}$, and Σ_{ij}^{-1}. The latter are the parameters of multi-variate normal distribution, which can be viewed as a generative model of the displacement. Note, that the terms containing z_1, and z_2 are hidden in $\boldsymbol{\mu}_{ij}$, to be more specific, $z_1 = 2z_3(\mu_{ij}^3 - \mu_{ij}^1)$, and $z_2 = 2z_4(\mu_{ij}^4 - \mu_{ij}^2)$). In this perspective, (3.5) is related to the log-likelihood of this generative model of the displacement.

Another advantage of having the deformation cost in the form of a separable quadratic function is a possibility to use DT [Felzenszwalb and Huttenlocher, 2012] for solving the max-sum problem (3.1) in time linearly dependent on the number of positions searched. DT will be described more in detail in Section 3.5.2.

3.5. Inference Problem

Evaluation of detector (3.1), as well as (3.2), amounts to solving an instance of the max-sum optimization problem. A tractability of the max-sum problem depends on the complexity of graph G. Let us suppose for simplicity that all landmarks have the same search space $\mathcal{S}'$. There is overall $|\mathcal{S}'|^{|V|}$ possible solutions. For a general graph, the max-sum problem is known to be NP-hard. However, there are polynomial algorithms for certain graph structures (e.g. sequences [Rabiner, 1989], trees [Felzenszwalb and Zabih, 2010], or a low-width graphs [Amit and Kong, 1996]). While our framework does not limit the graph structure in general, for the sake of speed, we set the structure of graph G to form a tree. In such case, the inference can be computed by DP in $\mathcal{O}(|V||\mathcal{S}'|^2)$ time.

3.5.1. Dynamic Programming on a Tree Graph

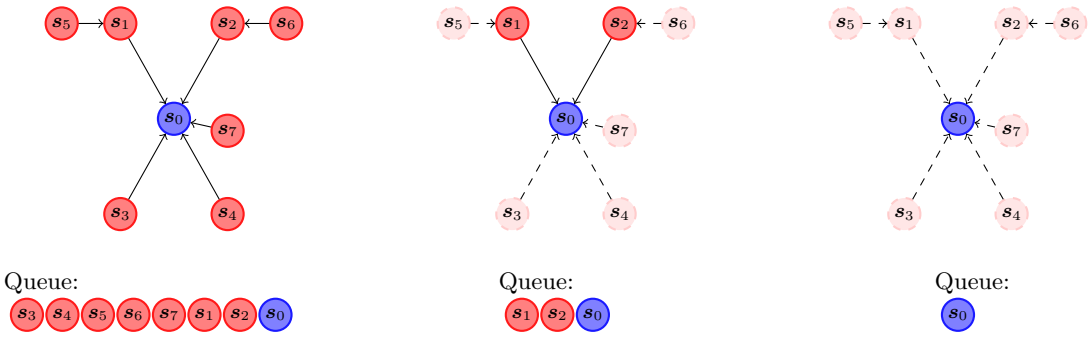


Figure 3.9. The illustration of step-by-step calculation of the inference, i.e. the max-sum problem, for a tree graph. The graph is topologically sorted, and the resulting queue is used to direct the calculations. Vertices are popped from the queue and DP tables B_i and B'_i are calculated using (3.11), and (3.12), respectively. The graph is progressively shrinking until there is just the last vertex, i.e. the root, to be processed.

DP for the max-sum on a tree is organized in a similar way as for sequences (i.e. the well known Viterbi algorithm [Rabiner, 1989]). The main idea is illustrated in Figure 3.9. We consider a directed graph G in the following, although the algorithm can be applied to an undirected graph as well, by selecting an arbitrary vertex as the root.

The graph is first topologically sorted, the root vertex to be evaluated as the last one. Then, the first vertex $\boldsymbol{s}_i$ in the queue is picked up, and its children vertices C_i are examined. The DP is filling the tables $B_i[\boldsymbol{s}_i]$, $B'_i[\boldsymbol{s}_i]$, $\forall \boldsymbol{s}_i$, storing the score of the optimal partial solution, and the corresponding label, respectively

$$B_i[\boldsymbol{s}_i] = q_i(\boldsymbol{s}_i, I;\ \boldsymbol{w}_i) + \sum_{\boldsymbol{s}_j \in C_i} \max_{\boldsymbol{s}_j} \left(B_j[\boldsymbol{s}_j] + g_{ij}(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z})\right) \tag{3.11}$$

$$B'_i[\boldsymbol{s}_i] = \arg\max_{\boldsymbol{s}_i} \left(B_i[\boldsymbol{s}_i] + g_{ij}(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z})\right) \ . \tag{3.12}$$

Note, that in case of $C_i = \emptyset$, we have $B_i[\boldsymbol{s}_i] = q_i(\boldsymbol{s}_i, I;\ \boldsymbol{w}_i)$. After computing $B_i[\boldsymbol{s}_i]$, and $B'_i[\boldsymbol{s}_i]$, vertices with $C_i = \emptyset$ are cut off graph G. The calculations stop after the last vertex in the queue, i.e. the root vertex $\boldsymbol{s}_r$, is evaluated. $B_i[\boldsymbol{s}_r]$ contains the global optimum of the scoring function, while $B'_i[\boldsymbol{s}_i]$ are used to construct $\boldsymbol{s}^*$, i.e. the best configuration of the landmarks positions, by backtracking from the root vertex to the leaves.

The overall algorithm runs in $\mathcal{O}(|V||\mathcal{S}'|^2)$ time. However, it can be significantly sped up to $\mathcal{O}(|V||\mathcal{S}'|)$, if $g_{ij}(\boldsymbol{s}_j, \boldsymbol{s}_j;\ \boldsymbol{z})$ is of a special form, by incorporating DT [Felzenszwalb and Huttenlocher, 2012], which is described in the next section.

3.5.2. Distance Transform

Let us look closer on the form of equations (3.11), and (3.12). We know, that the landmark coordinates $\boldsymbol{s}_i$,$\boldsymbol{s}_j \in \mathbb{N}^2$ are points of a discrete 2D grid. Furthermore, $g(\boldsymbol{s}_j, \boldsymbol{s}_j;\ \boldsymbol{z})$ is defined as a quadratic surface between $\boldsymbol{s}_i$, and $\boldsymbol{s}_j$, of a form $z_1\delta x + z_2\delta y + z_3\delta x^2 + z_4\delta y^2$, $z_1, \ldots, z_4 \in \boldsymbol{z}$, $z_1, \ldots, z_4 \in \mathbb{R}$. Its concavity can be enforced by constraining the individual components of weight vectors $\boldsymbol{z}$ appropriately. With this in mind, we can see the similarity between (3.11), (3.12), and generalized DT [Felzenszwalb and Huttenlocher, 2012]

$$\mathcal{D}_f(p) = \min_{q \in \mathcal{G}} \left((p-q)^2 + f(q)\right) \ , \tag{3.13}$$

where q are points from a grid $\mathcal{G}$, and $f(q)$ is an arbitrary function on a grid. Transformation of (3.13) to maximization if the function is concave is straightforward. For the sake of simplicity, we will restrain ourselves with the minimization in the following text.

Let us expand $g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{\Psi}_E(\boldsymbol{s}_i, \boldsymbol{s}_j)\rangle$:

$$g_{ij}(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z}) = z_1(x - x') + z_2(y - y') + z_3(x - x')^2 + z_4(y - y')^2 \ , \tag{3.14}$$

where $z_1, \ldots, z_4$ are the components of weight vector $\boldsymbol{z}$, x, y, and x', y' are x and y coordinates, of $\boldsymbol{s}_i, \boldsymbol{s}_j$, respectively. First, we need to show that DT can be used for this 2D function.

$$\begin{aligned}
\mathcal{D}_f(x,y) &= \min_{x',y'} \left[w_3(x-x')^2 + w_1(x-x') + w_4(y-y')^2 + w_2(y-y') + f(x',y')\right] \\
&= \min_{x'} \left[w_3(x-x')^2 + w_1(x-x') + \min_{y'} \left[w_4(y-y')^2 + w_2(y-y') + f(x',y')\right]\right] \\
&= \min_{x'} \left[w_3(x-x')^2 + w_1(x-x') + \mathcal{D}_{f|_{x'}}(y)\right] \ ,
\end{aligned} \tag{3.15}$$

DT is applicable in an arbitrary dimension.

where $\mathcal{D}_{f|_{x'}}(y)$ denotes a 1D distance transform (1D-DT) of f, restricted to the column indexed by x'. Therefore, we can apply the same algorithm as for the 1D-DT computation— first for the columns and then for the rows.

Algorithm 1 1D-DT for quadratic function of (3.5). Modified version of [Felzenszwalb and Huttenlocher, 2012, Algorithm 1].

Require: $a, b, l, d_{\text{len}}, d_{\text{shift}}$

Initialization:
  $k \leftarrow 0$
  $v[0] \leftarrow 0$ {Index of the rightmost parabola in lower envelope}
  $z[0] \leftarrow -\infty$ {Locations of parabolas in lower envelope}
  $z[1] \leftarrow +\infty$ {Locations of boundaries between parabolas}
for $q = 1$ **to** $l - 1$ **do**
  $s \leftarrow \frac{(f(q)-f(v[k]))-b(q-v[k])+a(q^2-v[k]^2)}{2a(q-v[k])}$ {Compute lower envelope}
  while $s \leq z[k]$ **do**
    $k \leftarrow k - 1$
    $s \leftarrow \frac{(f(q)-f(v[k]))-b(q-v[k])+a(q^2-v[k]^2)}{2a(q-v[k])}$
  end while
  $k \leftarrow k + 1$
  $v[k] \leftarrow q$
  $z[k] \leftarrow s$
  $z[k+1] \leftarrow -\infty$
end for
$k \leftarrow 0$
$q_2 \leftarrow d_{\text{shift}}$ {Fill in values of DT}
for $i = 0$ **to** $i < d_{\text{len}} - 1$ **do**
  while $z[k+1] < q_2$ **do**
    $k \leftarrow k + 1$
  end while
  $\mathcal{D}_f(q) \leftarrow a(q_2 - v[k])^2 + b(q_2 - v[k])^2 + f(v[k])$
  $q_2 \leftarrow q_2 + 1$
end for

Algorithm 1 shows the 1D-DT in pseudocode. The algorithm has two stages. In the first stage, it computes the lower envelope of the parabolas. In the second stage, it

uses this lower envelope to fill in the values of $\mathcal{D}_f$ in the grid. The main part of the algorithm is the lower envelope computation. Note, that the parabolas defining DT are sorted because they are rooted at the grid positions from $\mathcal{S}$ and they intersect at exactly one point. The intersection s of two parabolas coming from the grid positions q and r can be computed analytically as

Intersection of parabolas.

$$s = \frac{(f(r) - f(q)) - b(r - q) + a(r^2 - q^2)}{2a(r - q)}\,, \tag{3.16}$$

where coefficients a, b are the weighting components from $\boldsymbol{z}$, i.e. either z_1, z_3 or z_2, z_4, depending for which coordinate we compute the DT.

We keep track of the lower envelope structure, by using two arrays. The horizontal grid location of the i-th parabola is stored in $v[i]$. The range, in which the i-th parabola is below the others, is given by $z[i]$, and $z[i+1]$. Variable k represents the number of parabolas in the lower envelope.

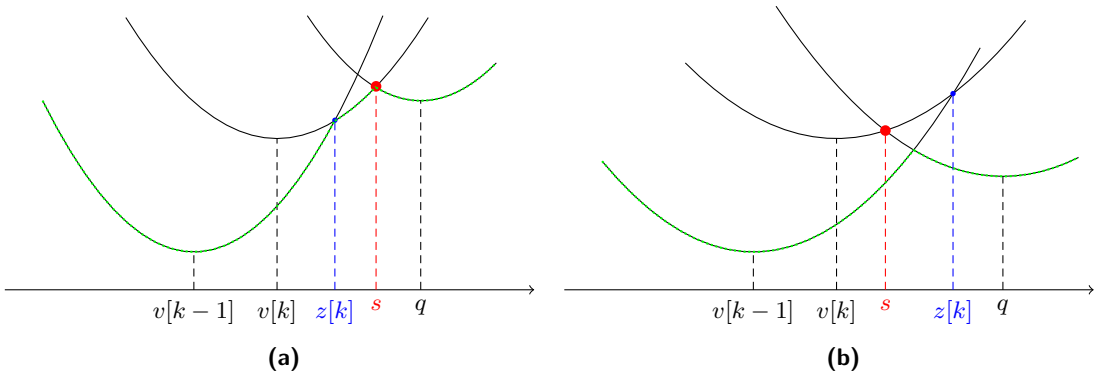


Figure 3.10. The two possible cases for the intersection point s, when adding the parabola from q to the lower envelope constructed so far. In 3.10a, we have $s > z[k]$, the parabola at q is added to the lower envelope, which consists of parabolas rooted at $v[k-1], v[k]$. In 3.10b, we have $s \leq z[k]$ and see, that parabola at $v[k]$ is not a part of the lower envelope and, therefore, can be removed from the list. Green dots emphasize the lower envelope.

There are just two cases that may occur, when adding a new parabola rooted at grid location q and the rightmost parabola in the lower envelope computed so far $v[k]$. See Figure 3.10 for the illustration. In the first case, Fig 3.10a, $s > z[k]$, new parabola is simply added to the envelope. In the second case, Fig 3.10b, $s \leq z[k]$, the addition of new parabola results in removing a parabola that used to be a part of the lower envelope.

Once the lower envelope is computed, DT values can be filled-in by sampling the height of the lower envelope at each grid location. Here, offset d_{shift}, and grid size, i.e. the search space, d_{len} come into place, to express the possibility of a different sized $\mathcal{S}_i, \mathcal{S}_j$ search spaces.

The overall time complexity of the inference calculation is $\mathcal{O}(|V||\mathcal{S}'|)$, since Algorithm 1 operates in $\mathcal{O}(|\mathcal{S}'|)$ time, and it needs to be called exactly $|V|$ times.

Algorithm 1 assumes, that functions (3.14) are convex. Therefore, the inference computation requires the concavity of $g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z})$, to work correctly, because we are dealing with maximization. Let us construct the Hessian matrix of (3.14):

$$g''(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z}) = \begin{pmatrix} 2z_3 & 0 \\ 0 & 2z_4 \end{pmatrix}\,. \tag{3.17}$$

Note, that g'' is a symmetric matrix. Its principal minors are $D_1 = 2z_3$ and $D_2 = 4z_3z_4$. In order to get the negative definite matrix, and therefore the concave function

$g(\boldsymbol{s}_i, \boldsymbol{s}_j;\ \boldsymbol{z})$, we need $(-1)^k D_k > 0$ to hold for all principal minors. That gives us the following conditions:

$$\begin{aligned} -2z_3 &> 0 \\ 4z_3 z_4 &> 0\,, \end{aligned} \tag{3.18}$$

from the second condition, we immediately see, that both $z_3 < 0$, and $z_4 < 0$. Therefore, the only requirement to use DT are linear constraints on $\boldsymbol{z}$. In particular, we need to keep the 3rd and the 4th components of all vectors $\boldsymbol{w}_{ij}^{\phi}$, $(i,j) \in E^{\phi}$ negative. We denote the corresponding set of indices of the 3rd and the 4th components of $\boldsymbol{w}_{ij}^{\phi}$, $(i,j) \in E$ within the joint parameter vector $\boldsymbol{w}$ by symbol J^{-} for a later usage.

3.6. Coarse-to-fine Strategy to Speed Up DPM detector

A practical limitation of DPM detectors is their computational cost, scaling with the size of search spaces of individual landmark positions, $\mathcal{S}_i$, $i \in V$. The size of search space is a function of the resolution of the normalized frame and *a priori* knowledge of the landmark's position. The *a priori* landmark position depends on the accuracy and the robustness of the used face detector. That is, an imprecise localization provided by face detector has to be compensated by a large-sized search space, not to miss the correct landmark position. The search is done in the normalized frame, and the found landmark location is projected back to the original image. Therefore, resolution of the normalized frame *de facto* lower bounds the accuracy of the landmark localization. In turn, improving the localization accuracy increases the search time.

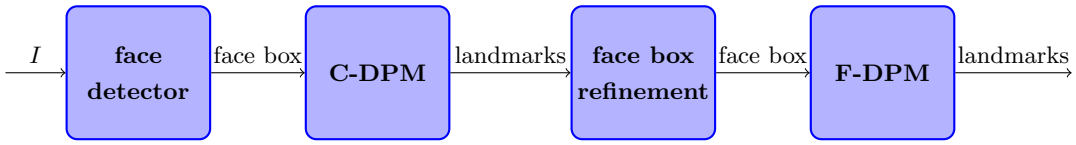


Figure 3.11. The visualization of the C2F-DPM strategy used to improve localization accuracy and to keep the processing time of DPM based detector low. The C-DPM detector operates on a low-resolution image which is localized by a face detector. Resulting rough estimate of landmarks helps to obtain a corrected face localization. The corrected face box allows computing narrow search spaces of the F-DPM operating on higher resolution images.

To alleviate the problem, we propose a coarse-to-fine strategy, also denoted as C2F-DPM, with two stages. In the first stage, we use a DPM detector, denoted as C-DPM, which operates in a lower resolution normalized frame. The output of the C-DPM detector is used to compute a better estimate of face location than the one provided by the face detector itself. Hence, the C-DPM detector serves as a precise face detector. In the second stage, we apply another DPM detector, denoted as F-DPM, which searches for landmarks in a higher resolution normalized frame. The initial estimate by C-DPM allows setting much tighter search spaces in the high-resolution normalized frame of the F-DPM detector without a danger of overlooking the landmarks. The scheme of C2F-DPM is outlined in Figure 3.11.

The precise face box used to initialize F-DPM is constructed from the response of C-DPM detector as follows. Its center is computed as the mean of estimated landmarks. Then, the centers of both eyes $\boldsymbol{c}_l$, $\boldsymbol{c}_r$ are calculated (again as the mean position of the corresponding estimated landmarks). Size of the precise face box is defined as $2.7 \cdot \|\boldsymbol{c}_l - \boldsymbol{c}_r\|_2$. Finally, the in-plane rotation of the precise face box is computed as a deviation of the least squares optimal line l, fitted to eyes landmarks, and the x-axis.

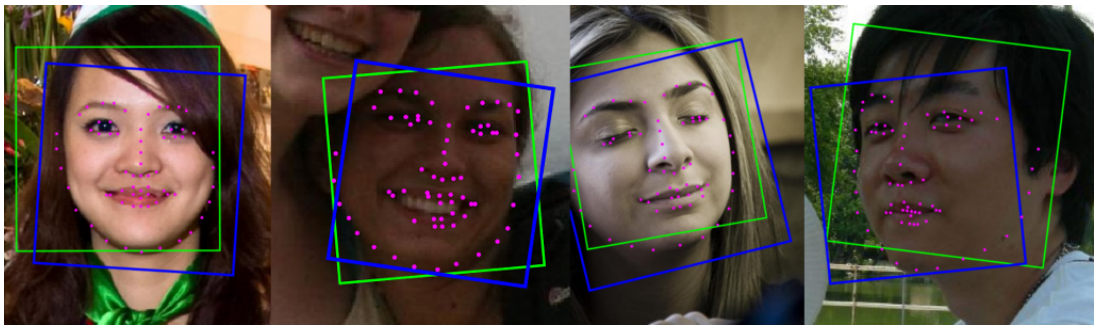
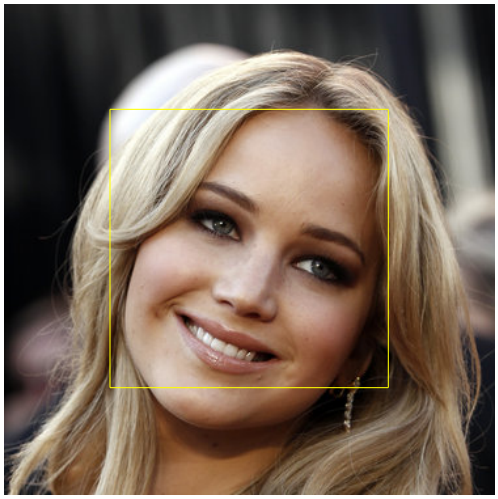


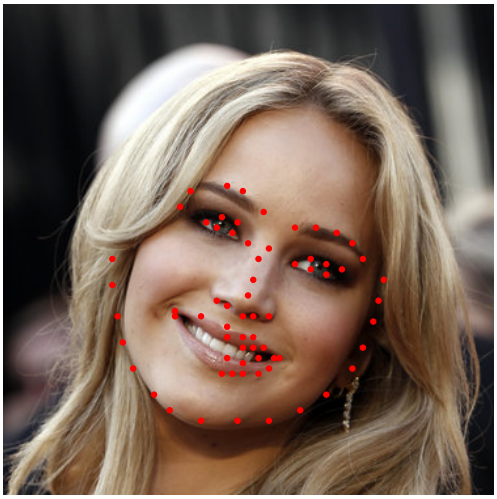
Figure 3.12. A few examples of the corrected face box computed from the response of C-DPM. The original face box detected by a face detector is green; the corrected face box is blue. Landmark positions predicted by C-DPM, and used for the face box correction are depicted in magenta.

The whole process of the precise face box computation is depicted in Figure 3.13. A few examples of the corrected face box are depicted in Figure 3.12.

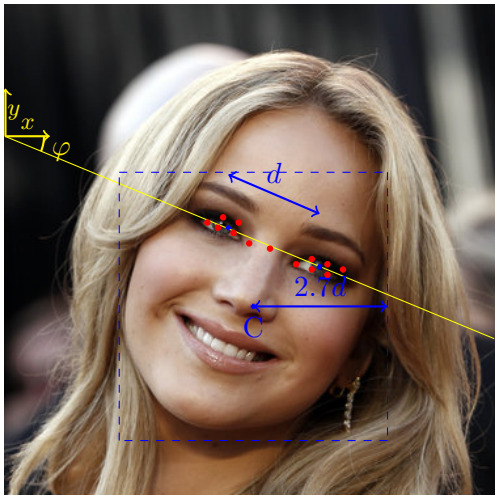
Specific implementation details and experimental evaluation are provided in Chapter 5. Here, we point out, that we use the same number of landmarks to be detected for both C-DPM, and F-DPM. It is evident, that this is suboptimal, regarding the processing time. C-DPM, and therefore also the resulting C2F-DPM, might be sped up significantly by using a sparser set of landmarks. However, that would require changes in the precise face box construction, because we are using the mean position of all landmarks for its center. We leave the possibility of this speedup as an open problem for a possible future work.



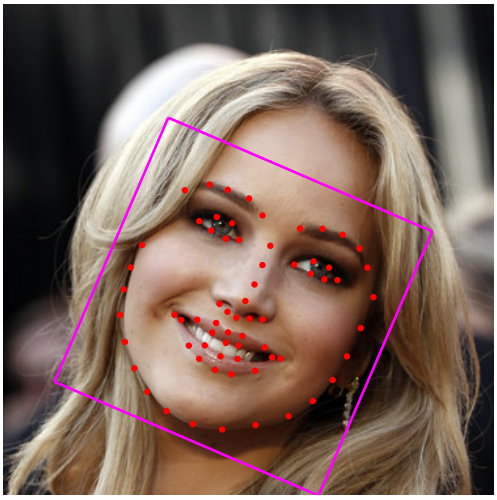
(a) Initial face box



(b) Landmarks detected by C-DPM



(c) Acquisition of corrected face box parameters



(d) Corrected face box and the landmarks detected by F-DPM

Figure 3.13. Face box correction. Eye landmarks detected by C-DPM are used to fit the best line in a least-squares manner (yellow line). Deviation of this line from x-axis defines the rotation angle φ. The corrected box is placed in the centroid of landmarks C, detected by C-DPM, and its size is set to 2.7× multiple of the length of line segment d connecting the centers of both eyes.

4. Learning

In this chapter, we describe methods which we use for learning the weights $\boldsymbol{w}$ of the detectors introduced in Chapter 3. We begin with defining the SO-SVM framework [Tsochantaridis et al., 2005] for learning a generic linear classifier. Then, in Section 4.1 and 4.3, we describe variants of SO-SVM suitable for learning parameters of the single-stage and two-stage landmark detectors both being instances of a generic linear classifier. The Stochastic Gradient Descent (SGD) and the BMRM algorithms, which are solvers commonly used in SO-SVM learning, are outlined in Section 4.4 and 4.5, respectively. Finally, we describe two proposed improvements of the BMRM algorithm. In particular, Section 4.6 is dedicated to the proposed Proximal BMRM algorithm, and Section 4.7 describes the BMRM with Multiple Cutting Plane model.

In this chapter we consider the following general definition of a prediction problem. Let $p\colon \mathcal{X} \times \mathcal{Y} \to [0, 1]$ be a probability density function defined over a set of inputs $\mathcal{X}$, elements of which $\boldsymbol{x} \in \mathcal{X}$ are arbitrary objects (e.g. images), and a set of hidden labels $\mathcal{Y}$, whose elements $\boldsymbol{y}$ are structured objects (e.g. configuration of landmark positions). For simplicity, we assume that both $\mathcal{X}$ and $\mathcal{Y}$ are finite sets [1]. Then, let $\Delta\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$ be a non-negative loss function, such that $\Delta(\boldsymbol{y}, \boldsymbol{y}) = 0,\ \forall \boldsymbol{y} \in \mathcal{Y}$ and $\Delta(\boldsymbol{y}, \boldsymbol{y}') > 0, \forall \boldsymbol{y} \neq \boldsymbol{y}'$. Let $\boldsymbol{\Psi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ be a fixed mapping from the input-output space onto the space of parameters. The goal is to find parameters (weights) $\boldsymbol{w}$ of a linear classifier

$$h(\boldsymbol{x}; \boldsymbol{w}) \in \operatorname*{Argmax}_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \tag{4.1}$$

linear classifier

which minimizes the expected risk

$$R_{\text{exp}}(\boldsymbol{w}) = \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{\boldsymbol{y} \in \mathcal{Y}} p(\boldsymbol{x}, \boldsymbol{y}) \Delta\left(\boldsymbol{y}, h(\boldsymbol{x}; \boldsymbol{w})\right) \ . \tag{4.2}$$

expected risk

The risk (4.2) cannot be minimized directly because $p(\boldsymbol{x}, \boldsymbol{y})$ is unknown. However, we are given a training set $\mathcal{T} = \{(\boldsymbol{x}^j, \boldsymbol{y}^j) \in \mathcal{X} \times \mathcal{Y} \mid j = 1, \ldots, m\}$ assumed to be drawn from i.i.d. random variables with the distribution $p(\boldsymbol{x}, \boldsymbol{y})$. In this case, we can approximate the expected risk $R_{\text{exp}}(\boldsymbol{w})$ by the empirical risk

$$R_{\text{emp}}(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} \Delta\left(\boldsymbol{y}^i, h(\boldsymbol{x}^i, \boldsymbol{w})\right) \ . \tag{4.3}$$

empirical risk

The empirical risk $R_{\text{emp}}(\boldsymbol{w})$ can be already minimized in principle. However, the optimization problem is for most choices of the loss function $\Delta(\boldsymbol{y}, \boldsymbol{y}')$ intractable. Hence, in practice, learning is formulated as a minimization of a convex regulairized risk

$$F(\boldsymbol{w}) = \Omega(\boldsymbol{w}) + R(\boldsymbol{w}) \,, \tag{4.4}$$

regularized risk

where $R\colon \mathbb{R}^n \to \mathbb{R}$ denotes a convex surrogate of the empirical risk $R_{\text{emp}}(\boldsymbol{w})$ and $\Omega\colon \mathbb{R}^n \to \mathbb{R}$ is convex regularization term used to prevent overfitting.

There are many learning algorithms which in their core minimize a regularized risk (4.4). The SO-SVM algorithm [Tsochantaridis et al., 2005] is one of the, translating the learning into the following convex problem

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w} \in \mathbb{R}^n} F(\boldsymbol{w}) \qquad \text{where} \qquad F(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} r_i(\boldsymbol{w}) \ . \tag{4.5}$$

unconstrained risk minimization

[1] Note that in our application $\mathcal{X}$ is a large yet finite set containing $256^{H \cdot W}$ gray-scale images.

The surrogate empirical risk $R(\boldsymbol{w})$ is expanded to $\frac{1}{m}\sum_{i=1}^{m} r_i(\boldsymbol{w})$, where $r_i(\boldsymbol{w})$ is a convex surrogate loss incurred by the classifier on the i-th training example $(\boldsymbol{x}^i, \boldsymbol{y}^i)$, and $\Omega(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2$ is a quadratic regularizer. The value of the regularization constant $\lambda > 0$ is tuned on a validation set. There are several options how to construct the surrogate loss $r_i(\boldsymbol{w})$. In our work, we use the so called margin-rescaling loss [Tsochantaridis et al., 2005] defined as

$$r_i(\boldsymbol{w}) = \max_{\boldsymbol{y}\in\mathcal{Y}} \left[\Delta(\boldsymbol{y}, \boldsymbol{y}') + \left\langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}) \right\rangle - \left\langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}^i) \right\rangle \right] . \tag{4.6}$$

It is easy to see, that $r_i(\boldsymbol{w})$ is a convex function of $\boldsymbol{w}$, and that it upper bounds the value of the true loss $\Delta(\boldsymbol{y}^i, h(\boldsymbol{x}^i; \boldsymbol{w}))$ for all $\boldsymbol{w} \in \mathbb{R}^n$. Evaluation of the proxy loss $r_i(\boldsymbol{w})$ is equivalent to evaluation of the linear classifier (4.1) whose score function is augmented by the true loss $\Delta(\boldsymbol{y}, \boldsymbol{y}')$. In the case of our landmark detectors, the evaluation of $r_i(\boldsymbol{w})$ can be done by the same DP algorithms which we use for classification, since the true loss $\Delta(\boldsymbol{y}, \boldsymbol{y}')$ decomposes over landmarks. The particular choices of the true loss $\Delta(\boldsymbol{y}, \boldsymbol{y}')$ is discussed later in Section 4.2.

The existing solvers of the optimization problem (4.5) can be categorized into *approximate online algorithms*, and *batch algorithms*. In general, the online algorithms converge quickly at the first stages of the optimization process, but they require a long time to achieve a precise solution. The batch methods behave the other way around, that is, their convergence is slow (and often fluctuates) at the beginning, but they are faster when approaching the optimum. The batch algorithms provide a certificate of optimality that can be used as a strict stopping condition. On the other hand, the online methods do not have clearly defined stopping condition. The online methods are suitable for learning from very large sets of training examples. The batch methods are preferable when a solution with a guaranteed precision is required. Which is the case when the number of training examples is relatively small compared to the number of parameters to be learned, and the regularization is thus necessary. In our case, the batch methods are preferable because our detector has a large number of parameters (hundreds of thousands) due to the usage of high-dimensional S-LBP features. The SGD algorithm representing the online methods is described in Section 4.4. The most prominent representative of the batch methods nowadays is the BMRM algorithm which is described in Section 4.7.

The batch as well as the online methods access the objective function of (4.5) via the first order oracle of the surrogate loss $r_i(\boldsymbol{w})$. For given $\boldsymbol{w}$, the first order oracle returns the value of $r_i(\boldsymbol{w})$ and its sub-gradient $\boldsymbol{r}_i'(\boldsymbol{w})$. The sub-gradient of $r_i(\boldsymbol{w})$ defined by (4.6) can be computed by Danskin's theorem (see e.g. [Bertsekas, 1999]) as follows

$$\boldsymbol{r}_i'(\boldsymbol{w}_t) = \boldsymbol{\Psi}(\boldsymbol{x}^i, \hat{\boldsymbol{y}}) - \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}^i) , \tag{4.7}$$

where

$$\hat{\boldsymbol{y}} \in \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \left[\Delta(\boldsymbol{y}, \boldsymbol{y}^i) + \left\langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}) \right\rangle \right] . \tag{4.8}$$

Computation of the sub-gradient requires solving the same augmented classification problem (4.8) as is needed when evaluating $r_i(\boldsymbol{w})$ defined in formula (4.6).

4.1. Learning DPM landmark detector by SO-SVM

The single-view DPM detector (3.1) as well as the single-stage multi-view DPM detector (3.2) are special instances of the generic linear classifier (4.1). In particular, the input set $\mathcal{X}$ is a set of all normalized images $\mathcal{I}^{H\times W}$, and the hidden labels $\boldsymbol{y} \in \mathcal{Y}$ correspond to 2D landmark coordinates $\boldsymbol{s}$ for the singe-view detector, tupled with the

viewpoint $\phi \in \Phi$ in the case of the multi-view single-stage detector. That is, particular instances of the SO-SVM algorithm for landmark detectors are obtained after setting $\boldsymbol{x} \equiv I$, $\boldsymbol{y} \equiv \boldsymbol{s}$, or $\boldsymbol{y} \equiv (\phi, \boldsymbol{s})$, respectively.

It can be also see, that the score functions $f(I, \boldsymbol{s}; \boldsymbol{w})$ and $f_\phi(I, \boldsymbol{s};\ \boldsymbol{w})$ of detectors (3.1) and (3.2), respectively, are linear in parameters $\boldsymbol{w}$. Recall, that we have defined both appearance model (3.4), and deformation cost (3.5) as linearly parametrized functions. For example, score function of the single-view detector (3.1) can be written as the dot product $f(I, \boldsymbol{s}; \boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{\Psi}(I, \boldsymbol{s}))\rangle$ where the joint parameter vector $\boldsymbol{w}$ is constructed as a concatenation of parameter vectors of the individual appearance models $\boldsymbol{w}_i$, $i \in V$ and parameter vectors of all deformation costs $\boldsymbol{w}_{ij}$, $(i,j) \in E$. The joint feature map $\boldsymbol{\Psi}(I, \boldsymbol{s})$ is constructed by concatenating the feature maps $\boldsymbol{\Psi}_i(I, \boldsymbol{s}_i)$, $i \in V$, and $\boldsymbol{\Psi}_e(\boldsymbol{s}_i, \boldsymbol{s}_j)$, $(i,j) \in E$, in an appropriate order. Showing that the single-stage multi-view DPM detector has a linear score function is analogous.

To speed up the evaluation of landmark detectors by using DT, we need to enforce a subset of weights w_i, $i \in J^-$, of the parameter vector $\boldsymbol{w}$ to be strictly negative (c.f. Section 3.5.2). To this end, we augment the unconstrained SO-SVM problem (4.5) by linear constraints which yield the following convex program

$$\begin{aligned} \boldsymbol{w}^* \quad &= \quad \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} F(\boldsymbol{w})\ , \qquad \text{where} \qquad F(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} r_i(\boldsymbol{w}) \\ &\text{s.t.} \quad w_i \le c^-, \quad i \in J^-\ , \end{aligned} \tag{4.9}$$

constrained risk minimization

and c^- is set to a small negative constant.

4.2. Loss function

A substantial advantage of empirical risk minimization based learning, of which SO-SVM is a special instance, is the option to directly optimize the classifier's performance regarding a user-defined loss function $\Delta(\boldsymbol{y}, \boldsymbol{y}')$. The loss function measures the discrepancy between the ground-truth annotation of a training example and the output predicted by a classifier. In this section, we define loss functions suitable for learning of landmark detectors. The loss functions are tightly connected to the evaluation metrics which will be used to evaluate the landmark detectors empirically in Chapter 5.

Single-view loss

In the case of single-view landmark detector (3.1), we design the loss function to measure the average Euclidean distance between the ground-truth landmark positions $\boldsymbol{s}$ and the predicted landmark positions $\boldsymbol{s}'$ (also denoted as a point-to-point (P2P) error). In particular, the loss function is defined as

$$\Delta_{\text{loc}}(\boldsymbol{s}, \boldsymbol{s}') = \frac{1}{\kappa(\boldsymbol{s})|V|}\sum_{j\in V} \|\boldsymbol{s}_j - \boldsymbol{s}'_j\|\ , \tag{4.10}$$

single-view loss

where $\kappa(\boldsymbol{s})$ is the face size computed from the ground truth annotation $\boldsymbol{s}$. The term $\frac{1}{\kappa(\boldsymbol{s})}$ serves as a normalization factor which makes the measurement of the localization error invariant to a different size of the input face. Recall, that the input image is size normalized based on a response of the face detector which is, usually, not very precise. In our experiments, we consider two ways how to define the face size $\kappa(\boldsymbol{s})$. The first options is to define the face size horizontally, i.e. $\kappa(\boldsymbol{s})$ is set to be an IOD of the face, see Figure 4.1a. The IOD has become *de facto* the standard normalization factor used in the literature. Hence, we consider this option to make our results comparable with

(a) (b)

Figure 4.1. Face size $\kappa(\boldsymbol{s})$ definitions. The Fig. 4.1a shows the horizontal definition, which is the IOD. The Fig. 4.1b shows the vertical definition, which is the size of the line connecting the root of the nose and the cheek, i.e. the face height (FH). Note, that the vertical definition is the same for both frontal and profile view, while the horizontal would be 0 for the profile view. Images are taken from the Multi-PIE database [Gross et al., 2010].

face size

the work of others. However, we prefer a different definition, which is to compute the face size vertically, see Figure 4.1b. The vertical definition, measuring face as a distance between the root of the nose and the chin, is not affected by the yaw angle of the head which varies considerably in the multi-view setting. We refer to this vertical definition of the face size also as a face height (FH) in the following text. It is easy to see that the horizontal definition of the face size (IOD) approaches zero 0 for the full profile views pushing the error to the infinity. In contrast, the vertical definition (FH) dependents on the pitch head rotation which has much smaller variation in the consumer images.

Multi-view loss

In the case of the multi-view DPM detector (3.2), the loss function measures a discrepancy between the ground-truth landmark positions $\boldsymbol{s}$, the viewpoint ϕ, and their predictions $\boldsymbol{s}'$, and ϕ'. The multi-view loss is defined as

multi-view loss

$$\Delta_{\text{mv}}(\phi, \boldsymbol{s}, \phi', \boldsymbol{s}') = \begin{cases} \frac{1}{\kappa(\boldsymbol{s})|V^{\phi}|} \sum_{j \in V^{\phi}} \|\boldsymbol{s}_j - \boldsymbol{s}'_j\|, & \text{if } \phi = \phi', \\ 1, & \text{otherwise.} \end{cases} \tag{4.11}$$

In the multi-view setting, we use only the vertical definition of the normalization constant $\kappa(\boldsymbol{s})$ for the reasons described above. Provided the viewpoint is predicted correctly, $\phi = \phi'$, the value of the multi-view loss equals to the same localization error as defined for the single-view case. In the case of the viewpoint misclassification, $\phi \neq \phi'$, the penalty is set to 1 regardless the ground-truth positions $\boldsymbol{s}$ and predicted positions $\boldsymbol{s}'$ which need not be comparable due to a different number of landmarks. The penalty 1 is much larger value than an acceptable localization error. Hence, the loss function discourages mistakes in the viewpoint predictions more than landmark misplacement.

0/1 Single-view loss

To have a fair comparison with the detector of Zhu and Ramanan [2012], we define 0/1 loss which best corresponds to the learning problem (2.29) proposed in their paper.

The 0/1-loss is defined as

$$\Delta_{0/1}(\boldsymbol{s}, \boldsymbol{s}') = \begin{cases} 0, & \text{if } \boldsymbol{s} = \boldsymbol{s}' , \\ 1, & \text{otherwise} , \end{cases} \tag{4.12}$$

0/1 single-view loss

which means that the loss is 0 if all landmark coordinates are predicted precisely. Otherwise the loss is 1 (even if prediction of a single coordinate differs from the ground truth by only one pixel).

4.3. Two-stage Multi-view Detector Learning

In this section, we describe the learning of the two-stage multi-view landmark detector proposed in Section 3.2.2. The two-stage detector (3.2.2) is composed of $|\Phi| + 1$ predictions rules which are all instances of a generic linear classifier (4.1). Parameters of all the linear rules are learned separately by the SO-SVM algorithm. Learning of the parameters is split into two stages:

1. In the first stage, we learn a single-view landmark detector (3.3) for each viewpoint $\phi \in \Phi$ separately. We use the same instance of the SO-SVM algorithm as for the "frontal" single-view detector whose learning was described in Section 4.1. In particular, for each viewpoint $\phi \in \Phi$ we learn parameters $\boldsymbol{w}_1^\phi$ by solving the SO-SVM problem (4.9) using the multi-view loss (4.11). The training set contains a subset of examples for the particular viewpoint only.

2. In the second stage, we learn a single multi-class linear classifier (3.3) predicting the viewpoint ϕ based on the features computed from the responses of $|\Phi|$ single-view landmark detectors from the first stage. Parameters $\boldsymbol{w}_2 = (\boldsymbol{w}_2^\phi \mid \phi \in \Phi)$ are learned by solving the SO-SVM problem (4.5) using the 0/1-loss $\Delta(\phi, \phi') = [\![\phi \neq \phi']\!]$. In this case, the entire training set is used.

Table 4.1. Comparison of learning procedures of the single-stage and the two-stage multi-view detectors. Learning is compared regarding the number of parameters, the number of possible outputs $|\mathcal{Y}|$ of the learned classifiers, and the optimized loss functions. The single-stage detector requires solving a single large convex problem while the two-stage detector requires solving $|\Phi| + 1$ smaller problems.

	The number of parameters		The number of outputs		Loss Function	
single stage	$\sum_{\phi\in\Phi} \left[\sum_{i\in V^\phi} n_i^\phi + \sum_{ij\in E^\phi} n_{ij}^\phi \right]$		$\prod_{\phi\in\Phi} \left[\prod_{i\in V^\phi} \lvert\mathcal{S}_i^\phi\rvert \right]$		Δ_{mv}	
two stage	1. stage	2. stage	1. stage	2. stage	1. stage	2. stage
	$\sum_{i\in V^\phi} n_i^\phi + \sum_{ij\in E^\phi} n_{ij}^\phi, \ \forall\phi\in\Phi$	$\sum_{\phi\in\Phi}\sum_{i\in V^\phi} n_i^\phi$	$\prod_{i\in V^\phi} \lvert\mathcal{S}_i^\phi\rvert, \ \forall\phi\in\Phi$	$\lvert\Phi\rvert$	Δ_{loc}	$\Delta_{0/1}$

Table 4.1 compares learning of the single-stage and the two-stage multi-view detectors regarding the number of parameters, the number of possible outputs $|\mathcal{Y}|$ of the learned classifiers, and the optimized loss functions. The number of parameters and the number of classifier outputs influences the complexity of the convex problems to be solved. Note that the number of classifier outputs corresponds effectively to the number of constraints which are implicitly involved in the definition of the margin-rescaling loss function. It is seen that the single-stage approach requires solving a single but enormous instance of the convex problem (4.9). Although the two-stage approach requires solving $|\Phi|$ instances of the problem (4.9) and a single instance of the problem (4.5), these convex problems are considerably smaller. In turn, learning of the two-stage detector is substantially faster as will be shown empirically in Chapter 5.

4.4. Stochastic Gradient Descent

Algorithm 2 Projected SGD with constant step-size and averaging

Require: $\lambda > 0$, $\alpha > 0$, $(l_i \leq u_i)$, $i = 1, \ldots, n$
1: set $\boldsymbol{w}_0 := \mathbf{0}$, $\boldsymbol{v}_0 = \mathbf{0}$, $t := 0$
2: **repeat**
3: **for** i in randperm(m) **do**
4: compute sub-gradient $\boldsymbol{r}'_i(\boldsymbol{w}_t)$ of i-th example at $\boldsymbol{w}_t$
5: $\boldsymbol{g}_t = \frac{\lambda}{m}\boldsymbol{w}_t + \frac{1}{m}\boldsymbol{r}'_i(\boldsymbol{w}_t)$
6: $\boldsymbol{w}_{t+1} = P(\boldsymbol{w}_t - \alpha \boldsymbol{g}_t)$
7: $\boldsymbol{v}_{t+1} = \frac{t-1}{t}\boldsymbol{v}_t + \frac{1}{t}\boldsymbol{w}_{t+1}$
8: **end for**
9: **until** convergence

A prominent representative of the online methods is the SGD algorithm. There are many variants of SGD. In order to deal with the linear constraints $w_i \leq c^-$ in the problem (4.9), we use the Projected Stochastic Gradient Descent (P-SGD) with constant step size and averaging [Shamir and Zhang, 2013], the pseudo-code of which is outlined in Algorithm 2.

In each epoch (step 2), the P-SGD algorithm goes through the training examples in a randomly generated order (step 3). This is a common strategy which simulates random sampling of the data without overlooking some examples from the training set. A randomly picked example is used to compute a stochastic estimate of the sub-gradient $\boldsymbol{g}_t$ of the objective function (step 4 and step 5). The parameters $\boldsymbol{w}_t$ are updated by subtracting α fraction of the gradient, and the result is projected to the space of feasible solutions by orthogonal projector $P\colon \mathbb{R}^n \to \mathbb{R}^n$ (step 6). For given $\boldsymbol{w} \in \mathbb{R}^n$, the projector $\boldsymbol{w}' = P(\boldsymbol{w})$ is the closest point satisfying the constraints $w'_i \leq c^-$, $i \in J^-$, which can be written in a closed form as

negativity constraints

$$P(\boldsymbol{w}) \;=\; \begin{bmatrix} P_1(w_1) \\ \vdots \\ P_n(w_n) \end{bmatrix} \quad \text{where} \quad P_i(w_i) = \begin{cases} \min(c^-, w_i) & \text{if} \quad i \in J^-\,, \\ w_i & \text{if} \quad i \notin J^-\,. \end{cases} \tag{4.13}$$

The scalar $\alpha > 0$ is a constant step-size (learning rate). We tune α based on the value of the objective function $F(\boldsymbol{w}_t)$ obtained after ten epochs of the algorithm run on a subset created from 10% of training examples. To speed up the convergence, we use the averaging scheme where a new "averaged" iterate $\boldsymbol{v}_{t+1}$ is defined as the average of all previous SGD iterates $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_t$ (step 7).

The main benefit of P-SGD is its simplicity. Another advantage is a fast convergence at the early stages of the optimization, that is, it gets relatively close to the optimum very soon. However, then it gets stalled and obtaining sufficiently precise solution takes prohibitively long time. We found experimentally that in our task a precise solution of the learning problem is essential to get a detector with high performance. Another issue that prolongs the training time is the need to tune the learning rate α appropriately for each run of the algorithm. Another disadvantage is the lack of a strict stopping criterion which is typically resolved by monitoring the progress of the validation error and stopping the algorithm when there is no significant improvement. Finally, the basic variant of the P-SGD is online in nature, and it cannot be easily parallelized. Due to these disadvantages, we started to use a batch optimization algorithms described in the next section.

4.5. Batch optimization algorithms

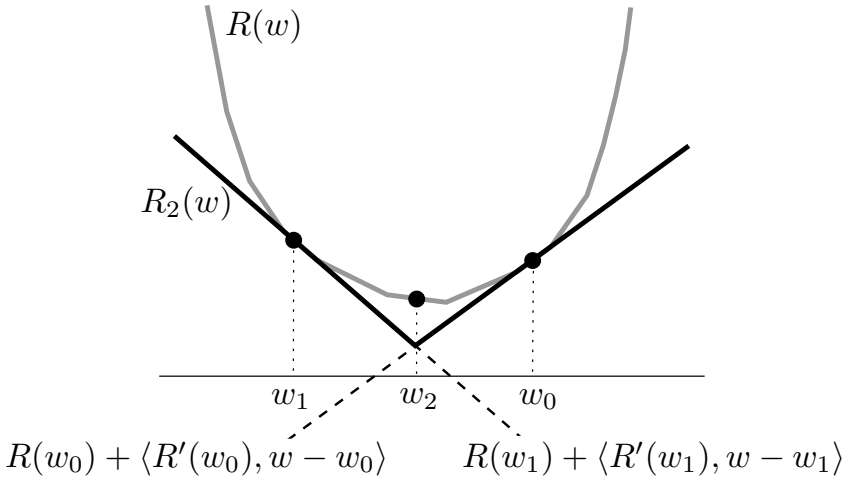


Figure 4.2. A convex function $R(\boldsymbol{w})$ (gray) is approximated by a point-wise maximum $R_2(\boldsymbol{w})$ (black) of two linear under-estimators of $R(\boldsymbol{w})$, the so called cutting planes, which are computed at points $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$. It is seen that the minimum of $R(\boldsymbol{w})$ can be well approximated by the minimum of much the simpler function $R_2(\boldsymbol{w})$.

In this section, we describe batch optimization algorithms suitable for solving convex minimization problems of the form

$$\boldsymbol{w}^* \in \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^n} F(\boldsymbol{w}) \quad \text{where} \quad F(\boldsymbol{w}) = \Omega(\boldsymbol{w}) + R(\boldsymbol{w}) \,, \tag{4.14}$$

$R\colon \mathbb{R}^n \to \mathbb{R}$ is an arbitrary convex function and $\Omega\colon \mathbb{R}^n \to \mathbb{R}$ is a convex regularization term. The function $R(\boldsymbol{w})$ is assumed to be complex and expensive to evaluate while the regularization term $\Omega(\boldsymbol{w})$ is assumed to be simple and easy to evaluate. In our application, $R(\boldsymbol{w})$ is a surrogate of the empirical risk and the regularization term is quadratic, $\Omega(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2$, where $\lambda > 0$ is a regularization constant. Note, that quadratic regularization term guarantees that problem (4.14) has a unique solution.

4.5.1. Cutting Plane Algorithm

Let us assume for a moment a special variant of problem (4.14) without the regularization term $\Omega(\boldsymbol{w})$, that is,

$$\boldsymbol{w}^* \in \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^n} R(\boldsymbol{w}) \,. \tag{4.15}$$

Thanks to the convexity of $R(\boldsymbol{w})$ it can be approximated by the Cutting Plane (CP) Model (CPM)

$$R_t(\boldsymbol{w}) = \max_{i=1,\dots,t} \left[R(\boldsymbol{w}_i) + \left\langle \boldsymbol{R}'(\boldsymbol{w}_i), \boldsymbol{w} - \boldsymbol{w}_i \right\rangle \right] \,, \tag{4.16}$$

CPM

where $\boldsymbol{w}_1, \dots, \boldsymbol{w}_t$ are points at which risk $R(\boldsymbol{w})$ is sampled, and $\boldsymbol{R}'(\boldsymbol{w}_i) \in \mathbb{R}^n$, $i = 1, \dots, t$, denote sub-gradients computed at these points. By definition, the CPM $R_t(\boldsymbol{w})$ is a piece-wise linear under-estimator of risk $R(\boldsymbol{w})$ which is tight at points $\boldsymbol{w}_1, \dots, \boldsymbol{w}_t$. Figure 4.2 illustrates the approximation on a simple function.

The CPA [Cheney and Goldstain, 1959] is a simple iterative procedure exploiting the CPM to solve the problem (4.15). Starting from an initial solution $\boldsymbol{w}_1 \in \mathbb{R}^n$, the CPA computes new iterates by solving the so-called *reduced problem*:

$$\boldsymbol{w}_{t+1} \in \arg\min_{\boldsymbol{w} \in \mathbb{R}^n} R_t(\boldsymbol{w}) \,, \tag{4.17}$$

reduced problem

where risk function $R(\boldsymbol{w})$ is replaced by its CPM $R_t(\boldsymbol{w})$. It is well-known that the iterates generated by the CPA show a strong "zig-zag" behavior, especially at early iterations when the CPM is still quite inaccurate, resulting in a slow convergence. See experiments presented in Section 5.6, for the illustration.

4.5.2. Bundle Methods

Bundle Methods (BM) [Lemaréchal, 1978] refines the CPA by adding a quadratic prox-term to the reduced problem, i.e. the next iterate becomes

$$\boldsymbol{w}_{t+1} \in \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} \left[R_t(\boldsymbol{w}) + \rho_t \|\boldsymbol{w} - \boldsymbol{w}_t^+\|^2\right] , \tag{4.18}$$

where $\boldsymbol{w}_t^+$ is the prox-center and $\rho_t \in \mathbb{R}$ is the prox-term penalty. When the improvement in the objective value is sufficiently large, i.e. $R(\boldsymbol{w}_t) - R(\boldsymbol{w}_{t+1}) \leq \gamma_t,\ \gamma_t \in \mathbb{R}^+$ holds, the prox-center is updated to $\boldsymbol{w}_{t+1}^+ = \boldsymbol{w}_{t+1}$. Otherwise, the prox-center remains unchanged, i.e. $\boldsymbol{w}_{t+1}^+ = \boldsymbol{w}_t^+$. The prox-term reduces the influence of the inaccurate CPM by constraining the distance between the iterations, thereby removing the detrimental behavior of the CPA. The BM is controlled by two rules with a significant impact on the convergence [Lemaréchal et al., 1995]. The first rule defines the minimal decrease threshold γ_t, and the second rule sets the prox-term penalty ρ_t.

4.5.3. Bundle Methods for Regularized Risk Minimization

Teo et al. [2010] adopted BM for a specific problem (4.15). In particular, they propose to replace problem (4.5) by a following *reduced problem*

regularized reduced problem

$$\boldsymbol{w}_{t+1} \in \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^n} F_t(\boldsymbol{w}) \quad \text{where} \quad F_t(\boldsymbol{w}) = \Omega(\boldsymbol{w}) + R_t(\boldsymbol{w}) . \tag{4.19}$$

Regularization term $\Omega(\boldsymbol{w})$ serves as a natural prox-center and the reduced problem objective $F_t(\boldsymbol{w})$ is obtained just by replacing the risk term with its CPM. This is an elegant solution which avoids designing rules for updating the prox-center penalty, and the sufficient decrease threshold as needed by the original BM.

Algorithm 3 BMRM algorithm

Require: ϵ, first order oracle evaluating $R(\boldsymbol{w})$ and $\boldsymbol{R}'(\boldsymbol{w})$
1: Initialization: $\boldsymbol{w} \leftarrow \boldsymbol{0}, t \leftarrow 0$
2: **repeat**
3: $\quad t \leftarrow t + 1$
4: $\quad$ Call oracle to compute $R(\boldsymbol{w}_t)$ and $\boldsymbol{R}'(\boldsymbol{w}_t)$
5: $\quad$ Update the cutting plane model $R_t(\boldsymbol{w}_t)$
6: $\quad$ Solve the reduced problem (4.19)
7: **until** $F(\boldsymbol{w}_t) - F_t(\boldsymbol{w}_t) \leq \epsilon$

The BMRM is outlined in Algorithm 3. Starting from an initial guess $\boldsymbol{w}_1 \in \mathbb{R}^n$, BMRM iteratively solves the reduced problem (4.19), and use new iterate $\boldsymbol{w}_{t+1}$ to update the CPM (4.16), which becomes progressively more and more accurate. This process is repeated until a gap between primal and reduced objective gets below a prescribed $\epsilon > 0$. It is easy to see, that the inequality in the stopping condition, $F(\boldsymbol{w}_t) - F_t(\boldsymbol{w}_t) \leq \epsilon$, implies that $F(\boldsymbol{w}_t) \leq F(\boldsymbol{w}^*) + \varepsilon$ holds. Therefore, in contrast to the online methods, BMRM provide a certificate of the optimality. Also, the first order oracle called at step 4, which is the main bottleneck of the algorithm, can be efficiently parallelized. However, besides all mentioned advantages, the BMRM can still suffer

from a slow convergence especially when the value of the regularization constant λ is small as we discuss in Section 4.5.4.

Finally, we describe an efficient solution of the reduced problem (4.19). In our application, we use a quadratic regularization term $\Omega(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2$ the problem (4.19) is therefore equivalent to a convex quadratic program. Let us define the following shortcuts $\boldsymbol{a}_i = \boldsymbol{R}'(\boldsymbol{w}_i)$ and $b_i = R(\boldsymbol{w}_i) - \langle \boldsymbol{R}'(\boldsymbol{w}_i), \boldsymbol{w}_i \rangle$. Then, we can rewrite the reduced problem (4.19) as

$$\boldsymbol{w}_t \in \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^n} \left[\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \xi\right] \quad \text{s.t.} \quad \langle \boldsymbol{a}_i, \boldsymbol{w}\rangle + b_i \leq \xi, \quad i = 1, \ldots, t\,. \tag{4.20}$$

The problem (4.20) has $n+1$ variables, where n is a dimension of parameters which is hundreds of thousands in our application. It is thus advantageous to solve (4.20) in its dual form having only t variables as shown next. The Lagrangian of (4.20) reads

$$L(\boldsymbol{w}, \xi, \boldsymbol{\alpha}) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \xi - \sum_{i=1}^{t} \alpha_i \left[\xi - b_i - \langle \boldsymbol{a}_i, \boldsymbol{w}\rangle\right]\,. \tag{4.21}$$

The objective of the dual problem is the value of $\min_{\boldsymbol{w},\xi} L(\boldsymbol{w}, \xi, \boldsymbol{\alpha})$ which has a closed form solution derived from equations defining the stationary point:

$$\frac{\partial L(\boldsymbol{w}, \xi, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = \lambda \boldsymbol{w} \sum_{i=1}^{t} \alpha_i \boldsymbol{a}_i = 0 \Rightarrow \boldsymbol{w} = -\frac{1}{\lambda}\sum_{i=1}^{t} \alpha_i \boldsymbol{a}_i\,, \tag{4.22}$$

$$\frac{\partial L(\boldsymbol{w}, \xi, \boldsymbol{\alpha})}{\partial \xi} = 1 - \sum_{i=1}^{t} \alpha_i = 0 \Rightarrow \sum_{i=1}^{t} \alpha_i = 1\,, \tag{4.23}$$

$$\frac{\partial L(\boldsymbol{w}, \xi, \boldsymbol{\alpha})}{\partial \alpha_i} = b_i + \langle \boldsymbol{a}_i, \boldsymbol{w}\rangle - \xi = 0 \Rightarrow \xi = b_i + \langle \boldsymbol{a}_i, \boldsymbol{w}\rangle\,. \tag{4.24}$$

By plugging (4.22) and (4.23) back to (4.21), primal variables ξ, and $\boldsymbol{w}$ are eliminated and we get a following formulation of the dual task

$$\alpha_t \in \arg\max_{\boldsymbol{\alpha}\in\mathbb{R}^t} \left[\sum_{i=1}^{t} \alpha_i b_i - \frac{1}{2\lambda}\left\|\sum_{i=1}^{t} \alpha_i \boldsymbol{a}_i\right\|^2\right] \quad \text{s.t.} \quad \sum_{i=1}^{t} \alpha_i = 1\,, \quad \alpha_i \geq 0\,, i = 1, \ldots, t\,. \tag{4.25}$$

Equation (4.22) connects the primal, and the dual optimal solution. In particular, the primal solution $\boldsymbol{w}_t$ can be obtained from the dual optimal solution $\boldsymbol{\alpha}_t$ by

$$\boldsymbol{w}_t = -\frac{1}{\lambda}\sum_{i=1}^{t} \alpha_i \boldsymbol{a}_i\,. \tag{4.26}$$

4.5.4. Disadvantages of the BMRM algorithm

The prox-term penalty is in the BMRM replaced by a fixed regularization parameter λ and the prox-center is constantly zero. This is an elegant solution which simplifies the original BM. However, it comes at the cost of making the algorithm less efficient in some settings. In particular, for small values of λ, the influence of the regularizer is weak, and BMRM become closer to the CPA. Consequently, BMRM exhibits a "zig-zag" behavior leading to a slow convergence. See the convergence curves of experiments presented in Section 5.6 for illustration. The detrimental effect of a small λ is also seen from the upper bound on the maximal number of iterations, which is $\mathcal{O}(\log_2 \lambda + \frac{C}{\lambda\epsilon})$ (e.g. [Teo et al., 2010]). The mentioned inefficiency of BMRM can have serious

practical implications, because the optimal value of λ is unknown and thus needs to be discovered in the model selection stage. The model selection involves solving the optimization problem (4.14) with a range of λ's including small values, which might require prohibitively many iterations and long computational time in return. In the following section, we propose two improvements of the existing BMRM algorithm which significantly speed up its convergence as we show in Chapter 5 empirically.

4.6. Proximal Point BMRM

In this section, we describe our first improvement of the BMRM algorithm. We propose to integrate a quadratic prox-term to the objective of the reduced problem (4.19) to prevent the "zig-zag" behavior of the BMRM. This modification, which we call Prox-BMRM, returns the BMRM algorithm closer to its roots (i.e. BM [Lemaréchal, 1978]). There are two main differences between the proposed Prox-BMRM and the original BM:

1. We do not approximate the innate quadratic regularizer by the CPM.
2. We propose a new adaptive strategy for adjusting the prox-term penalty and the minimal improvement threshold.

The reason for introducing an additional prox-term is to prevent the overly large changes of the solution in two consecutive iterations. To this end, we require that Euclidean distance between two successive iterations $\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\|$ is not larger than some reasonably chosen constant $K > 0$. This constraint is implemented by adding a prox-term to the objective function of the reduced problem, i.e. the modified objective of the reduced problem becomes

$$F_t(\boldsymbol{w}, \rho_t) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + R_t(\boldsymbol{w}) + \rho_t\|\boldsymbol{w} - \boldsymbol{w}_t\|^2 , \tag{4.27}$$

where $\rho_t \geq 0$ is the prox-term penalty. Similarly to the original BMRM, the Prox-BMRM computes a new iterate by minimizing the reduced objective (4.27). The value of the prox-term penalty ρ_t is set adaptively as described in Algorithm 4.

Algorithm 4 Prox-BMRM

Require: $\varepsilon > 0$, $T > 0$, $K > 0$, $\boldsymbol{w}_1 \in \mathbb{R}^n$

1: Set $\rho_1 = 0$ and $\gamma_t = \infty$
2: **repeat**
3: Solve the reduced problem $\boldsymbol{w}_{t+1}^{\rho_t} \in \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} F_t(\boldsymbol{w}, \rho_t)$
4: **if** $F(\boldsymbol{w}_t) - F(\boldsymbol{w}_{t+1}^{\rho_t}) \geq \gamma_t$ **then**
5: accept the solution and set: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_{t+1}^{\rho_t}, \quad \rho_{t+1} = \rho_t, \quad \gamma_{t+1} = \gamma_t$
6: **else**
7: Find the minimal $\hat{\rho} \in \{0, 1, 2, 4, \ldots\}$, such that $\|\boldsymbol{w}_{t+1}^{\hat{\rho}} - \boldsymbol{w}_t\| \leq K$, where $\boldsymbol{w}_{t+1}^{\rho} \in \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} F_t(\boldsymbol{w}, \rho_t)$
8: Set $\boldsymbol{w}_{t+1} = \boldsymbol{w}_{t+1}^{\hat{\rho}}$, $\rho_{t+1} = \hat{\rho}$ and
9: $\gamma_{t+1} = \frac{F(\boldsymbol{w}_{t+1}^{\hat{\rho}})}{T} - \frac{F_t(\boldsymbol{w}_{t+1}^{0})}{T(1-\varepsilon)}$
10: **end if**
11: **until** $F(\boldsymbol{w}_{t+1}) - F_t(\boldsymbol{w}_{t+1}^{0}) \leq \varepsilon \cdot |F(\boldsymbol{w}_{t+1})|$

In each iteration, the Prox-BMRM first tries to compute a new iterate by minimizing the reduced objective with the prox-center penalty ρ_t used in the previous step (line 3).

If the new iteration improves primal objective sufficiently, i.e. primal objective value decreases by more than γ_t (line 4), the solution is accepted, and setting of the penalty ρ_t as well as the minimal improvement threshold γ_t are unchanged (line 5). If the improvement is not sufficient, the prox-term penalty is tuned to guarantee that distance between the previous and the new iterate is not higher than constant K (line 7). At the same time, minimal improvement threshold is set to a new value γ_{t+1} according to the formula on line 9. It can be seen, that if the improvement in all subsequent iterations is not less than γ_{t+1}, i.e. condition on the line 4 holds, then the stopping condition is satisfied after at most T iterations. In turn, the prox-center penalty is re-adjusted not later than after T iterations. As a result, Prox-BMRM guarantees in each iteration that either the primal objective is sufficiently improved or the new iterate is not overly far from the previous one. In Section 5.6, we experimentally show, that this strategy avoids the "zig-zag" behavior, and also significantly decreases the number of iterations needed to converge to the ε-optimal solution.

Compared to the original BMRM, proposed Prox-BMRM introduces an additional overhead because the solution of the reduced problem can be required several times in a single iteration. However, the overhead is not dramatic. Moreover, it can be significantly reduced by using several tricks. First, in the search for ρ_t on line 7 one should use the fact that $\|\boldsymbol{w}_{t+1}^{\rho_1} - \boldsymbol{w}_t\| > \|\boldsymbol{w}_{t+1}^{\rho_2} - \boldsymbol{w}_t\|$ holds for any $\rho_1 < \rho_2$ which follows from the strict convexity of the quadratic prox-term. Also, the search can start from the previous value of ρ_t instead of always going sequentially from $\rho_t = 0$. Second, one can significantly speed up solving the reduced problem by using the warm start strategy. Third, the stopping condition on the line 11, which also requires solving the reduced problem with $\rho_t = 0$ to get lower bound on the optimum, does not need to be evaluated in every iteration. It turns out to be sufficient to evaluate the stopping condition only when ρ_t readjusting takes place as it requires solving the reduced problem nonetheless. With these tricks implemented we observed, that the reduced problem is solved on average 2–3 times instead of only once as in BMRM, which constitutes a negligible increase of a computational time. This increase is amply compensated by a reduced number of iterations.

Besides the precision parameter ε, Prox-BMRM algorithm requires the setting of an initial solution $\boldsymbol{w}_1$ and two constants: i) K which is a maximal distance between two consecutive iterations, and ii) T which is a maximal number of iterations without readjusting the prox-center penalty α. An efficient and straightforward way to find a non-trivial initial solution, i.e. $\|\boldsymbol{w}_1\| > 0$, is discussed in the next section. We found empirically that setting $T = 100$ and $K = 0.01\|\boldsymbol{w}_1\|$ worked consistently well in all our experiments.

Algorithm 4 converts solving the original problem (4.5) to a sequence of reduced problems (4.27). Problem (4.27) is equivalent to the following convex quadratic program:

$$\begin{aligned} \boldsymbol{w}_{t+1} &\in \arg\min_{\boldsymbol{w}\in\mathbb{R}^n} \left[\frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \rho_t\|\boldsymbol{w} - \boldsymbol{w}_t\|^2 + \xi\right] \\ \text{s.t.} \quad \xi &\geq b_i + \langle \boldsymbol{a}_i, \boldsymbol{w}\rangle,\ i = 0, \ldots, t-1\ . \end{aligned} \tag{4.28}$$

In practice, the number of cutting planes t required by Algorithm 4 to converge is usually much lower than dimension n of parameter vector $\boldsymbol{w} \in \mathbb{R}^n$. Thus, one can benefit from solving the reduced problem (4.19) in its dual formulation, the form of which is very similar to the one used in the standard BMRM. Let $\mathbf{A} \in \mathbb{R}^{n\times t}$ be a matrix the columns of which are sub-gradients $\boldsymbol{a}_i = \boldsymbol{R}'(\boldsymbol{w}_i)$, $\boldsymbol{b} \in \mathbb{R}^t$ a vector the elements of which are $b_i = R(\boldsymbol{w}_i) - \langle \boldsymbol{R}'(\boldsymbol{w}_i), \boldsymbol{w}_i\rangle$, $\mathbf{H} = \frac{1}{\lambda+2\rho_t}\mathbf{A}^\top\mathbf{A}$ and $\boldsymbol{z} = \boldsymbol{b} + \frac{2\rho_t}{\lambda+2\rho_t}\mathbf{A}^\top\boldsymbol{w}_t$. Using

these shortcuts, the Lagrange dual of (4.28) can be concisely written as

$$\boldsymbol{\beta}_{t+1} \in \operatorname{Arg}\max_{\boldsymbol{\beta}\in\mathbb{R}^t} \left[-\frac{1}{2}\boldsymbol{\beta}^\top \mathbf{H}\boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{z} \right] \quad \text{s.t.} \quad \sum_{i=1}^{t} \beta_i = 1\,, \beta_i \geq 0, \quad i = 1, \ldots, t\,. \tag{4.29}$$

The primal solution can be obtained from the dual solution by the formula

$$\boldsymbol{w}_{t+1} = \frac{2\rho_t \boldsymbol{w}_t - \mathbf{A}\boldsymbol{\beta}_{t+1}}{\lambda + 2\rho_t}\,. \tag{4.30}$$

The experimental evaluation of proposed Prox-BMRM algorithm is covered in Section 5.6. In the following section, we introduce another improvement of BMRM, the BMRM with multiple cutting plane models, which can be used either independently, or in combination with the Prox-BMRM.

4.7. Multiple cutting plane model BMRM

In this section, we describe our second improvement of the BMRM algorithm. While the genuine BMRM uses a single cutting plane model (4.16) to approximate risk $R(\boldsymbol{w})$, we propose to decompose the risk into a sum of $P > 1$ functions and then approximate each partial risk by a separate CPM.

Let us assume that the risk is decomposed into P functions

$$R(\boldsymbol{w}) = \sum_{p=1}^{P} R(\boldsymbol{w}, p) \qquad \text{where} \qquad R(\boldsymbol{w}, p) = \sum_{i\in I_p} r_i(\boldsymbol{w})\,, \tag{4.31}$$

and $I_1, \ldots, I_P$ is a partitioning of the index set $\{1, \ldots, m\}$. It is wise to define the partitioning evenly so that the sets I_p, $p = 1, \ldots, P$, have approximately the same size.

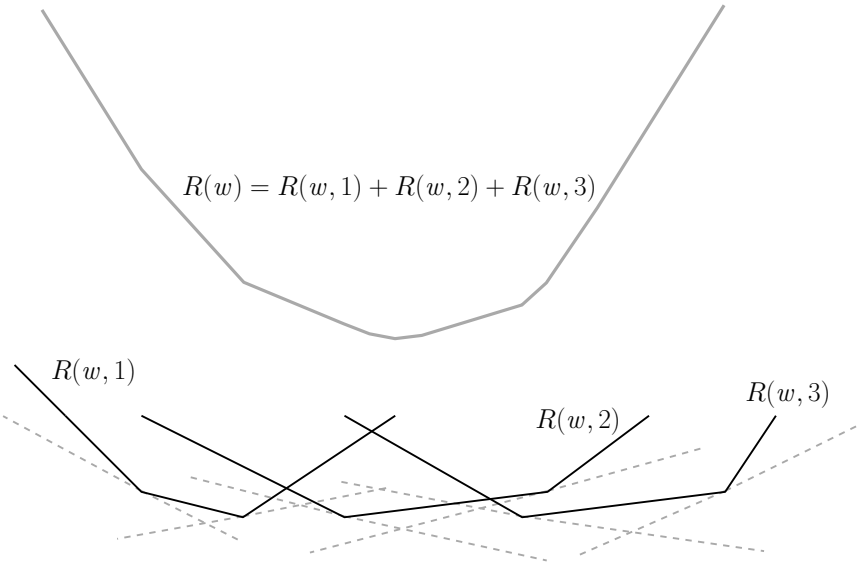


Figure 4.3. The figure illustrates how is the risk $R(\boldsymbol{w})$ (upper figure) decomposed into a sum of functions $R(\boldsymbol{w}, p)$, $p = 1, \ldots, P$, (lower figure) each of which is approximated by its cutting planes (dashed lines).

We propose to approximate each of P partial risks $R(\boldsymbol{w}, p)$ by its own cutting plane model. Similarly as in the previous sections, let us define shortcuts $\boldsymbol{a}_{i,p} = \boldsymbol{R}'(\boldsymbol{w}_i, p)$

and $b_{i,p} = R(\boldsymbol{w}, p) - \langle \boldsymbol{R}'(\boldsymbol{w}_i, p), \boldsymbol{w}_i \rangle$. Then, the cutting plane model of $R(\boldsymbol{w}, p)$ can be expressed as

$$R_t(\boldsymbol{w}, p) = \max_{i=1,\ldots,t} \left[b_{i,p} + \langle \boldsymbol{a}_{i,p}, \boldsymbol{w} \rangle \right] . \tag{4.32}$$

The multiple cutting plane model of the risk $R(\boldsymbol{w})$ is then defined as a sum

$$R_t(\boldsymbol{w}) = \sum_{p=1}^{P} R_t(\boldsymbol{w}, p) . \tag{4.33}$$

Note, that for $P = 1$, equation (4.33) reduces to the original model (4.16). It is immediately seen, that the multiple cutting plane model (4.33) preserves the crucial properties of the original model, i.e., $R_t(\boldsymbol{w})$ is a lower bound of $R(\boldsymbol{w})$, which is tight at points $\boldsymbol{w}_i$, $i = 1, \ldots, t$. The idea is illustrated in Figure 4.3.

The proposed multiple CP BMRM denoted as P-BMRM, is obtained by substituting multiple CPM (4.16) for the original model (4.16) in the definition of the reduced problem (4.19). Otherwise, the genuine BMRM Algorithm 3 stays unchanged. Updating of the multiple CPM (step 5 of Algorithm 3) has the same computational complexity as in the genuine BMRM. The higher accuracy is compensated by increased memory requirements due to storing P times more CPs. The hyperparameter P allows controlling the trade-off between the precision of the approximation and the memory requirements. As we will verify experimentally in Section 5.6, the higher value of P implies more accurate model and the subsequently smaller number of iterations to achieve the desired precision.

The reduced problem with the multiple CPM (4.33) can be expressed as a following quadratic program

$$\begin{aligned} \boldsymbol{w}_t \quad \in \quad & \arg\min_{\boldsymbol{w} \in \mathbb{R}^n} \left[\frac{\lambda}{2} \|\boldsymbol{w}\|^2 + \sum_{p=1}^{P} \xi_p \right] \\ \text{s.t.} \quad & \langle \boldsymbol{a}_{i,p}, \boldsymbol{w} \rangle + b_{i,p} \leq \xi_p, \quad i = 1, \ldots, t , \quad p = 1, \ldots, P . \end{aligned} \tag{4.34}$$

The Lagrange dual of (4.34) reads

$$\begin{aligned} \boldsymbol{\alpha}_t \quad \in \quad & \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^t} \left[\sum_{i=1}^{t} \sum_{p=1}^{P} \alpha_{i,p} b_{i,p} - \frac{1}{2\lambda} \left\| \sum_{i=1}^{t} \sum_{p=1}^{P} \alpha_{i,p} \boldsymbol{a}_{i,p} \right\|^2 \right] \\ \text{s.t.} \quad & \sum_{i=1}^{t} \alpha_{i,p} = 1, \quad p = 1, \ldots, P , \\ & \alpha_{i,p} \geq 0, \quad i = 1, \ldots, t , \quad p = 1, \ldots, P . \end{aligned} \tag{4.35}$$

The primal solution is obtained from the dual solution $\boldsymbol{\alpha}_t$ by

$$\boldsymbol{w}_t = \frac{1}{\lambda} \sum_{i=1}^{t} \sum_{p=1}^{P} \alpha_{i,p} \boldsymbol{a}_{i,p} . \tag{4.36}$$

The dual problem (4.35) has P linear equality constraints instead of a single one as in the original dual (4.25). However, the variables bound by the constraints are decoupled which allows using sequential minimal solvers, if needed.

Experimental evaluation of proposed P-BMRM algorithm is presented in Section 5.6. Besides this, the multiple CPM can be used together with the Prox-BMRM described in the previous section. The algorithm using both ideas, that is, the adaptively updated

prox-term and the multiple CPM, is denoted as Prox-P-BMRM. Experiments show that while both proposed algorithms, Prox-BMRM, and P-BMRM, improve the genuine BMRM algorithm significantly, the best results are achieved by Prox-P-BMRM, being their combination.

5. Experiments

This chapter is divided into six sections dedicated to an empirical evaluation of methods proposed in this thesis. In Section 5.1, we summarize specific instances of proposed DPM detector as well as all competing landmark detection methods assessed in our evaluation. The metrics used for evaluation of the detection accuracy are defined in Section 5.2. Section 5.3 presents results on single-view databases, and Section 5.4 on multi-view databases. Single-view databases contain faces in a near frontal view in contrast to multi-view databases containing also profile faces. The detection time of compared detectors is evaluated in Section 5.5. The last section, Section 5.6, presents an empirical evaluation of proposed improvements of the BMRM algorithm for learning SO-SVM classifiers.

5.1. Evaluated landmark detection methods

In this section, we describe instances of proposed DPM detector and the contemporary state-of-the-art methods to which we compare the proposed detector. Namely, we compare to a tree-based DPM detector of Zhu and Ramanan [2012], which is a most related method to our work. Despite having relatively small localization accuracy, it is up to our best knowledge the only publicly available truly multi-view detector, that is, working in the full range of the yaw angle. We also compare to the IntraFace [Xiong and la Torre, 2013] detector, considered to be the current state-of-the-art in both precision and speed, the detector of Kazemi and Sullivan [2014], GN-DPM [Tzimiropoulos and Pantic, 2014], Chehra [Asthana et al., 2014], STASM [Milborrow and Nicolls, 2014], and the detector of Everingham et al. [2006]. Also, we include the results of two commercial solutions: landmark detector provided in the iPad SDK and detector of face.com, a web service dedicated to face recognition which was later acquired by Facebook.

Some of the competing algorithms use a face detector as their integral part. To make a fair comparison to these methods, we crop test images around the face box enlarged by 30%. Doing this should minimize the influence of failures of the particular face detector. Different methods may detect a different number of landmarks. In all evaluations, we use a subset of landmarks which is common to all compared methods. In particular, we use a set of 49 landmarks for the single-view (frontal faces) and 18 landmarks for the multi-view databases. In the multi-view experiments, we also evaluate the detector's ability to estimate the viewpoint. Since some competing methods do not estimate the viewpoint, we extend these methods with the head-pose estimator based on a fitting of a 3D-model to the localized landmarks [Asthana et al., 2013]. The estimated yaw angle (i.e. the viewpoint) is then rounded to the intervals defined in Table 3.1.

5.1.1. Proposed single-view detectors

In this section, we describe our implementations of proposed DPM detector suitable for detection in a single-view near frontal facial images. Here, we summarize only a particular configuration and design choices that are not described in Chapter 3. The detector's parameters are learned by the SO-SVM algorithm (4.5) optimizing the single-view loss (4.10).

The input of the detector is a sub-image cropped from the input image around a bounding box localized by a face detector. We use a commercial face detector[1] learned by the WaldBoost algorithm [Šochman and Matas, 2005]. The face detector returns a face box encoding a rough estimate of the face location, the face size, and the in-plane rotation. In addition, our face detector provides a rough estimate of the viewpoint.

DPM detector of 8 facial landmarks (L8-DPM) The graph of deformable parts for the set of 8 landmarks (L8 landmark configuration) is depicted in Figure 3.9. The detector uses the normalized frame of size 40×40 pixels. The normalized frame is obtained by applying a face detector, enlarging the face box by a factor of 1.2 and affinely transforming the cropped image to size 40×40 pixels, without the Gaussian filter smoothing. The patches of the local parts, from which we compute the features, are 10×10 pixels for the non-root landmarks and 20×20 pixels for the root landmark. The root landmark is an artificial landmark representing a face center. The artificial landmark is computed from 7 landmarks (eye corners, mouth corners, and the nose tip) annotated in the database. We evaluate two variants of the detector using S-LBP features, and HOG features. The S-LBP variant of the DPM detector has $\dim(\boldsymbol{w}) = 232,476$ parameters in total, the majority of them corresponding to templates on top of the S-LBP features. In contrast, the HOG variant of the DPM detector has only $\dim(\boldsymbol{w}) = 6,976$ parameters in total.

Independent SVM detector We evaluate a baseline method composed of independent Support Vector Machine (SVM) detectors. The SVM detectors use the S-LBP features and the same patch sizes and search spaces as the L8-DPM detector. The only difference is that landmarks are estimated independently. In turn, detectors are trained independently by the SVM algorithm which optimizes the L_1 distance between the true and the estimated landmark position, similarly as in the structured case.

Coarse DPM detector of 68 landmarks (C-DPM) The graph of deformable parts for a dense set of 68 landmarks (L68 landmark configuration) is depicted in Figure 3.1. The very same graph is used by F-DPM and C2F-DPM detectors described below. The C-DPM detector uses the normalized frame of size 80×80 pixels. The normalized frame is obtained by affinely transforming an image cropped around a face box enlarged by a factor of 1.5. Before computing the S-LBP features, the normalized frame is smoothed out by applying a 2D Gaussian filter with $\sigma = 1.0$. The patches used to compute S-LBP features for the appearance model are of size 13×13 pixels for all landmarks except the root landmark (the landmark $\boldsymbol{s}_{31}$), which has a size of 21×21 pixels. The C-DPM detector has $\dim(\boldsymbol{w}) = 2,478,348$ parameters in total.

Fine DPM detector of 68 landmarks (F-DPM) A size of the normalized frame is set to 160×160 pixels. The face box is extended by a factor of 1.25. The 2D Gaussian filter with $\sigma = 0.85$ is applied on the normalized frame for smoothing. Patches of the appearance model are of size 15×15 pixels for non-root landmarks and 21×21 pixels for the root landmark. The F-DPM detector has $\dim(\boldsymbol{w}) = 3,456,012$ parameters in total.

Coarse-to-fine DPM detector of 68 landmarks (C2F-DPM) Using a normalized frame with a higher resolution leads to a higher detection accuracy, but also higher processing time. To keep the processing time low, we use the proposed coarse-to-fine

[1] Provided by courtesy of Eyedea Recognition `www.eyedea.cz`

search strategy described in Section 3.6. Briefly, the idea is to use a coarse detector C-DPM to obtain precisely localized face box. The precise face box is then used as an input of a fine detector F-DPM operating on a higher resolution image. The processing time remains relatively low since the precise face box allows us to use tighter search spaces for individual landmarks.

5.1.2. Proposed multi-view detectors

In this section, we describe the implementation of our multi-view DPM detectors suitable for processing facial images when the viewpoint is unknown. All implemented detectors are learned to localize a set of 21 landmarks.

Multi-view detector from independent DPM detectors As a baseline multi-view detector, we use the following method. We learn a set of independent single-view DPM detectors each for a different viewpoint $\phi \in \Phi$. The particular single-view detector is selected based on a response of the face detector which outputs a rough estimate of ϕ. The individual single-view detectors use the same graph structure and features as the two multi-view detectors described below. The normalized frame is of size 60×60 pixels. The S-LBP features are computed from patches which have the same size, 9×9 pixels, for all but the root landmark. The root landmark corresponding to the tip of the nose (visible in all views) is of size 15×15 pixels. Before computing the features, the normalized frame is smoothed out by a 2D Gaussian filter with $\sigma = 0.4$. Thanks to the self-occlusions the number of landmarks depends on the particular view. The number of landmarks and the corresponding graph structured for individual views are described in Table 3.1 and Figure 3.3, respectively.

Parameters of the independent detectors are learned by the SO-SVM algorithm (4.5) which optimizes the single-view loss (4.10). As the normalization factor $\kappa(\boldsymbol{s})$ we use the face size defined as a distance between the root of the nose and the chin (i.e. the face height), that is, the distance $\|\boldsymbol{s}_{09} - \boldsymbol{s}_{21}\|_2$ using the notation from Figure 3.3.

Single-stage multi-view DPM detector The graph structure and features of the individual components of the multi-view detector are identical to the independent single-view detectors described in the previous section. The main difference is that the single-stage multi-view detector estimates the viewpoint $\phi \in \Phi$ and landmark coordinates simultaneously, solving a single (discrete) max-sum problem (3.2). For more details, see Section 3.2.1. Parameters of the detector are learned by SO-SVM algorithm (4.5) which optimizes the multi-view loss function (4.11) defined in Section 4.2. As the normalization factor $\kappa(\boldsymbol{s})$ we use the face height (FH) defined as a distance between the root of the nose and the chin.

Two-stage multi-view DPM detector The graph structure and features used for individual viewpoints are identical to the independent single-view detectors. The two-stage multi-view detector in the first stage applies a set of single-view detectors. Second, it uses the response of the first stage to compute features which are consequently used to estimate the correct viewpoint, that is, it selects the single-view detector whose response best matches the input image. The two-stage detection model is described in Section 3.2.2. The parameters of the single-view detectors are learned in the same way as the independent detectors. The multi-class classifier of the second stage is learned by the SO-SVM optimizing the 0/1-loss function defined on the viewpoint $\phi \in \Phi$.

5.1.3. Existing methods

Detector of Zhu and Ramanan [2012] We use the code provided by the authors with the fully shared model "p99". This detector simultaneously works as the face detector and the detector of facial landmarks. The detector returns 68, or 39 landmarks for the near-frontal, or the profile viewpoint, respectively. The detector of Zhu and Ramanan [2012] uses a part of the Multi-PIE database for training which is not consistent with our split. Hence, the corresponding results on the Multi-PIE dataset might be positively biased for this detector.

Chehra [Asthana et al., 2014] We use the implementation of a recently published facial landmark tracker provided by the authors. The detector is based on a cascade of discriminatively trained regressors estimating the pose and shape parameters of a 3D face model. The detector was trained on the 300-W dataset [Sagonas et al., 2013a]. It returns 49 landmarks as well as the estimation of the 3D head-pose orientation [Asthana et al., 2013].

IntraFace [Xiong and la Torre, 2013] We use the code kindly provided by the authors. The detector's learning is formulated as nonlinear least squares (NLS) problem the goal of which is to match the face model to the image. The NLS is solved by the SDM algorithm, learning a descent direction from the training data. The detector returns a set of 49 landmarks and an estimate of the viewpoint. The detector was learned on a subset of Multi-PIE and LFW [Huang et al., 2007] datasets. Therefore, the results on these two datasets might be positively biased in favor of this detector because we use a different split.

GN-DPM [Tzimiropoulos and Pantic, 2014] We use a code provided by the authors. This detector is an instance of a generative DPM, where the optimization of the appearance and global shape model is done simultaneously by the Gauss-Newton algorithm. It detects 49 landmarks. The detector is initialized from a response of the Zhu and Ramanan [2012] detector. The detector was trained on the LFPW [Belhumeur et al., 2011] dataset which is a part of the 300-W benchmark.

Kazemi & Sullivan [Kazemi and Sullivan, 2014] We use the implementation from the "dlib C++" library. This detector is based on a gradient boosting of an ensemble of regression trees. The detector estimates a set of 68 landmarks. The detector is trained on the iBUG dataset which overlaps with the testing part of the 300-W benchmark. For this reason, we compare to this method only on the Annotated Facial Landmarks in the Wild (AFLW) and Multi-PIE datasets.

Active Shape/Appearance Models (ASM/AAM) We use the STASM detector [Milborrow and Nicolls, 2014] which is a well-tuned instance of the ASM. The STASM detector is considered to be one of the best publicly available facial landmark detectors nowadays [Çeliktutan et al., 2013]. In the experimental evaluation, we use this detector on the precisely same input as our proposed detector, and we select just a relevant subset of facial landmarks out of the total number of 77.

Detector of Everingham et al. [Sivic et al., 2009] The detector of Everingham et al. [2006, 2008]; Sivic et al. [2009] is yet another representative of the DPM detectors. It was trained on a collection of consumer images which are, however, not available. This detector outputs 9 landmarks: canthi of eyes, corners of the mouth and three points

on the nose. Unlike our approach, the appearance model and the deformation costs are learned independently.

iPad (SDK 2012) The iPad device comes with a facial recognition SDK. One component of the SDK from 2012 is a simple facial landmarks detector. The detector returns just 3 points: the center of the left and right eye and the center of the mouth. Unfortunately, we do not know what type of detector is used nor from which training examples it was learned. Although the SDK is a bit outdated, we include it to the comparison just for curiosity.

face.com The `face.com` used to be a commercial web service dedicated to face recognition. It was acquired by Facebook in the middle of 2012. It provides a facial landmark detector estimating 6 landmarks: the face center, centers of eyes, the mouth corners, the center of the mouth and nose. The used detection method, as well as training data, are unknown.

5.2. Evaluation metrics

In this section, we describe evaluation metrics used to measure the performance of single-view and multi-view detectors.

5.2.1. Single-view error

To evaluate the single-view detector, we use a localization error (also called P2P error) defined as

$$E_{\text{loc}}(\hat{\boldsymbol{s}}, \boldsymbol{s}) = \frac{1}{\kappa(\boldsymbol{s})|V|} \sum_{j=1}^{|V|} \|\hat{\boldsymbol{s}}_j - \boldsymbol{s}_j\| \,, \tag{5.1}$$

where $\kappa(\boldsymbol{s})$ is a normalization factor, $\boldsymbol{s}$ are the ground truth landmark locations and $\hat{\boldsymbol{s}}$ are their predictions, returned by the detector. In existing literature $\kappa(\boldsymbol{s})$ is most often set to the IOD computed from the ground truth annotation $\boldsymbol{s}$, namely, using the location of the eyes (c.f. Section 4.2 for more details). The IOD is a reasonable option for near frontal images. However, for non-frontal images, the value of IOD is significantly influenced by the yaw angle of the face. Moreover, the IOD is zero for profile views. A more appropriate normalization factor for non-frontal images is the vertical face size (face height) as defined in Section 4.2 since the pitch angle is more stable in natural face images. Despite the mentioned disadvantage, in most experiments, we use the IOD to make our evaluation compatible with the results in the existing literature. The vertical face size is used as the normalization factor in the experiments on the LFW, AFLW, and Multi-PIE benchmarks.

To assess a given detector, we report a cumulative histogram of its localization errors computed on test examples. The y-axis of the cumulative histogram corresponds to the percentage of test faces with a localization error not higher than the corresponding value on the x-axis. Also, we report the A5, and A10 score being the percentage of test examples with the localization error not higher than 5%, or 10% of the IOD (or face size), respectively. In other words, A5, and A10 scores represent two points on the cumulative histogram. Intuitively, A5 score is the percentage of test faces where the detector provides very precise prediction (on the level of a human precision). The A10 score is then the percentage of test faces on which the evaluated detector still works reasonably well.

Finally, it is worth mentioning that the definition of the localization error (5.1) coincides with the single-view loss (4.10) the convex proxy of which is optimized by the SO-SVM algorithm when learning parameters of proposed DPM detectors.

5.2.2. Multi-view errors

Besides landmark locations $\hat{\boldsymbol{s}}$, the multi-view detector also returns an estimate of the viewpoint $\hat{\phi}$. Because the evaluated methods are integrated with a face detector, they first try to detect whether the input image contains a face or not. If no face is detected, the detector outputs an empty set $\hat{\phi} = \emptyset$. To evaluate multi-view detectors, we use the multi-view localization error defined as

$$E_{\text{mv}}(\hat{\phi}, \hat{\boldsymbol{s}}, \phi, \boldsymbol{s}) = \begin{cases} \frac{1}{\kappa(\boldsymbol{s})|V^{\phi}|} \sum_{j=1}^{|V^{\phi}|} \|\hat{\boldsymbol{s}}_j - \boldsymbol{s}_j\|, & \text{if } \hat{\phi} = \phi \\ \infty, & \text{if } \hat{\phi} = \emptyset \text{ or } \hat{\phi} \neq \phi \end{cases} \tag{5.2}$$

where $\boldsymbol{s}$, $\hat{\boldsymbol{s}}$ are defined as before, ϕ is a ground truth viewpoint, and $\hat{\phi}$ is its prediction. Compared to the single-view error (5.1), the multi-view localization error, in addition, accounts for the option that the detector incorrectly predicts the viewpoint, $\hat{\phi} \neq \phi$, or that it fails to detect the face, $\hat{\phi} = \emptyset$. In such case, the penalty is set to ∞.

For a given detector, we report the cumulative histogram of the multi-view localization error computed on test examples. We also report the A5, and A10 score defined above.

Here, we also mention a similarity between the multi-view localization error (5.2), and the loss function (4.11) which is optimized during the learning the proposed multi-view detector. The only difference is that ∞ penalty in the definition of (5.2) is replaced by a finite constant in (4.11) to make the usage of the margin re-scaling loss possible.

Yaw angle prediction error To quantify the error made in predicting a correct viewpoint ϕ (i.e. the yaw angle), we use the viewpoint prediction error

$$E_{\text{yaw}} = \frac{1}{m} \sum_{i=1}^{m} [\![\hat{\phi}^i \neq \phi^i]\!] \,, \tag{5.3}$$

where $[\![\cdot]\!]$ stands for the Iverson brackets, ϕ^i is the ground truth viewpoint of the i-th example, $\hat{\phi}^i$ is the detector's estimate on the i-th example viewpoint, and the sum goes over m examples in a test set. The value of E_{yaw} is thus an estimate of the probability that the detector predicts the viewpoint incorrectly.

Face detection failure To measure the probability that the detector overlooks a face, we define a face detection error

$$E_{\text{fd}} = \frac{1}{m} \sum_{i=1}^{m} [\![\phi^i = \emptyset]\!] \,, \tag{5.4}$$

To measure the probability that the detector overlooks a face, we define a face detection error where the sum goes over m test examples. In other words, E_{fd} is an estimate of the recall of the integrated face detector. Since the existing benchmarks do not contain examples of non-face images, we measure only the recall and not the precision of the detector.

5.3. Single-view experiments

In this section, we report experiments on two datasets. First, we use the LFW dataset composed of photographs of celebrities in near frontal poses taken in uncontrolled environments. The LFW is a standard benchmark for face verification algorithms. We endowed the LFW dataset with a manual annotation of 7 landmarks. We were using the LFW dataset at the beginnings of our work on the topic when there was no reasonably large unconstrained, i.e. "in-the-wild", publicly available landmark detection benchmark. Second, we evaluate detectors on the 300-W dataset, which is nowadays the standard benchmark for near-frontal landmark detection. The 300-W ataset [Sagonas et al., 2013b,a, 2016; Tzimiropoulos et al., 2012] is a collection of "in-the-wild" faces annotated with 68 landmarks.

5.3.1. LFW dataset

The first experiment is conducted on the LFW dataset [Huang et al., 2007] with manual annotation of 7 landmarks, namely, 4 canthi (corners of the eyes), 2 mouth corners, and the tip of the nose. We found experimentally that the DPM detector is more robust if it estimates an additional artificial landmark representing the center of the face. The location of the 8-th landmark is computed from the 7 manually annotated landmarks. However, the artificial 8-th landmark is not included in the computation of the localization error.

Except for the iPad and face.com, all other competing methods estimate, *inter alia*, these 7 landmarks. For iPad and face.com, we compute the localization error on a subset of 7 landmarks the two methods return. All methods evaluated in this experiments are listed in Table 5.1. In this experiment, we compute the localization error normalized by vertical face size, i.e. a face height (c.f. Section 4.2).

$13,233$ facial examples from the LFW dataset are randomly split into training, validation, and test sets, in a ratio $60/20/20$. $6,919$ examples from the training set are used to learn parameters of the DPM detector. The best performing regularization constant is selected on $2,307$ examples from validation examples. The remaining $2,316$ examples from the test set are used to compute localization error for all competing methods. Unfortunately, one of the competing methods, the IntraFace, uses a different subset of the LFW dataset for training, rendering the results positively biased in favor of this method.

We compare two variants of the proposed DPM detector using the same structure but different features to represent the landmark appearance. Namely, we use the DPM detector with S-LBP features and with the HOG features. In addition, we evaluate a baseline method composed of a set of independent SVM detectors using the S-LBP features but no structure between the landmarks.

Figure 5.1 shows the cumulative histogram of the localization error E_{loc} evaluated on test images of the LFW dataset. The $A5$, and $A10$ scores are summarized in Table 5.1. Based on the results we can draw the following conclusions:

- The DPM detector with S-LBP features performs significantly better than the DPM detector with the same structure using the HOG features. However, the DPM detector with HOG features still performs significantly better than the independent SVM detectors which use the S-LBP features. This suggests that employing the structure can increase the detector's performance more than using better features and ignoring the structure.

- In the overall comparison, we see, that the proposed DPM detector with S-LBP features outperforms all but one competing methods, which is the commercial In-

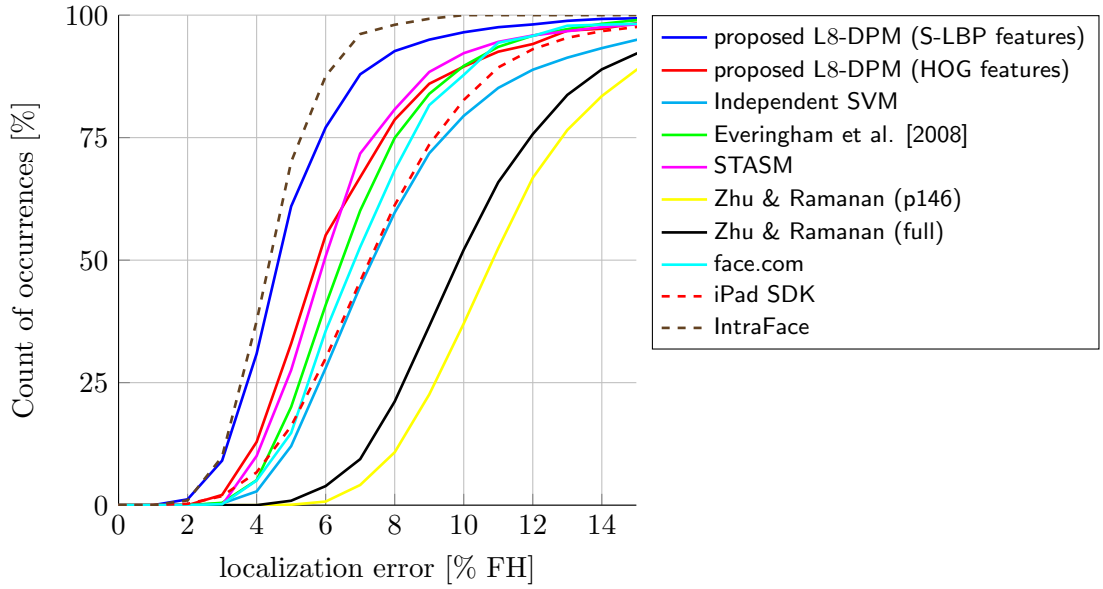


Figure 5.1. The cumulative histograms of localization error E_{loc} evaluated on test images of the LFW dataset.

traFace [Xiong and la Torre, 2013] detector. However, as we mentioned above, the results for the IntraFace might be positively biased, due to using a different subset of the LFW examples for training, i.e. IntraFace probably trains on some examples from our test set.

- The proposed DPM detector (both using the S-LBP and HOG features) is significantly better than all competing DPM-based detectors, i.e. Everingham et al. [2008] and two variants of Zhu and Ramanan [2012].

Table 5.1. The $A5$, and $A10$ scores obtained on the LFW dataset.

LFW dataset

method	$A5$	$A10$
proposed L8-DPM detector (S-LBP features)	60.97%	96.50%
proposed L8-DPM detector (HOG features)	32.97%	89.50%
independent SVM detector	12.04%	79.44%
Everingham et al. [2008]	20.03%	89.64%
STASM [Milborrow and Nicolls, 2014]	27.56%	92.23%
Zhu and Ramanan [2012] (p146)	0.09%	37.06%
Zhu and Ramanan [2012] (full)	0.90%	52.12%
face.com	14.74%	87.80%
iPad SDK	16.14%	82.68%
IntraFace [Xiong and la Torre, 2013]	70.05%	100.0%

5.3.2. 300W dataset

The 300-W dataset [Sagonas et al., 2013b,a, 2016] was created by the organizers of the "300 Faces in the Wild Challenge"[2]. The 300-W dataset consists of a public and

[2]The challenge has been organized twice to this date. Firstly, the challenge results were evaluated at the ICCV workshop in 2013 [Sagonas et al., 2013a]. Secondly, the challenge was associated to

non-public part. The public part released to the challenge participants is a collection of 6,193 re-annotated examples from LFPW, AFW, HELEN, XM2VTS, and IBUG datasets. The non-public part was hidden to the participants, and it was released after the end of the challenge in 2016. The non-public part consists of 600 faces images taken in uncontrolled environments. Out of the 600 images 300 are taken indoors, and 300 outdoors. The quality of images varies, but they are mostly sharp and of a high resolution.

The annotation of 68 landmarks was acquired by a semi-automatic method described in [Sagonas et al., 2013b]. To compute the localization error, we used a subset of 49 landmarks which is common to all competing methods[3]. Similarly to the evaluation metric of the 300-W challenge, we use the IOD as the face size normalization factor, and inf penalty to evaluate failures of a face detector.

The public part contains 6,193 images in total. We used its original split into training, and testing subsets. As our algorithm also requires a validation set for tuning the regularization constant, we further divide the original training subset into two parts. Finally, we have 3 subsets: 5,124 examples for training parameters of the detector, 551 validation examples for tuning the regularization parameter, and 518 testing images for evaluation of the localization error.

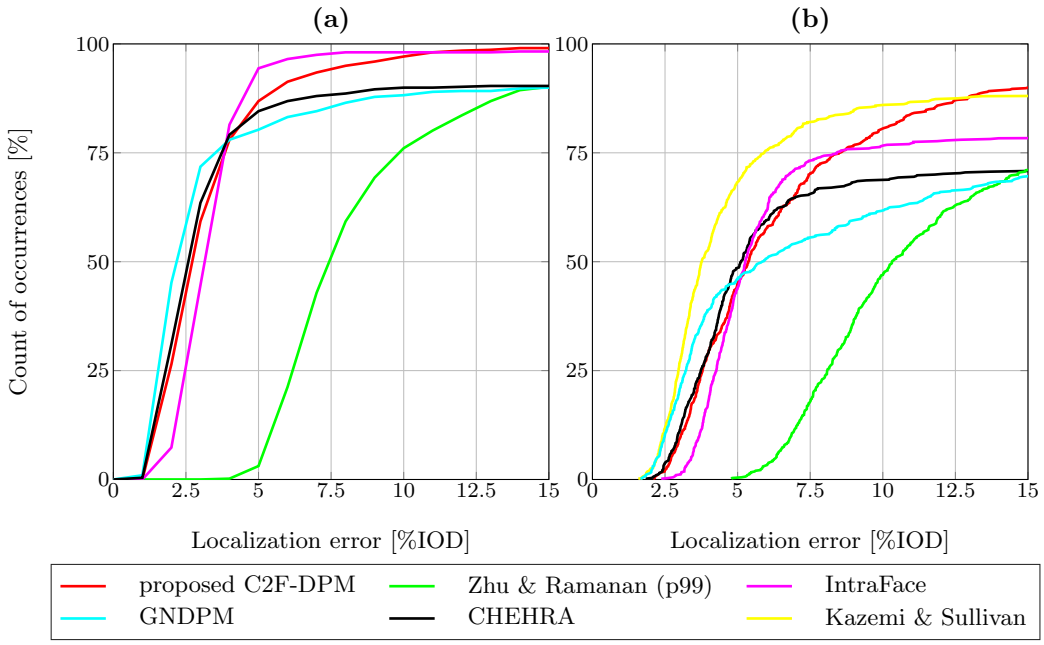


Figure 5.2. The cumulative histograms of the average localization error evaluated on (a) the public part and (b) non-public part of the 300-W dataset.

Cumulative histograms of the localization error are shown in Figure 5.2 for both the public and non-public part of the 300-W dataset. The corresponding $A5$ and $A10$ scores are summarized in Table 5.2. Exemplary images on which the proposed C2F-DPM detector provides the lowest, and the highest localization error are presented in Figure 5.3, and Figure 5.4, respectively. The obtained empirical results suggest that:

- Except for the detector of [Zhu and Ramanan, 2012], all competing methods return comparable results on the public part. Regarding the $A5$, and $A10$ score the best

a special issue of the "Image and Vision Computing Journal (IMAVIS)" with results summarized in [Sagonas et al., 2016].

[3]The 300-W competition used a subset of 51 landmarks for the error evaluation. The 2 missing landmarks in our setting correspond to the inner corners of the mouth.

detector is the IntraFace followed by the proposed C2F-DPM detector. It should be emphasized, that the worst performing method of [Zhu and Ramanan, 2012] is the only fully multi-view detector in this comparison, hence not exploiting the prior knowledge that test images contain near-frontal faces. Despite this fact, the [Zhu and Ramanan, 2012] detector still works reasonably well judging based on the $A10$ score.

- Localization errors of all competing methods are significantly worse in the non-public part of the 300-W dataset, compared to the public one. Recall, that the public part is composed of standard datasets while the non-public part was collected independently by the challenge organizers. The different results could be caused by over-fitting or, possibly, by a bias of the creators of the non-public part towards selecting more complicated images for a final test.

- The detector of Kazemi and Sullivan [2014] is significantly better than the other methods on the non-public part. The runner-up according to the $A5$ score is the IntraFace and the proposed C2F-DPM detector according to the $A10$ score.

- It is seen, that the proposed C2F-DPM detector has the highest percentage of test images with localization error not higher than 15% of the IOD. This holds for both public and non-public parts. In other words, the proposed detector makes the smallest number of "big mistakes" which can be attributed to its global inference procedure and a good face detector.

Table 5.2. The $A5$ and $A10$ scores obtained on the public and non-public parts of the 300-W dataset evaluated for all compared detection methods.

300-W dataset 49 **landmarks**

	public		non-public	
method	$A5$	$A10$	$A5$	$A10$
proposed C2F-DPM	86.87%	97.10%	45.33%	80.67%
Zhu and Ramanan [2012] (p99)	3.09%	76.06%	0.50%	47.00%
IntraFace [Xiong and la Torre, 2013]	94.40%	98.07%	44.00%	76.67%
GN-DPM [Tzimiropoulos and Pantic, 2014]	80.31%	88.22%	46.00%	61.83%
Chehra [Asthana et al., 2014]	84.56%	89.96%	48.50%	68.67%
Kazemi and Sullivan [2014] (dlib C++)	—	—	68.33%	85.83%

5.3.3. Evaluation of the Coarse-to-Fine search strategy

In this section, we evaluate the effectiveness of the proposed coarse-to-fine search strategy. We use the public part of the 300-W dataset and the same evaluation protocol as we described in the previous section. In Figure 5.5a, we show the localization error of different variants of the DPM detectors, which differ in the resolution of the normalized frame and the initialization method used. Recall, that the initialization method determines the size, and position of search spaces, by which it influences the accuracy and the execution time of the detector. We can see, that the F-DPM detector working on a high-resolution normalized frame but initialized directly from the response of the face detector works significantly worse than the C-DPM detector using a low-resolution normalized frame and the same initialization. The C-DPM detector works better because the low dimensional normalized frame implies wider area in the input image covered by search spaces by which it compensates the inaccurate initial

location of the face box. We can see, that using the response of the C-DPM to construct a better face box, and running the very same F-DPM detector, as implemented in the C2F-DPM detector, yields significantly better results. At the same time, the C2F-DPM detector has a smaller execution time because search spaces of the F-DPM detector are smaller. We also show the accuracy of the F-DPM detector initialized by an ideal face box computed directly from the ground truth landmark locations. We can see that the accuracy of the C2F-DPM detector closely matches the accuracy of the ideally initialized F-DPM detector.

5.3.4. Comparison of different loss functions

The most similar method to our approach is the detector of Zhu and Ramanan [2012]. One of the main conceptual differences is the loss function used in the learning of the detector's parameters. The detector of Zhu and Ramanan [2012] optimizes a convex surrogate of 0/1-loss function, which penalizes any deviation of the prediction from the ground truth positions equally. In contrast, we use a surrogate of the localization error, i.e. the objective function of our learning algorithm is directly connected to the actual performance measure.

Besides a different loss function, the detector of Zhu and Ramanan [2012] uses different design options, like the features, size of the normalized frame, optimization algorithm for learning, etc. In order to measure the effect of using a different loss function only, we learn the C-DPM detector using the 0/1-loss (4.12) and the single-view loss (4.10) which we propose.

In this experiment, we use the public part of the 300-W dataset and evaluation protocol as in the previous section. The resulting cumulative histogram of the localization error for the C-DPM detector learned with two different loss functions is depicted in Figure 5.5b. Also, we include results of the detector of Zhu and Ramanan [2012]. It is seen that using the proposed single-view loss function brings a constant improvement of approximately 1%, compared to the detector learned with the 0/1 loss.

In Figure 5.6, we show a sample of test examples with the highest absolute difference between the localization errors of the C-DPM detectors learned with two different loss functions. It can be seen that the detector learned with 0/1 loss fails significantly when the initial face box is not precise, or when the facial expression is exaggerated. It is also seen that the detector trained with 0/1 loss has problems with estimation of landmarks on the mouth and nose contours.

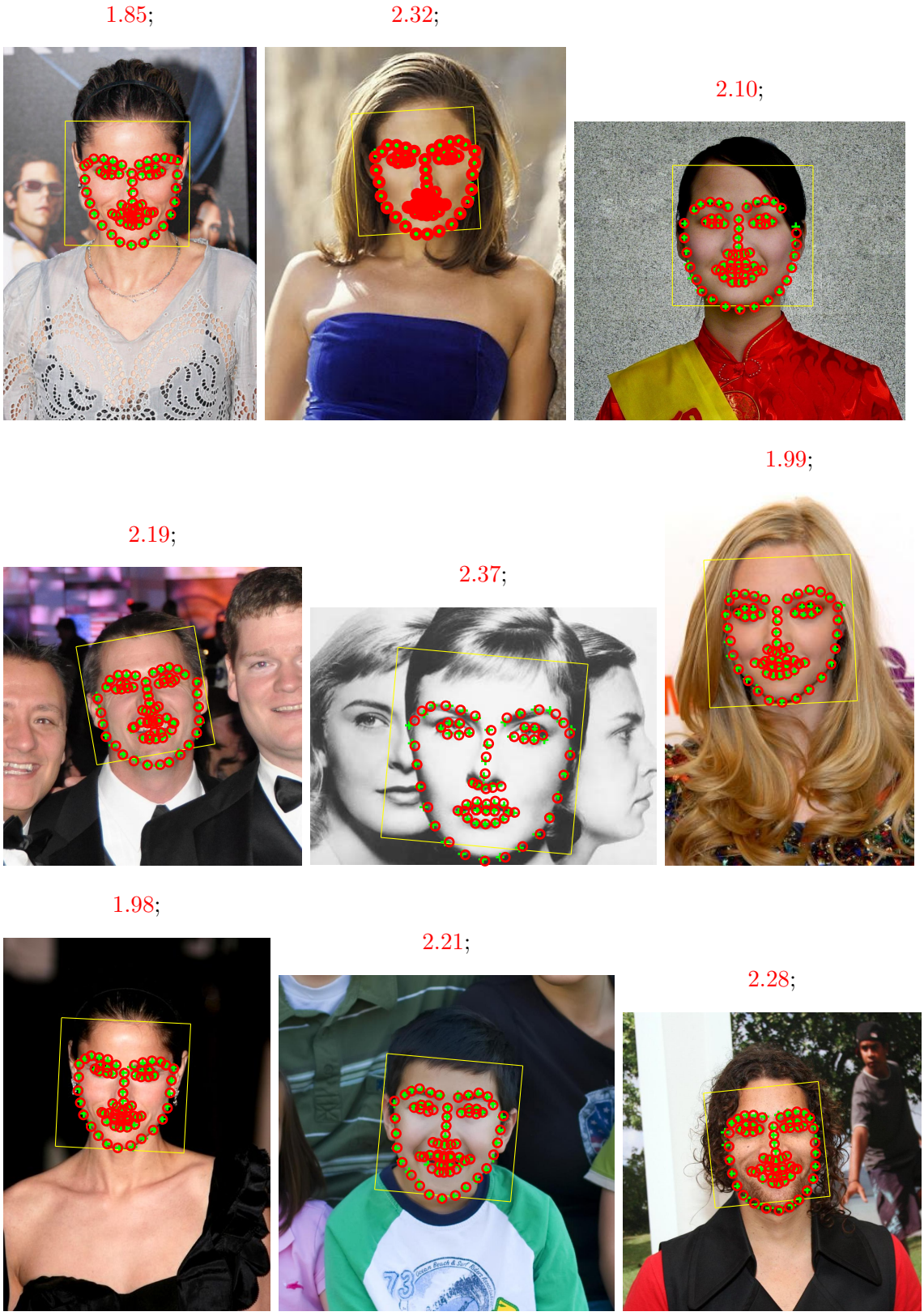


Figure 5.3. Exemplary images on which the proposed C2F-DPM detector achieved the smallest localization error. The localization error is normalized by the IOD and printed as a red number in the image title. The errors are computed on the public test part of the 300-W dataset. The ground truth annotation of 68 landmarks is represented by green plus markers while the predicted positions are red circles.

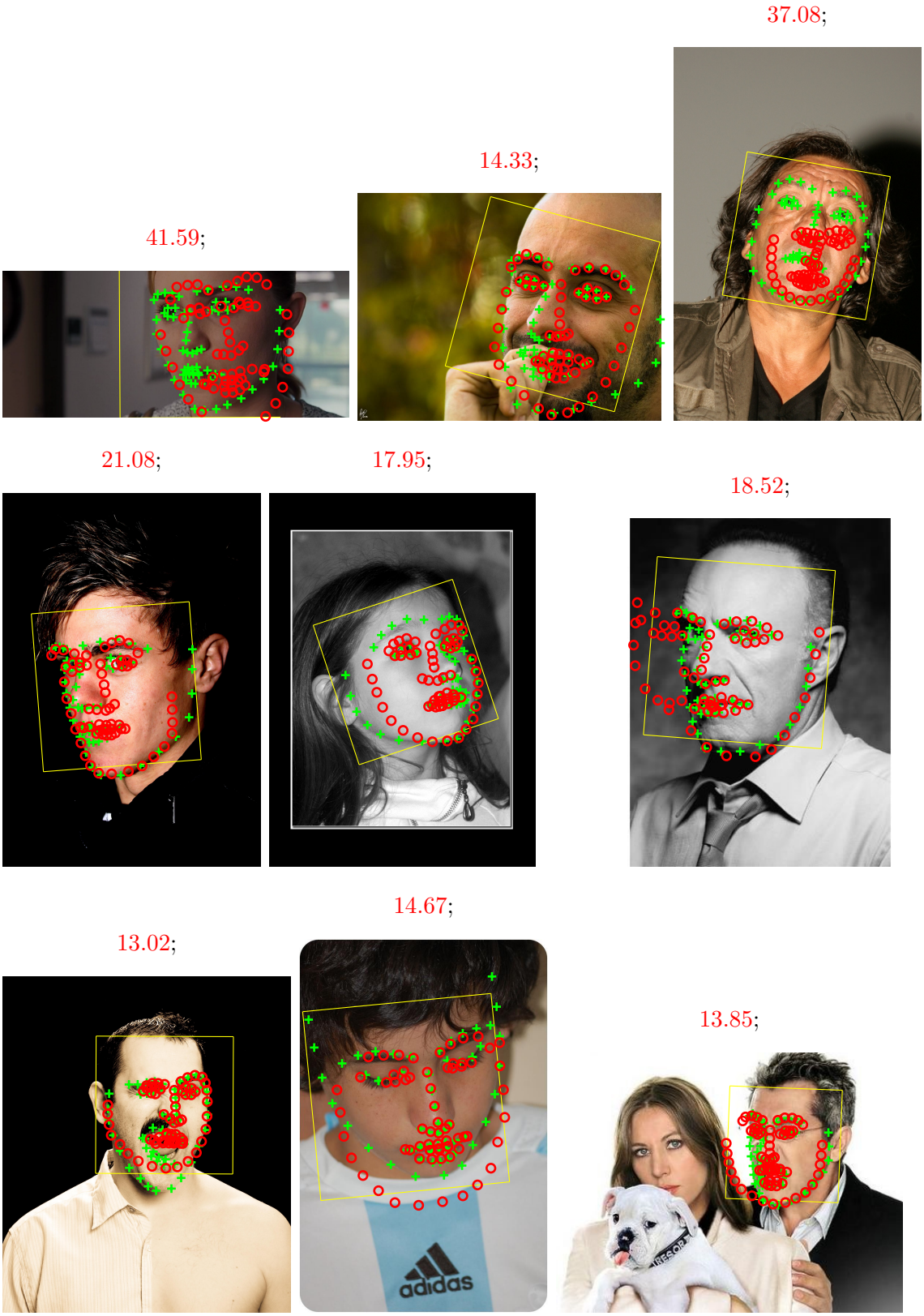


Figure 5.4. Exemplary images on which the proposed C2F-DPM detector achieved the highest localization error. The localization error is normalized by the IOD and printed as a red number in the image title. The errors are computed on the public test part of the 300-W dataset. The ground truth annotation of 68 landmarks is represented by green plus markers while the predicted positions are red circles.

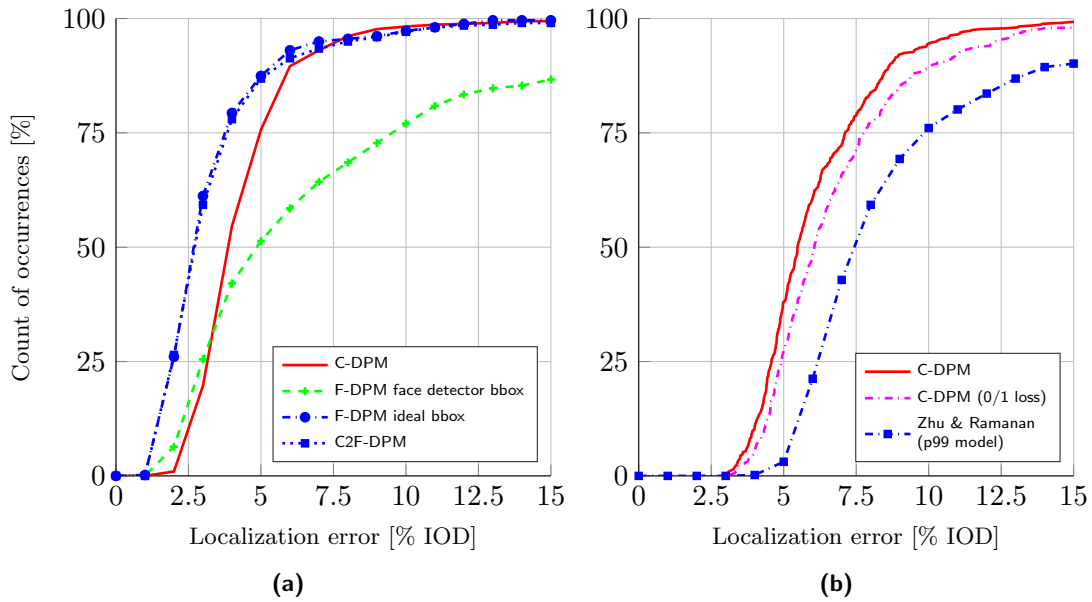


Figure 5.5. Figure (a) shows the localization error of different variants of DPM detectors which differ in the resolution of the normalized frame and the initialization method used. The C-DPM detector works on a low-resolution normalized frame, and it is initialized from the face detector. The localization error of the F-DPM detector working on a high-resolution normalized frame is shown for the initialization from a face detector and the ideal initialization computed from the ground truth annotations. The C2F-DPM detector uses the response of the C-DPM to initialize the F-DPM detector. Figure (b) shows the localization error of the C-DPM which was learned using different loss functions. Here, we measure the localization error on all 68 landmarks.

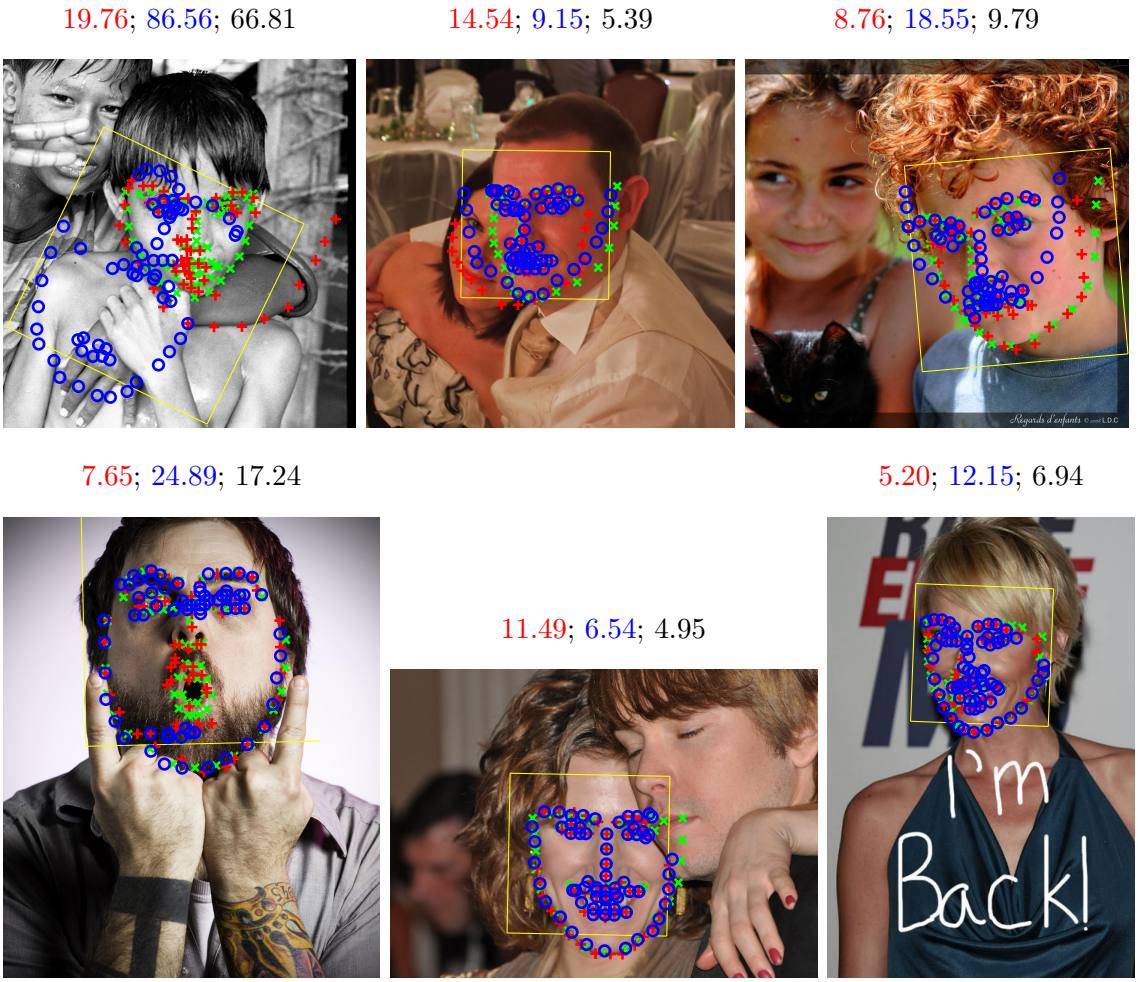


Figure 5.6. The comparison of results provided by the detector trained with the proposed single-view loss and the 0/1 loss on images from the 300-W public test set. The green crosses denote the ground truth landmarks. The red pluses are the landmarks estimated by the detector learned with the single-view loss. The blue circles are landmarks detected by the detector learned with the 0/1 loss. The yellow box is the initialization face box provided by the face detector. The corresponding localization error for both detectors is shown in the title of each example. The last number in the title is the absolute difference of the localization errors of the two detectors.

5.4. Multi-view experiments

In this section, we describe results of multi-view experiments. First, we compare our multi-view detector to the state-of-the-art methods on two datasets, namely the AFLW, and the Multi-PIE. Then, we compare the two approaches to learning of our multi-view detector— the single-stage approach, and the two-stage approach which were described in Sections 3.2.1, and 3.2.2, respectively.

5.4.1. AFLW & Multi-PIE dataset

Multi-view experiments are conducted on the AFLW [Köstinger et al., 2011], and Multi-PIE [Gross et al., 2010] datasets. We use the AFLW dataset for both training and evaluation, and the Multi-PIE dataset for evaluation only. Both datasets come with the annotation of 21 facial landmarks (see Figure 3.3(a)). We used a subset of $12,525$ images from the Multi-PIE, for which we have a precise ground truth annotation. The original AFLW database consists of $24,686$ images. However, the annotation of a large number of images is either inconsistent (confused landmarks) or imprecise. To correct the annotation, we fitted a 3D face model, proposed in [Čech et al., 2014], to the manually annotated landmarks. The projected landmarks of the 3D model were then manually inspected and corrected when necessary. The process reduced the number of images to $21,688$ (mainly due to failures of the face detector involved in the semi-automatic annotation procedure), but it significantly improved the quality of the ground truth annotation.

We randomly selected $\approx 37\%$ of images for training, $\approx 12\%$ for validation and $\approx 51\%$ for testing. The number of training examples is relatively small taking into account the number of model parameters $\dim(\boldsymbol{w}) = 1,335,360$. Surprisingly, the test accuracy of the learned detector is quite high which we attribute to the generalization ability of the SO-SVM algorithm.

5.4.2. Summary results on AFLW & Multi-PIE

This section summarizes a comparison of the proposed landmark detector with competing methods on the AFLW and Multi-PIE datasets. In Figure 5.7, and 5.8, we show the cumulative histograms of multi-view localization error E_{mv} (5.2), evaluated on test images of the AFLW, and the Multi-PIE dataset, respectively. The corresponding $A5$ and $A10$ scores are summarized in Table 5.3. Besides overall statistics computed on all test images in a given dataset, we also evaluate detectors on subsets of test images having the ground truth viewpoint within a specified range. In particular, we define the following subsets:

- **near-frontal** images with the yaw angle $\phi \in (-15°, 15°)$
- **non-profile** images with the yaw angle $\phi \in (-60°, -15°) \cup (15°, 60°)$
- **profile** images with the yaw angle $\phi \in (-110°, -60°) \cup (60°, 110°)$

The most of the detectors are not designed or trained to operate in the full range of the viewpoint, and thus they fail on profile images. For this reason, we show results on profile subset only for the proposed detector, the baseline independent DPM detector, and the detector of Zhu and Ramanan [2012], all operating in the full range of yaw angles. In Table 5.4, we show the viewpoint prediction error E_{yaw} and face detector error E_{fd} on non-profile images. Figure 5.9 depicts exemplary outputs of the proposed detector on a sample of test images from the AFLW dataset. We show both the examples with a small localization error, $E_{\text{mv}} \approx 5\%$, and the highest error, $E_{\text{mv}} = \infty$, that is, the images on which the viewpoint estimation failed.

Table 5.3. The $A5$ and $A10$ scores obtained on the *non-profile* testing subsets of the AFLW, and Multi-PIE datasets. The $A5$ and $A10$ score is a percentage of test images with the multi-view localization error not higher than 5%, and 10% of the face height, respectively.

	AFLW		Multi-PIE	
method	$A5$	$A10$	$A5$	$A10$
proposed MV single stage detector	44.17%	91.44%	58.79%	**77.06**%
proposed MV two-stage detector	49.58%	**92.18**%	65.07%	76.08%
independent DPM detectors	44.72%	91.30%	46.36%	55.22%
Zhu and Ramanan [2012] (p99)	10.98%	48.19%	47.59%	66.47%
IntraFace [Xiong and la Torre, 2013]	**57.80**%	76.63%	68.92%	71.24%
GN-DPM [Tzimiropoulos and Pantic, 2014]	31.06%	50.02%	30.51%	34.69%
Chehra [Asthana et al., 2014]	41.45%	62.29%	**69.07**%	73.92%
Kazemi and Sullivan [2014] (dlib C++)	49.58%	75.60%	55.63%	62.20%

Table 5.4. The viewpoint prediction error E_{yaw} and face detector error E_{fd} evaluated on *non-profile* testing examples from AFLW, and Multi-PIE datasets, respectively.

dataset / method	AFLW		Multi-PIE	
	E_{fd}	E_{yaw}	E_{fd}	E_{yaw}
proposed MV single-stage detector	**0.00** %	23.61 %	**0.06** %	**22.21** %
proposed MV two-stage detector	**0.00** %	**23.53** %	**0.06** %	23.37 %
independent DPM detectors	**0.00** %	30.34 %	**0.06** %	44.04 %
Zhu & Ramanan [Zhu and Ramanan, 2012]	35.60 %	56.47 %	0.08 %	33.29 %
CHEHRA [Asthana et al., 2014]	25.30 %	40.52 %	19.94 %	25.34 %
IntraFace [Xiong and la Torre, 2013]	19.10 %	32.76 %	11.94 %	27.96 %
Kazemi & Sullivan [Kazemi and Sullivan, 2014]	20.57 %	31.80 %	10.63 %	32.80 %
GN-DPM [Tzimiropoulos and Pantic, 2014]	19.05 %	48.09 %	0.09 %	62.98 %

Based on the empirical results we can draw the following conclusions:

- The results demonstrate that both variants of the proposed detector, i.e. the single-stage and the two-stage approach (they are compared in details later in Section 5.4.3), have consistently good localization accuracy in all viewpoints.

- The evaluation on the near-frontal, and non-profile faces shows that proposed detectors have the smallest multi-view localization error. More precisely, proposed detectors, as well as the IntraFace, detector of [Kazemi and Sullivan, 2014], and Chehra provide roughly the same amount of test faces with a localization error not higher than 5% of the face height. However, the proposed detector significantly dominates the others in the regime with still tolerable localization error of 7.5 to 10% of the face height. This behavior is consistent over both datasets.

- On the profile images, the proposed detector significantly outperforms the only fully multi-view competitor, the detector of Zhu and Ramanan [2012]. Both variants of the proposed detector are also consistently better than the baseline composed of the independent DPM detectors, demonstrating the benefits of using the structured classifier over the independent estimate of the viewpoint and landmark locations.

- The proposed detector has a significantly smaller viewpoint prediction error E_{yaw}, and face detector error E_{fd} on both datasets than the rest of competing methods. A small face detector error E_{fd} is a result of using the state-of-the-art commercial face detector. SA small E_{yaw} can be attributed to the multi-view loss function (4.11) used, thanks to which the learning algorithm explicitly minimizes the probability

of confusing the viewpoint. In contrary, the detector of [Zhu and Ramanan, 2012] learned with the 0/1-loss achieves the highest E_{yaw} among the competing methods.

5.4.3. Comparison of single-stage and two-stage approach

In this section, we compare two approaches to learning of the proposed multi-view facial landmark detector, which were described in Sections 3.2.1, and 3.2.2.

The main goal of this experiment is to show, that the two-stage approach can dramatically lower the training time while retaining the landmark localization and viewpoint estimation accuracy.

The accuracy curves from testing parts of the AFLW, and Multi-PIE datasets obtained using both approaches are depicted in Figures 5.7, and 5.8, respectively. The proposed detector learned by the two-stage approach is consistently outperforming the single-stage approach. This is not surprising since, in contrast to the single-stage, the detector learned by the two-stage approach has two sets of parameters. The first set is dedicated to the landmark localization, while the second one to the viewpoint estimation. In contrast, the detector trained by the single-stage has a single set of parameters for both tasks.

The entire learning procedure composed of tuning the regularization constant λ took around 14 days on a machine with 12 cores CPU for learning the detector by the single-stage approach. In contrast, the two-stage approach required less than 2 days for the two-stage approach with just 8 cores CPU. 1 day was spent on learning the individual detectors, i.e. the first stage, and less than a day took the learning of the second stage, i.e. the viewpoint detector. The execution time of both variants of the multi-view facial landmark detector is practically identical.

See Figures 5.7, 5.8, and Tables 5.3, 5.4 for a detailed quantitative comparison of both variants of the proposed multi-view detector.

5.4.4. Limitations of the evaluation protocol

In the case of the near-frontal, single-view landmark detection problem, there exist established benchmarks and evaluation metrics, e.g. the 300-W dataset and the evaluation protocol defined in [Sagonas et al., 2016]. There is no such benchmark for evaluation of full multi-view detectors. The evaluation protocol used in our experiments is an attempt to solve the issue. However, the protocol has its limitations. In particular, we have identified the following problems:

- The evaluated methods use different face detectors which are often an integral part of the landmark detector. This prevents to evaluate only the ability to localize landmarks. Our multi-view localization error E_{mv} penalizes face detector failures by inf penalty. In addition, we also measure face detector failures independently by the face detector error E_{fd}.

- Because there is no established dataset, competing methods are trained on different examples, or at least a different subset of the same benchmark. It is clear that the quality and the extent of training examples used have a significant impact on results.

- The viewpoint influences the set of visible landmarks. Therefore, the viewpoint estimate has to be included in the evaluation metric. Our definition of the multi-view localization error E_{mv} penalizes the yaw angle misclassifications by inf penalty which is a conservative option. Also, we measure the viewpoint prediction error independently using the metric E_{yaw}.

Table 5.5. The average time required by competing methods to process a single face. We show the average time, and the standard deviation in *seconds*. The results are computed separately for each dataset. The "proposed" stands for the multi-view detector on the AFLW (first column), and the Multi-PIE (second column) dataset, and C2F-DPM detector on the 300-W dataset (third column), respectively.

method \ dataset	AFLW	Multi-PIE	300-W
proposed (multi-view/C2F-DPM) detector	0.011 ± 0.005	0.012 ± 0.002	0.1 ± 0.02
independent detectors	0.003 ± 0.001	0.004 ± 0.001	—
Zhu & Ramanan [Zhu and Ramanan, 2012]	60.4 ± 24.0	18.9 ± 11.4	73.9 ± 144.4
Chehra [Asthana et al., 2014]	0.1 ± 0.08	0.2 ± 3.4	0.2 ± 2.6
IntraFace [Xiong and la Torre, 2013]	0.05 ± 0.1	0.03 ± 0.01	0.1 ± 0.2
Kazemi & Sullivan [Kazemi and Sullivan, 2014]	0.4 ± 0.4	0.4 ± 0.3	0.248 ± 0.363
GN-DPM [Tzimiropoulos and Pantic, 2014]	0.8 ± 0.4	0.5 ± 0.1	0.6 ± 1.8

- The evaluated methods estimate different sets of landmarks. In our evaluation, we use a subset that is common to all methods while the other landmarks are being ignored.

5.5. Evaluation of the processing time

In this section, we evaluate the CPU time required by individual landmark detectors to process a single image. Table 5.5 presents the results. The reported time is an average measured on a set of cropped images containing a single face only, to decrease the time spent on the face detector which is an integral part of the methods [Xiong and la Torre, 2013; Zhu and Ramanan, 2012; Kazemi and Sullivan, 2014]. We do not count initialization time, and if possible, we subtract the face detector time (e.g. for [Kazemi and Sullivan, 2014]).

The fastest among the compared approaches is the independent DPM detector using an external method for the viewpoint estimate. Otherwise, the proposed multi-view DPM detector is consistently significantly faster than the rest of methods on both AFLW and the Multi-PIE datasets. The achieved speedup is an order of magnitude at least.

The processing time required by individual stages of the proposed detector is detailed in Table 5.6. It is seen, that computations are dominated by feature evaluation, which depends on the resolution of the normalized frame and the size of search spaces. On the other hand, the max-sum inference (c.f. Section 3.5) takes less than 20% of the overall time, thanks to the distance transform described in Section 3.5.2. To demonstrate the benefit of a distance transform, we also present the time required by the max-sum inference solved by a plain dynamic programming.

In Section 3.3, we presented several choices of feature descriptors that can be used to describe the appearance model of landmarks. Here, we compare their processing times. We measure the time needed to compute all descriptors for the root landmark of the F-DPM detector, i.e. the patch of size 21×21 pixels across the whole search space corresponding to this root landmark. We compare the proposed S-LBP descriptor, the SIFT descriptor implemented in the Intraface [Xiong and la Torre, 2013], and the HOG descriptor provided in MATLAB. The S-LBP descriptor requires 1 millisecond, the SIFT 62 ms, and HOG approximately 930 ms.

Table 5.6. Time requirements of individual stages of the proposed detectors. The statistics are shown for the multi-view detector used on the AFLW, and Multi-PIE dataset, and the C-DPM, and F-DPM detectors evaluated on the 300-W datasets. We list the average time and standard deviation achieved on the testing images. The last row shows time needed to compute the max-sum inference without using the distance transform, to emphasize its importance. All times are shown in *milliseconds*.

type / stage	Multi-view detector (AFLW + Multi-PIE)	C-DPM (300-W)	F-DPM (300-W)
normalized frame	0.009 ± 0.003	0.4 ± 0.1	1.4 ± 0.4
feature computation	8.1 ± 4.1	35.5 ± 6.1	64.2 ± 9.0
max-sum inference	2.4 ± 1.9	5.3 ± 0.8	6.4 ± 0.9
overall	10.5 ± 4.7	41.2 ± 6.8	72.0 ± 9.9
max-sum inference without dist. transf.	93.0 ± 1.6 (38×slower)	970.0 ± 30.1 (183×slower)	2167 ± 38.8 (339×slower)

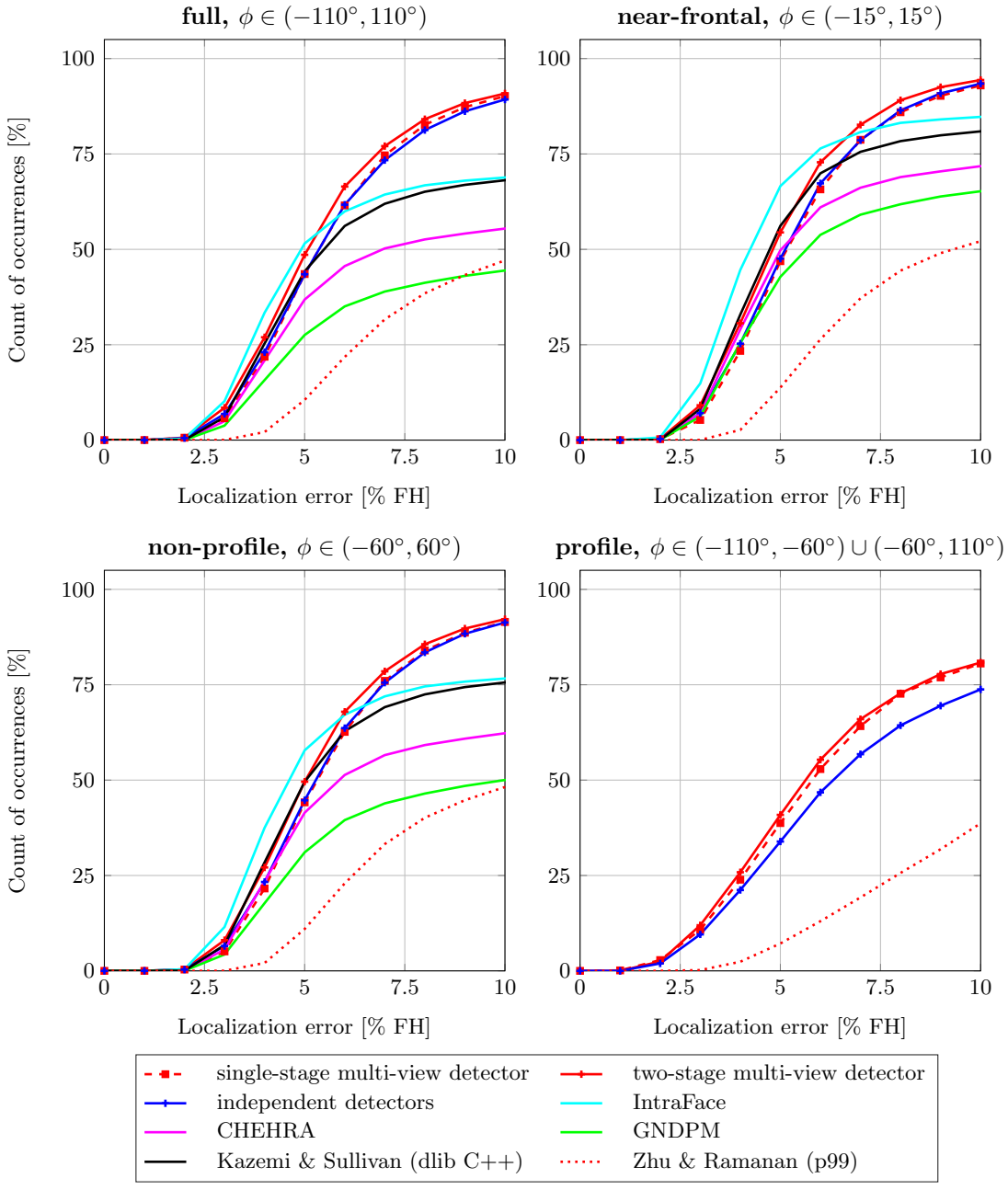


Figure 5.7. Cumulative histograms of the average localization error measured on a testing subset of the AFLW dataset. The localization error is normalized by a face height, computed as a distance between the root of the nose, and the chin. Individual sub-figures contain error measured on a subset of the test images with the ground-truth viewpoint within a corresponding range.

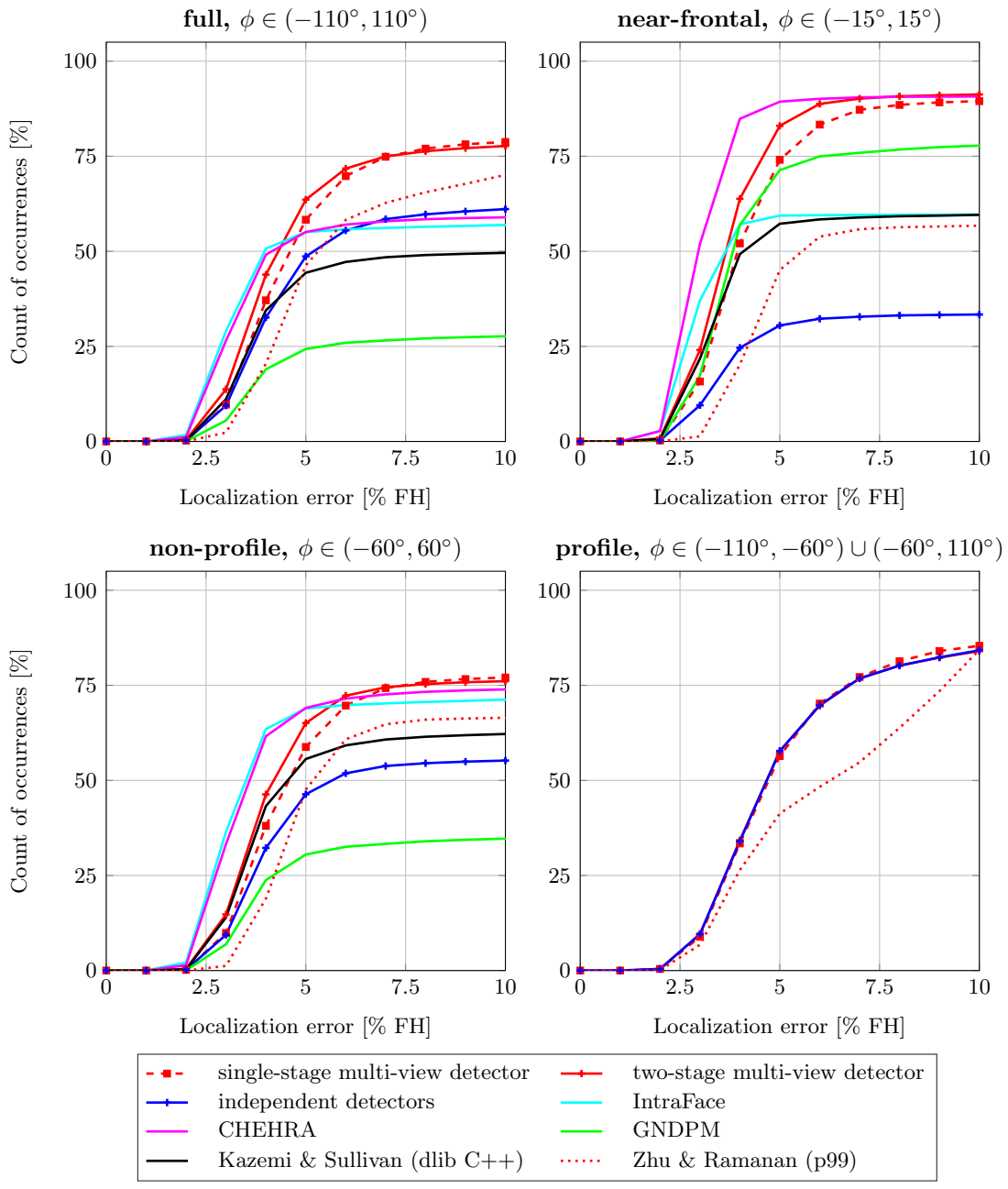


Figure 5.8. Cumulative histograms of the average localization error measured on a testing subset of the Multi-PIE dataset. The localization error is normalized by a face height, computed as a distance between the root of the nose, and the chin. Individual sub-figures contain the error measured on a subset of the test images with the ground-truth viewpoint within a corresponding range.

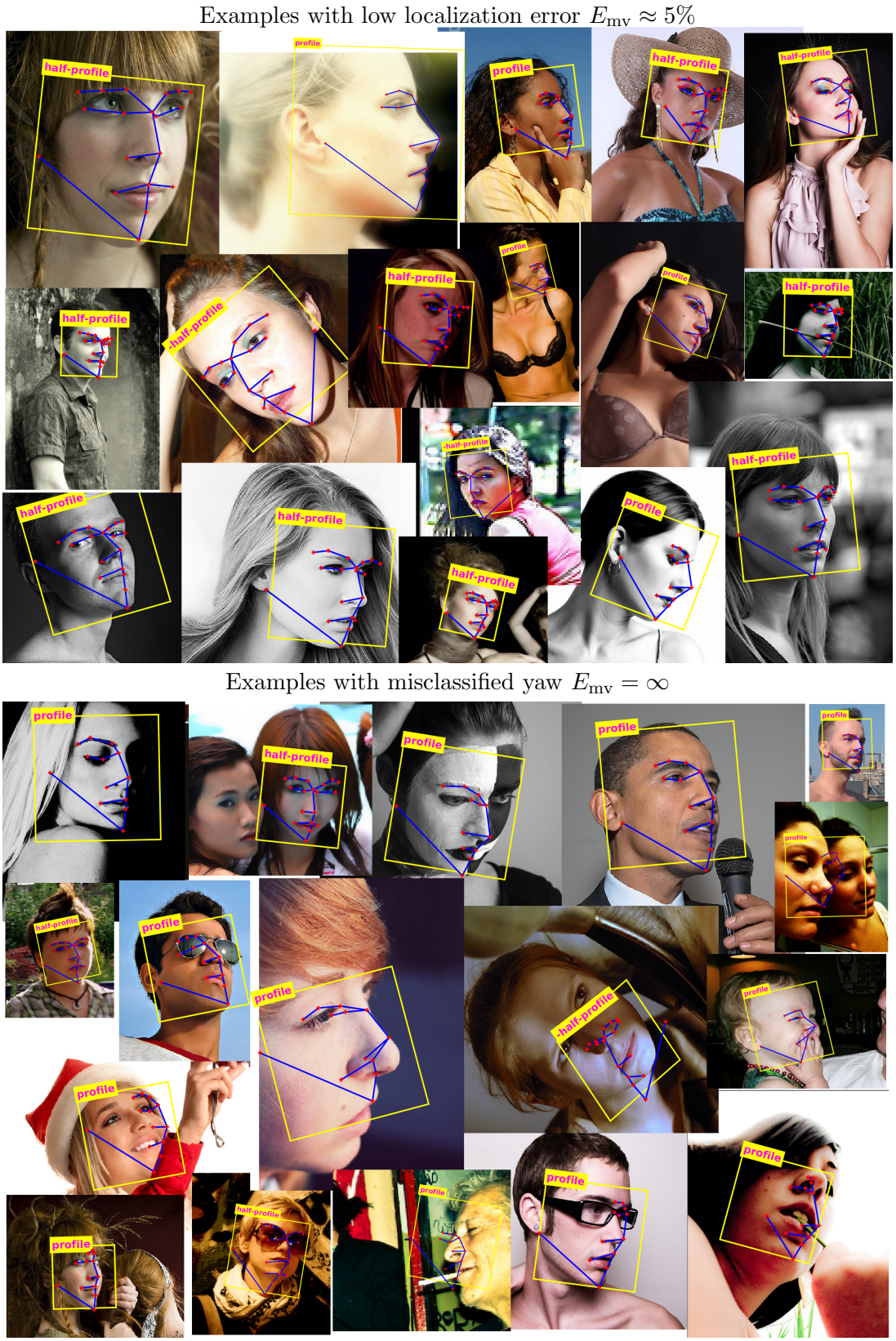


Figure 5.9. Exemplary images from the AFLW testing set with the average localization error not higher than $E_{\text{loc}} \approx 5\%$ (top), and with the misclassified viewpoint $E_{\text{loc}} = \infty$ (bottom). Yellow box represents the face detection as provided by the face detector (i.e. the input of proposed detector), the discretized viewpoint category is written on the top edge of the face box. Red crosses denote landmarks. Blue lines connecting landmarks emphasize the underlying graph corresponding to the viewpoint.

5.6. Evaluation of the improved BMRM solver

The SO-SVM algorithm converts learning of the DPM detector into an instance of a convex optimization problem. We found experimentally that a precise solution of the learning problem is necessary to obtain well-performing landmark detector. We conclude that the approximate online methods like the SGD [Bordes et al., 2009] yield significantly worse solution than the precise batch solvers with a guaranteed accuracy. Therefore, we use the BMRM solver, which provides the required precision but at the expense of a long training time. The training time matters, because it restrict the number of hyperparameters, and design options one can afford to evaluate. To cope with the problem, we have proposed two improvements of the BMRM algorithm with the aim to decrease the training time. The effect of these two improvements is evaluated in this section.

Let us briefly describe our two improvements first. The full description can be found in Chapter 4. The first improvement, which we denote as Prox-BMRM, is based on augmenting the objective function by an additional quadratic prox-term, whose strength is adaptively tuned. The prox-term prevents searching for a consecutive solution too far away from the previous estimate and, consequently, reduces the "zig-zag" behavior of the BMRM solver, which is typical for small values of the regularization constant λ. The second improvement, which we denote as P-BMRM, consists of using P cutting plane models instead of a single one as in the genuine BMRM solver. The multiple cutting plane model better approximates the risk function by which it reduces the number of iterations. Both improvements can be naturally combined, yielding the best results. We denote the variant combining the two improvements as the Prox-P-BMRM solver.

The proposed improvements of the BMRM solver are applicable to an arbitrary instance of the SO-SVM learning, i.e. they are not constrained to landmark detection problem. To demonstrate this, we evaluate the improvements on another two benchmark problems apart from the landmark detection. We describe these benchmark problems in the next section.

5.6.1. Benchmark problems

We consider the following three benchmark problems each of which is an entirely different instance of the SO-SVM classification problem. In all cases, however, we search for a generic linear structured output classifier defined by (4.1). where the particular interpretation of inputs $\boldsymbol{x} \in \mathcal{X}$, predicted outputs $\boldsymbol{y} \in \mathcal{Y}$ and joint feature map $\boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y})$ is described below for individual benchmarks.

Benchmark 1: Optical Character Recognition (OCR) We consider the OCR problem as the first benchmark. We use the MNIST database[4] composed of annotated examples of handwritten numerals. The classifier's input $\boldsymbol{x}$ is a gray scale image 28×28 pixels large. The classifier's output $\boldsymbol{y}$ is a digit label, i.e. $\boldsymbol{y} \in \mathcal{Y} = \{0, \ldots, 9\}$. We model each class by a single template image $\boldsymbol{w}_{\boldsymbol{y}} \in \mathbb{R}^{28 \times 28}$, $\boldsymbol{y} \in \mathcal{Y}$. As the scoring function of the classifier (4.1), we use $\langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle = \langle \boldsymbol{x}, \boldsymbol{w}_{\boldsymbol{y}} \rangle$. The parameter vector $\boldsymbol{w} \in \mathbb{R}^n$ has the dimension $n = 7,840$ resulting from a column-wise concatenation of 10 templates $\boldsymbol{w}_{\boldsymbol{y}}$, $\boldsymbol{y} \in \mathcal{Y}$, represented themselves as column vectors. We use the standard classification 0/1-loss defined as $\ell(\boldsymbol{y}, \boldsymbol{y}') = 1$ for $\boldsymbol{y} \neq \boldsymbol{y}'$ and $\ell(\boldsymbol{y}, \boldsymbol{y}') = 0$, otherwise. With these definitions, the classifier (4.1) becomes an instance of a linear multi-class SVM classifier.

We train on all $m = 60,000$ training examples, and we use the test part of the database for validation, and testing (5,000 examples each).

[4] `http://yann.lecun.com/exdb/mnist/`

Benchmark 2: Facial landmark detection We consider learning of a facial landmark detector as the second benchmark. To be more specific, we use the L8 variant of the single-view detector, with the same settings as defined in Section 5.1.1. We train the L8 landmark detector on the LFW dataset, i.e. on a set of $m = 6,919$ images with manually annotated landmark positions. The dimensionality of the joint parameter vector $\boldsymbol{w} \in \mathbb{R}^n$ is $n = 232,476$ in this experiment.

Benchmark 3: License plate segmentation We consider a segmentation of a car license plate images as the third benchmark. The classifier's input $\boldsymbol{x} \in \mathcal{X}$ is an image $H \times W$ pixels large, which contains a license plate, i.e. a line of text composed of a known set of characters. The columns of input image $\boldsymbol{x}$ are features extracted from the intensity values of a corresponding column of a raw image taken by a camera. The classifier outputs image segmentation $\boldsymbol{y} = (s_1, \ldots, s_L) \in \mathcal{Y}$, where $s = (a, k)$, $a \in A$ is a character code and $k \in \{1, \ldots, W\}$ is a character position. An admissible segmentation $\boldsymbol{y} \in \mathcal{Y}$ must satisfy

$$\left.\begin{array}{l} k(s_1) = 1\,, W = k(s_L) + \omega(s_L) - 1\,, \\ k(s_i) = k(s_{i-1}) + \omega(s_{i-1})\,, \forall i > 1\,, \end{array}\right\} \tag{5.5}$$

where $\omega\colon A \to \mathbb{N}$ are the widths of characters. The constraints (5.5) guarantee, that the segmentation $\boldsymbol{y}$ covers the whole image $\boldsymbol{x}$ by a sequence of characters $a_1, \ldots, a_L$, which do not overlap. Each character $a \in A$ is modeled by a template image $\nu_a \in \mathbb{R}^{H \times \omega(a)}$. The parameter vector $\boldsymbol{w} \in \mathbb{R}^n$ to learn is a column-wise concatenation of all templates ν_a, $a \in A$. The scoring function of the classifier (4.1) computes the correlation between image $\boldsymbol{x}$ and character templates placed one by one according to segmentation $\boldsymbol{y} \in Y$, i.e. $\langle \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{w} \rangle$ equals to

$$\sum_{i=1}^{L(y)} \sum_{j=1}^{\omega(a(s_i))} \langle \mathrm{col}(\boldsymbol{x}, j + k(s_i) - 1), \mathrm{col}(\boldsymbol{w}_{a(s_i)}, j) \rangle\,, \tag{5.6}$$

where $\mathrm{col}(I, i)$ denotes the i-th column of image I. We use the Hamming distance as the loss function to measure the number of incorrectly segmented columns w.r.t. to the annotated segmentation. The evaluation of the classifier (4.1), as well as the evaluation of $r_i(\boldsymbol{w})$, and its sub-gradient $\boldsymbol{r}_i(\boldsymbol{w})$, leads to an instance of the DP.

We use $m = 6,788$ annotated images for training, $1,692$ for computing the validation error, and $1,692$ images for testing [5]. The parameter vector $\boldsymbol{w}$ has $n = 4,059$ components.

5.6.2. Evaluation of the proposed P-BMRM algorithm

In this experiment, we compare the genuine BMRM solver to the proposed P-BMRM, and the online SGD algorithm. For BMRM and P-BMRM algorithm we measure the number of iterations needed to achieve the ϵ-precise solution

$$\frac{F(\boldsymbol{w}_t) - F_t(\boldsymbol{w}_t)}{F(\boldsymbol{w}_t)} \leq 0.01\,, \tag{5.7}$$

where $F(\boldsymbol{w}_t)$ is the original objective value and $F_t(\boldsymbol{w}_t)$ is the reduced objective value given by the CPM. When the stopping condition is satisfied, it implies that $F(\boldsymbol{w}_t) \leq F(\boldsymbol{w}^*)/0.99 \approx 1.01 F(\boldsymbol{w}^*)$. The stopping condition (5.7) cannot be used in the SGD algorithm because SGD does not provide the lower bound on the optimal value. To allow a fair comparison, we first run the BMRM algorithm, and then, we use the value of

[5] The data were provided by the courtesy of Eyedea recognition `http://www.eyedea.cz`.

the objective function in the last iteration as a stopping criterion for the SGD. In particular, we iterate the SGD, until the value of the objective function $F(\boldsymbol{w}_t)$ gets below the value returned by the BMRM algorithm when it was stopped based on (5.7). The reason for using the number of iterations as a measure of the computational complexity instead of the wall-clock time is two-fold. First, the per-iteration computational complexity of all tested algorithms is similar (in the case of the SGD algorithm, we consider one pass through all the training examples as one iteration), because it is dominated by calling classification oracle to evaluate the risk and its sub-gradient. Second, the number of iterations does not depend on the implementation and the hardware used.

Table 5.7 shows the number of iterations required to achieve an ϵ-optimal solution for different values of λ on Benchmarks 1, and 2. The table shows the results for the SGD and the P-BMRM for varying number P. Note, that standard BMRM is equivalent to P=1-BMRM using a single CPM. The tables also show a speedup achieved by processing individual CPMs in parallel measured relatively to the standard BMRM algorithm running on a single CPU. The standard BMRM can be naturally parallelized as well by decomposing the computation of the risk and its sub-gradient on P computers (CPUs) equally. However, the speedup of the standard parallelized variant of the BMRM is just P. The results show the following facts:

- The number of iterations of the P-BMRM decreases with increasing P. This is an expected result because a higher P implies a finer approximation of the objective. It should be mentioned that increasing P leads to a proportional increase of the memory requirements, caused by maintaining P CPMs instead of a single one.

- The P-BMRM algorithm is more efficient for low values of λ, i.e. for the most time-consuming instances of the learning problem (4.5). For example, the $(P = 64)$-BMRM requires 11 times less iterations, compared to the standard BMRM on Benchmark 1 when we use $\lambda = 10$. Using the parallel P-BMRM then leads to the speedup factor of 700.

The relative distance to the optimal value as a function of the iteration number is shown in Figures 5.10, and 5.11 for different values of λ, respectively.

In Figure 5.12, we also show the convergence of the objective function $F(\boldsymbol{w})$ and the value of the validation risk for the BMRM algorithm, and SGD algorithm. The convergence curves are given for Benchmark 2 (landmark detection), and regularization constant $\lambda = 100$. It is seen, that SGD reaches a relatively good precision of the objective function after a few iterations, but then it stalls. In contrast, the BMRM fluctuates at the beginning, but at the end, it manages to reach the ϵ-precise solution. Also, a more precise solution regarding the objective function $F(\boldsymbol{w})$ is translated to a lower validation risk. In particular, the precise BMRM algorithm outperforms the online SGD algorithm by a significant 1.5% regarding detection accuracy.

5.6.3. Evaluation of the proposed Prox-BMRM algorithm

In this section, we compare the standard BMRM, the Prox-BMRM algorithm implementing the second proposed improvement, and the Prox-P-BMRM algorithm combining both improvements. The proposed algorithms using prox-term are started from a non-trivial initial solution. Note, that such initialization is pointless in the case of the standard BMRM, which does not prevent consecutive solutions from jumping arbitrarily far from the previous one. We experiment with two different strategies to compute the initial solution. In the case of Benchmark 1, we start the algorithm from the solution obtained in a model selection for the previous value of the regularization constant λ. The run for the highest value of λ is started from $\boldsymbol{w} = \mathbf{0}$. In the evaluation

Table 5.7. The table shows the number of iterations needed to reach ϵ-optimal solution and the speedup of parallelized version of the algorithm, measured regarding a number of iterations required to converge. Each row corresponds to a different setting of a number of CPMs. The last row shows results for the SGD algorithm. Symbol "—" means, that the SGD did not converge.

Facial landmark detection — L8 on LFW database

	$\lambda = 10000$		$\lambda = 1000$		$\lambda = 100$		$\lambda = 10$	
	#iter	speedup	#iter	speedup	#iter	speedup	#iter	speedup
P = 1 BMRM	104	1	172	1	390	1	999	1
P = 8 BMRM	88	9.5	171	8.0	350	8.9	858	9.3
P = 16 BMRM	85	19.6	144	19.1	307	20.3	733	21.8
P = 32 BMRM	75	44.4	123	44.7	276	45.2	618	51.7
SGD	1	104	1	171	—	—	—	—

OCR — MNIST database

	$\lambda = 1000$		$\lambda = 100$		$\lambda = 10$		$\lambda = 1$	
	#iter	speedup	#iter	speedup	#iter	speedup	#iter	speedup
$P = 1$ BMRM	163	1	431	1	1384	1	5159	1
$P = 8$ BMRM	123	10.6	271	12.7	737	15.0	2132	19.4
$P = 16$ BMRM	98	26.6	208	33.2	512	43.3	1346	61.3
$P = 32$ BMRM	79	66.0	155	89.0	332	133.4	810	203.8
$P = 64$ BMRM	60	173.9	106	260.2	218	406.3	470	702.5

of Benchmark 2, and 3, the initial solution is obtained by performing 10 iterations of the SGD [Bordes et al., 2009].

We learn the SO-SVM classifiers on all 3 benchmarks for different values of the regularization parameter λ. All evaluated algorithms use the same stopping condition (5.7). For each algorithm, we report the number of iterations and a wall clock time. All algorithms are implemented in the same framework using the same implementation of the time-consuming procedures, which justifies the usage of a wall clock time. The obtained results are summarized in Table 5.8. Based on the results we can observe that:

- The Prox-BMRM significantly decreases both number of iterations, and a wall clock time, compared to the original BMRM. As expected, the speedup is higher for the lower values of λ when the standard quadratic regularization term has a small influence.

- The speedup is further improved by Prox-P-BMRM algorithm, which in addition to the prox-term also uses a multiple CPMs with $P = 16$ components. The improvement due to using more CPMs is well seen on Benchmark 3 (license plates), where the vector of parameters has relatively small dimension. Using more CPMs is less beneficial for Benchmark 2 (facial landmarks), where data are very high-dimensional and sparse. The maximal speedup 9.7 was obtained on Benchmark 1 (OCR — MNIST) for the smallest value of λ and Prox-16-BMRM.

To show the effect of the introduced prox-term on the convergence, we plot a relative distance to the optimal value as a function of the iteration number in Figure 5.13. It is seen, that convergence curve for the Prox-BMRM is much smoother, compared to the standard BMRM, whose curves fluctuate strongly due to the "zig-zag" behavior.

Because all solvers use the same stopping condition enforcing a high precision solution, the learned classifiers have practically the same classification accuracy. Hence, we report the validation errors and final test error only once in Table 5.9 for the sake of completeness.

Table 5.8. The number of iterations, wall clock time in hours, and the obtained speedups for all benchmark problems and all tested algorithms runs with different values of the regularization constant λ.

Benchmark 1: OCR — MNIST

Initial solution	$\lambda_1 = 1000$			$\lambda_2 = 100$			$\lambda_3 = 10$			$\lambda_4 = 1$		
REUSE	#iter	time	spdup	#iter	time	spdup	#iter	time	spdup	#iter	time	spdup
BMRM	157	0.049	1	417	0.128	1	1429	0.452	1	5932	2.018	1
Prox-BMRM	226	0.069	0.7	232	0.072	1.8	317	0.099	4.6	698	0.265	7.6
Prox-P=16-BMRM	244	0.078	0.6	256	0.086	1.5	291	0.107	4.2	408	0.209	9.7

Benchmark 2: Facial landmark detection

Initial solution	$\lambda_1 = 10000$			$\lambda_2 = 1000$			$\lambda_3 = 100$			$\lambda_4 = 10$		
SGD	#iter	time	spdup	#iter	time	spdup	#iter	time	spdup	#iter	time	spdup
BMRM	108	2.492	1	207	8.263	1	442	9.798	1	1084	47.767	1
Prox-BMRM	28	0.693	3.6	61	1.539	5.3	180	4.654	2.1	783	19.772	2.4
Prox-P=16-BMRM	32	0.783	3.2	55	1.301	6.3	142	3.365	2.9	555	14.411	3.3

Benchmark 3: License plate recognition

Initial solution	$\lambda_1 = 10^5$			$\lambda_2 = 10^4$			$\lambda_3 = 10^3$			$\lambda_4 = 10^2$		
SGD	#iter	time	spdup	#iter	time	spdup	#iter	time	spdup	#iter	time	spdup
BMRM	33	0.605	1	87	1.581	1	251	3.923	1	840	13.726	1
Prox-BMRM	34	0.631	1.0	67	1.236	1.3	126	2.065	1.9	286	4.665	2.9
Prox-P=16-BMRM	22	0.404	1.5	37	0.674	2.3	64	0.981	4.0	131	2.444	5.6

Table 5.9. The table shows validation risks for all benchmarks as a function of regularization constant λ. The test risk is computed for a classifier with the minimal validation risk. These results apply to all evaluated algorithms due the high precision solution enforced by the stopping condition used.

Benchmark 1: OCR — MNIST

	$\lambda = 10^3$	$\lambda = 10^2$	$\lambda = 10^1$	$\lambda = 10^0$	tst
val	0.0864	0.0740	0.0708	0.0720	0.0704

Benchmark 2: Facial landmark detection

	$\lambda = 10^4$	$\lambda = 10^3$	$\lambda = 10^2$	$\lambda = 10^1$	tst
val	11.03	6.36	5.46	5.80	5.46

Benchmark 3: License plate segmentation

	$\lambda = 10^5$	$\lambda = 10^4$	$\lambda = 10^3$	$\lambda = 10^2$	tst
val	22.22	13.16	7.02	4.61	4.21

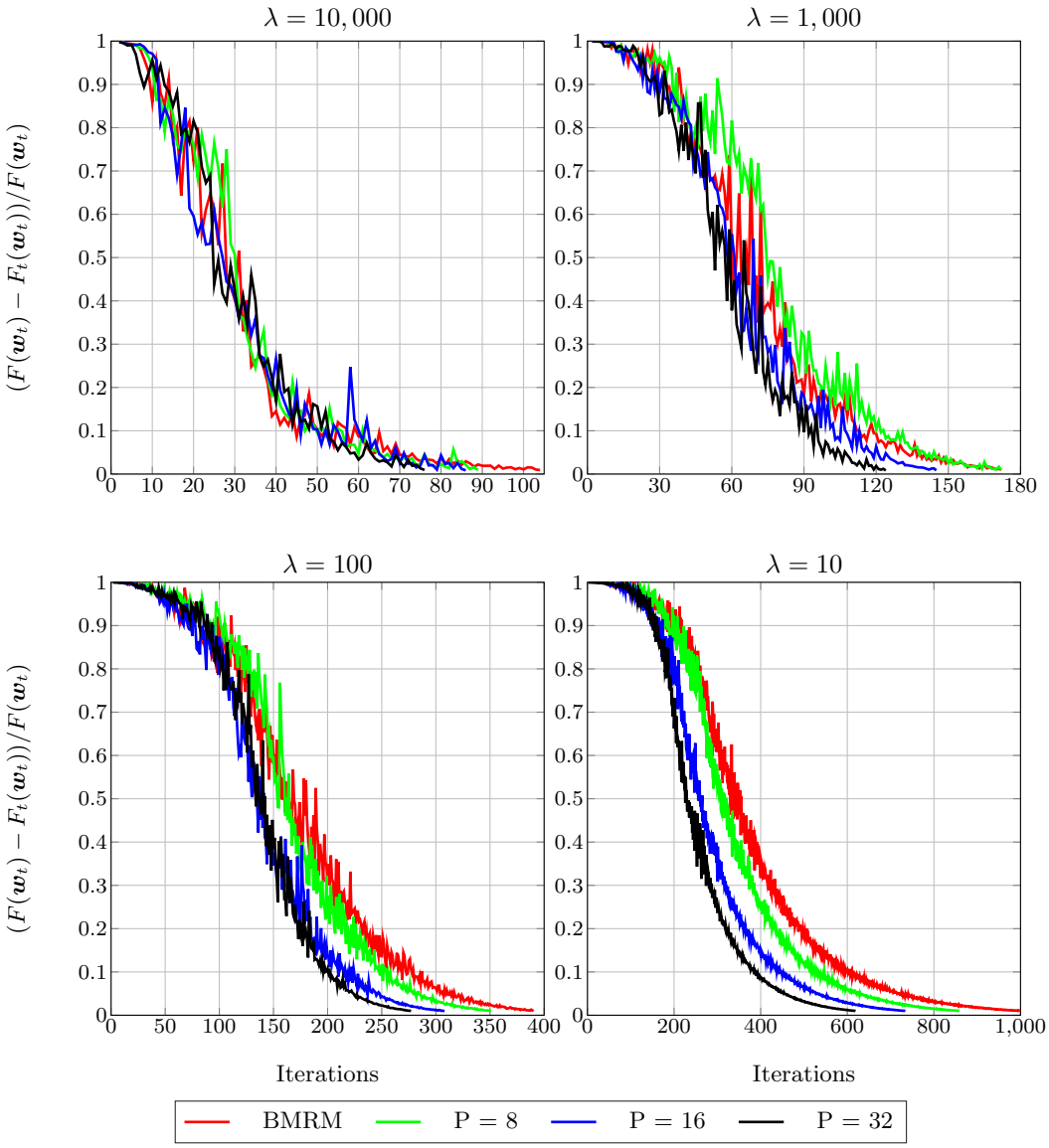


Figure 5.10. Convergence curves of the standard BMRM (i.e. P=1-BMRM), and P-BMRM algorithm applied with different setting of the regularization constant λ to Benchmark 2, the L8 face landmark detection problem. The curves show relative distance to the optimal solution as the function of the number of iterations.

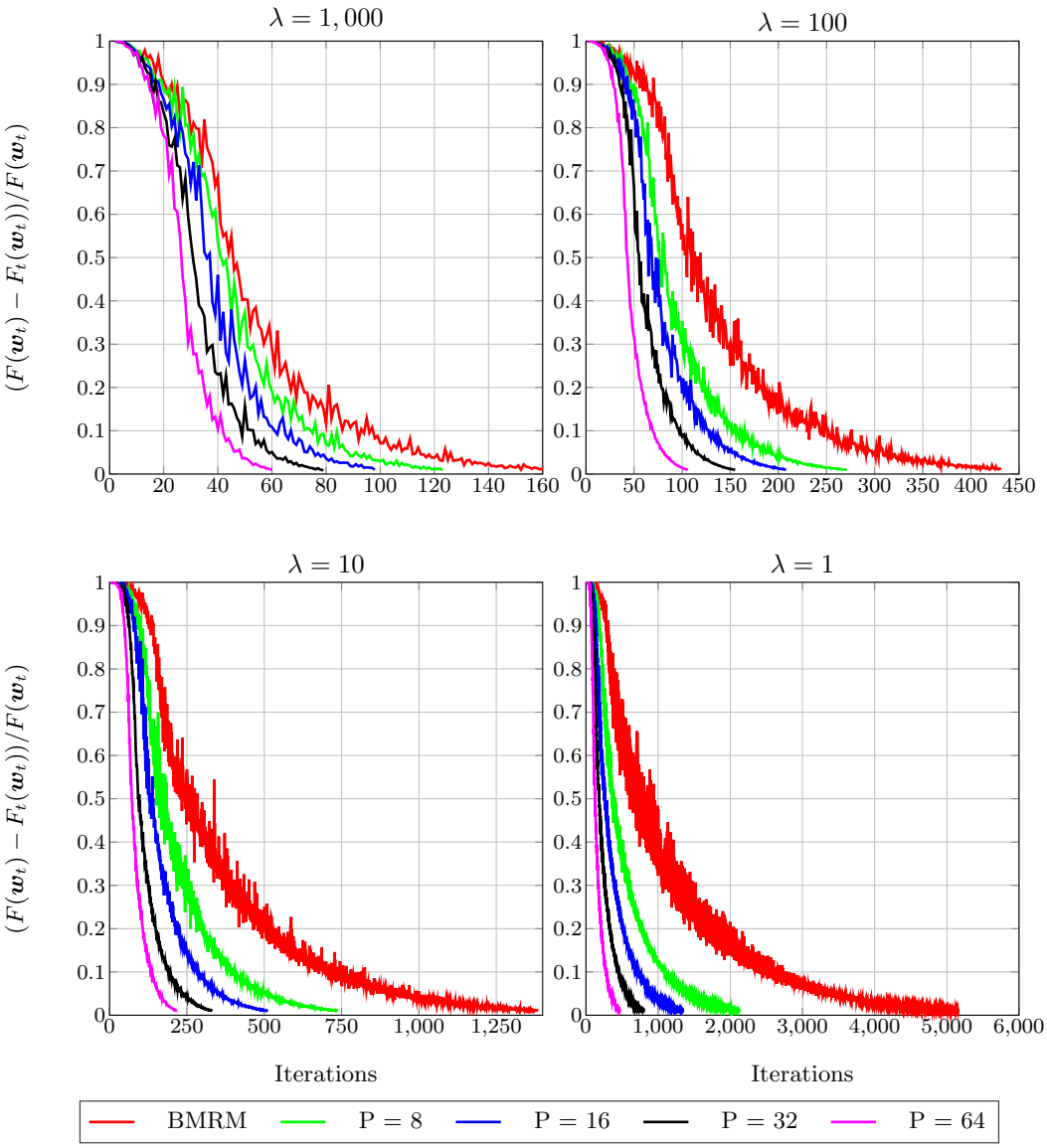
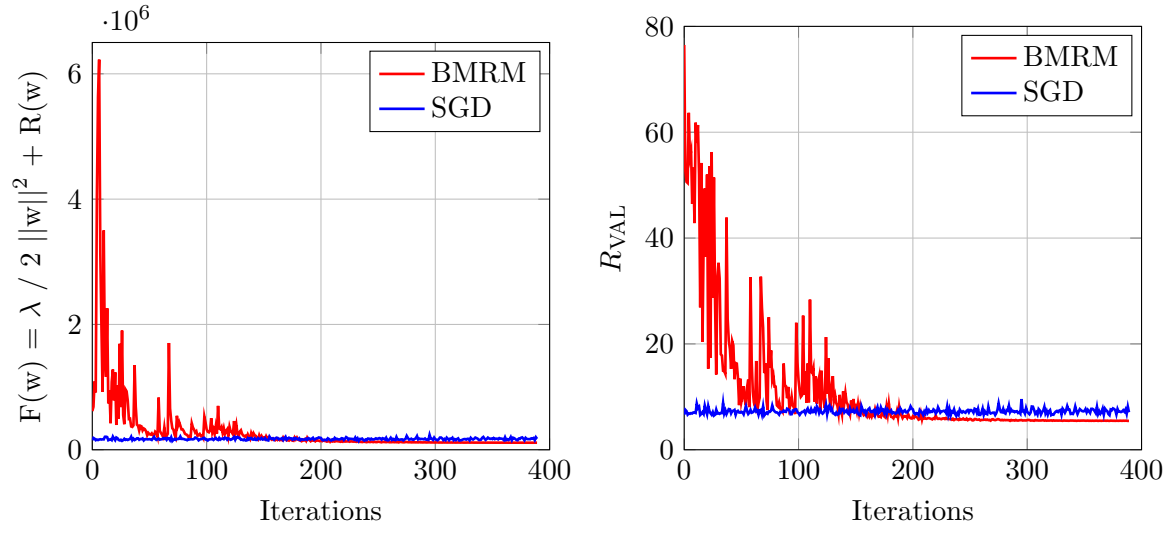
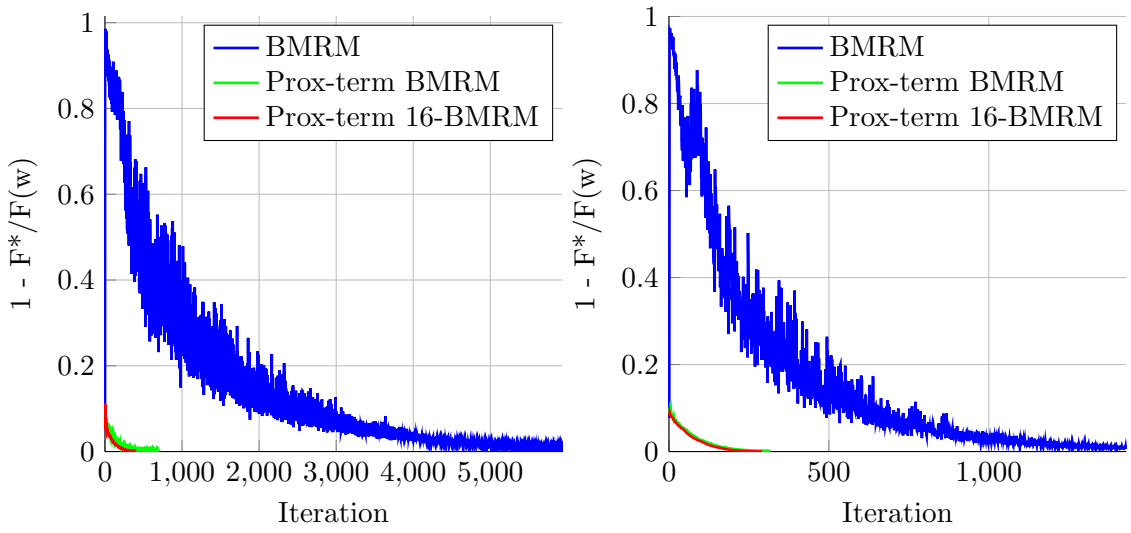


Figure 5.11. Convergence curves of the standard BMRM (i.e. P=1-BMRM), and P-BMRM algorithm applied with a different setting of the regularization constant λ to Benchmark 1, the OCR benchmark problem. The curves show relative distance to the optimal solution as the function of the number of iterations.



(a) Objective function **(b)** Validation risk

Figure 5.12. The convergence of (a) objective value, and (b) validation risk for the BMRM, and the SGD algorithm on Benchmark 2 (landmark detection), using the optimal setting of the regularization constant $\lambda = 100$.



(a) $\lambda = 1$ **(b)** $\lambda = 10$

Figure 5.13. Relative distance to the optimal value as a function of the iteration number shown for the standard BMRM and the proposed Prox-BMRM algorithm. The results are obtained on Benchmark 1 (OCR — MNIST) for two different regularization constants.

6. Conclusions

In this thesis, we tackled the problem of learning a real-time multi-view detector of facial landmarks. We focused on landmark detectors based on the DPM whose prediction problem leads to a combinatorial search, solvable efficiently by global optimization methods.

We showed that the parameters of the DPM detectors could be learned efficiently from examples by the SO-SVM framework. The objective function of the proposed instance of the SO-SVM algorithm is tightly connected to the performance measure of the learned detector. We show experimentally, that the optimization of the actual performance measure surrogate increases the landmark localization accuracy, compared to existing methods which optimize much simpler loss functions. We have also proposed a new two-stage strategy for learning the multi-view DPM detectors. The two-stage learning splits the optimization of landmark localization accuracy and the viewpoint prediction accuracy into two independent steps, where the latter step uses the results of the former one. We have shown experimentally that the two-stage learning strategy, compared to the single-stage approach, increases prediction accuracy and significantly reduces the learning time.

We have shown that the globally optimal prediction strategy of the DPM detector can be evaluated in real-time even for a dense landmark sets. The short prediction time is achieved mainly by two algorithmic improvements. First, we reduce the number of evaluations of the base LBP features by organizing their computation into the MIPMAP. Second, we speed up the combinatorial prediction strategy by using a coarse-to-fine search. In the first stage, a coarse detector operating on low-resolution images is used. In the second stage, landmarks detected by the coarse detector are used to restrict the search space of the fine detector operating on a higher resolution image and detecting the required number of landmarks.

We have proposed two improvements of the BMRM algorithm, being one of the most frequently used SO-SVM solvers nowadays. First, we propose to improve the approximation of the optimized objective function by using a multiple cutting plane models, in contrast to the standard BMRM, which use only a single model. Second, we propose to augment the objective function by an additional quadratic proxy term, whose strength is changed adaptively in the course of optimization. The additional proxy term helps to mitigate the "zig-zag" behavior of the standard BMRM, decreasing the number of iterations in turn. The experimental evaluation shows that the new BMRM algorithm, which uses both of the proposed improvements, speeds up the learning up to an order of magnitude on a standard computer vision benchmarks, and 3 to 4 times when applied to the learning of the DPM landmark detector.

We have released an open-source library **CLandmark**[1] which encapsulate all of this thesis contributions, providing an efficient implementation of proposed (multi-view) detectors, as well as the learning algorithms. The library has been already acknowledged by numerous projects, such as the winner of the "ChaLearn Looking at People Challenge 2016" Appearant Age Estimation track [Antipov et al., 2016], or the "OpenTrack" project (`https://github.com/opentrack/opentrack`), to mention a few.

[1] `https://cmp.felk.cvut.cz/~uricamic/clandmark`

A list of possible topics for future research is as follows:

Modeling occlusions. The proposed multi-view DPM detector deals with self-occlusions only. The other occlusions, caused by other objects like hands, non-transparent glasses, or hairs, are not modeled explicitly. A straightforward solution would be to learn an additional classifier for each landmark that would predict its visibility, based on a patch cropped around the predicted landmark position. A more sophisticated solution is to incorporate the prediction of the landmark visibility into the structured output classifier. The biggest obstacle for modeling occlusions is the lack of a sufficiently large set of annotated examples containing information about the landmark visibility.

Prediction confidence. The proposed DPM detector does not provide confidence for the predicted landmark configurations. The prediction confidence is important functionality, for example, in applications which combine the landmark detector with other processing blocks. One approach would be to calibrate the scoring function of the DPM detector, to serve as a confidence measure, e.g. by fitting a logistic function alike to the probabilistic output SVMs [Platt, 2000]. Another option is to learn an independent classifier, which would assess the correctness of the landmark prediction based on the quality of the input image and the prediction itself. An alternative option is to incorporate the confidence prediction as an additional hidden state to be predicted by the structured output classifier.

Deep learning. The recent overwhelming success of deep learning applied to various computer vision problems has not avoided the landmark detection. The deep learning has been applied to facial landmark detection problem, e.g. in [Sun et al., 2013; Lai et al., 2015; Zhang et al., 2014b,a; Yu et al., 2016]. An interesting problem would be to combine the DPM detector with the Convolutional Neural Networks (CNN). It is straightforward to incorporate the DPM detector as the last layer of the CNN and to train both models simultaneously by the back-propagation algorithm. A possible advantage of using the DPM layer is that this way one can easily enforce hard constraints on the landmark configurations.

Bibliography

Amit, Y. and Kong, A. (1996). Graphical templates for model registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:225–236.

Antipov, G., Baccouche, M., Berrani, S.-A., and Dugelay, J.-L. (2016). Apparent age estimation from face images combining general and children-specialized deep learning models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Asthana, A., Zafeiriou, S., Cheng, S., and Pantic, M. (2013). Robust Discriminative Response Map Fitting with Constrained Local Models. In *The 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'13*, pages 3444–3451, Portland, OR, USA.

Asthana, A., Zafeiriou, S., Cheng, S., and Pantic, M. (2014). Incremental Face Alignment in the Wild. In *The 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'14*, pages 1859–1866, Columbus, OH, USA.

Baker, S. and Matthews, I. A. (2004). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255.

Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. (2011). Localizing parts of faces using a consensus of exemplars. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'11*, pages 545–552, Colorado Springs, CO, USA.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, Belmont, MA.

Blanz, V. and Vetter, T. (2003). Face Recognition Based on Fitting a 3D Morphable Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074.

Bordes, A., Bottou, L., and Gallinari, P. (2009). SGD-QN: Careful Quasi-Newton Stochastic Gradient Descent. *Journal of Machine Learning Research*, 10:1737–1754.

Cao, X., Wei, Y., Wen, F., and Sun, J. (2012). Face alignment by explicit shape regression. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2887–2894.

Cao, X., Wei, Y., Wen, F., and Sun, J. (2014). Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190.

Čech, J., Franc, V., and Matas, J. (2014). A 3D Approach to Facial Landmarks: Detection, Refinement, and Tracking. In *22nd International Conference on Pattern Recognition, ICPR'14*, pages 2173–2178, Stockholm, Sweden.

Čech, J., Franc, V., Uřičář, M., and Matas, J. (2015). Multi-view facial landmark detection by using a 3D shape model. *Image and Vision Computing*, 47:60–70.

Çeliktutan, O., Ulukaya, S., and Sankur, B. (2013). A comparative study of face landmarking techniques. *EURASIP Journal on Image and Video Processing*, 2013(1):13.

Cheney, E. and Goldstain, A. (1959). Newton's method for convex programming and Tchebytcheff approximation. *Numerische Mathematick*, 1:253–268.

Chow, C. and Liu, C. (2006). Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theor.*, 14(3):462–467.

Chrysos, G., Antonakos, E., Zafeiriou, S., and Snape, P. (2015). Offline deformable face tracking in arbitrary videos. In *Proceedings of IEEE International Conference on Computer Vision, 300 Videos in the Wild (300-VW): Facial Landmark Tracking in-the-Wild Challenge & Workshop (ICCVW'15)*, Santiago, Chile.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

Cristinacce, D. and Cootes, T. F. (2006). Feature Detection and Tracking with Constrained Local Models. In *Proceedings of the British Machine Vision Conference 2006, BMVC'06*, pages 929–938, Edinburgh, UK.

Cristinacce, D. and Cootes, T. F. (2008). Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067.

Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *The 18th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'05*, pages 886–893, San Diego, CA, USA.

Dollár, P., Welinder, P., and Perona, P. (2010). Cascaded pose regression. In *The 23rd IEEE Conference on Computer Vision and Pattern Recognition, CVPR'10*, pages 1078–1085, San Francisco, CA, USA.

Ekman, P. and Friesen, W. (1978). *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto.

Escalera, S., Torres Torres, M., Martinez, B., Baro, X., Jair Escalante, H., Guyon, I., Tzimiropoulos, G., Corneou, C., Oliu, M., Ali Bagheri, M., and Valstar, M. (2016). Chalearn looking at people and faces of the world: Face analysis workshop and challenge 2016. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.*

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 2006, BMVC'06*, pages 899–908, Edinburgh, UK.

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`.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. A., and Ramanan, D. (2010). Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 61(1):55–79.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2012). Distance Transforms of Sampled Functions. *Theory of Computing*, 8(1):415–428.

Felzenszwalb, P. F. and Zabih, R. (2010). Dynamic programming and graph algorithms in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 721–740.

Fischler, M. A. and Elschlager, R. A. (1973). The Representation and Matching of Pictorial Structures. *IEEE Transactions on Computers*, C-22(1):67–92.

Goodall, C. (1991). Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–339.

Gross, R., Matthews, I., and Baker, S. (2005). Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093.

Gross, R., Matthews, I., Cohn, J. F., Kanade, T., and Baker, S. (2010). Multi-PIE. *Image and Vision Computing*, 28(5):807–813.

Hadid, A., Zhao, G., Ahonen, T., and Pietikäinen, M. (2008). Face analysis using local binary patterns. *Handbook of Texture Analysis*, pages 347–373.

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.

Jain, V. and Learned-Miller, E. (2010). FDDB: A Benchmark for Face Detection in Unconstrained Settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst.

Kasinski, A., Florek, A., and Schmidt, A. (2008). The put face database. *Image Processing and Communications*, 13(3-4):59–64.

Kazemi, V. and Sullivan, J. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. In *The 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'14*, pages 1867–1874, Columbus, OH, USA.

Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. (2016). The megaface benchmark: 1 million faces for recognition at scale. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Köstinger, M., Wohlhart, P., Roth, P. M., and Bischof, H. (2011). Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization. In *IEEE International Conference on Computer Vision Workshops, ICCV'11 Workshops*, pages 2144–2151, Barcelona, Spain.

Lai, H., Xiao, S., Cui, Z., Pan, Y., Xu, C., and Yan, S. (2015). Deep cascaded regression for face alignment.

Le, V., Brandt, J., Lin, Z., Bourdev, L., and Huang, T. S. (2012). *Interactive Facial Feature Localization*, pages 679–692. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lemaréchal, C. (1978). Nonsmooth optimization and descend methods. Technical report, IIASA, Laxenburg, Austria.

Lemaréchal, C., Nemirovskii, A., and Nesterov, Y. (1995). New variants of bundle methods. *Mathematical Programming*, 69:111–147.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

Martínez, B., Valstar, M. F., Binefa, X., and Pantic, M. (2013). Local Evidence Aggregation for Regression-Based Facial Point Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1149–1163.

Matthews, I. and Baker, S. (2004). Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(2):135–164.

Messer, K., Matas, J., Kittler, J., and Jonsson, K. (1999). Xm2vtsdb: The extended m2vts database. In *In Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77.

Milborrow, S. and Nicolls, F. (2014). Active Shape Models with SIFT Descriptors and MARS. *VISAPP*.

Ojala, T. and Pietikäinen, M. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *ICPR'94: The 12th IAPR International Conference on Pattern Recognition*.

Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.

Platt, J. C. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola et al., editor, *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Ren, S., Cao, X., Wei, Y., and Sun, J. (2014). Face Alignment at 3000 FPS via Regressing Local Binary Features. In *The 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'14*, pages 1685–1692, Columbus, OH, USA.

Roth, J., Tong, Y., and Liu, X. (2016). Adaptive 3d face reconstruction from unconstrained photo collections. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2016). 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing, Special Issue on Facial Landmark Localisation "In-The-Wild"*.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013a). 300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge. In *Proceedings of IEEE International Conference on Computer Vision Workshops, ICCV'13 Workshops*, Sydney, Australia.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013b). A Semi-automatic Methodology for Facial Landmark Annotation. In *The 26th IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR'13 Workshops*, pages 896–903, Portland, OR, USA.

Saragih, J. and Göcke, R. (2007). A Nonlinear Discriminative Approach to AAM Fitting. In *The 11th IEEE International Conference on Computer Vision, ICCV'07*, pages 1–8, Rio de Janeiro, Brazil.

Saragih, J. and Göcke, R. (2009). Learning AAM fitting through simulation. *Pattern Recognition*, 42(11):2628–2636.

Shamir, O. and Zhang, T. (2013). Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML (1)*, pages 71–79.

Shen, J., Zafeiriou, S., Chrysos, G., Kossaifi, J., Tzimiropoulos, G., and Pantic, M. (2015). The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of IEEE International Conference on Computer Vision, 300 Videos in the Wild (300-VW): Facial Landmark Tracking in-the-Wild Challenge & Workshop (ICCVW'15)*.

Sivic, J., Everingham, M., and Zisserman, A. (2009). "Who are you?" – Learning Person Specific Classifiers from Video. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Šochman, J. and Matas, J. (2005). WaldBoost - Learning for Time Constrained Sequential Detection. In *The 18th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'05*, pages 150–156, San Diego, CA, USA.

Sonnenburg, S. and Franc, V. (2010). COFFIN: A Computational Framework for Linear SVMs. In *Proceedings of the 27th International Conference on Machine Learning, ICML'10*, pages 999–1006, Haifa, Israel.

Sun, Y., Wang, X., and Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483.

Teo, C. H., Vishwanathan, S. V. N., Smola, A. J., and Le, Q. V. (2010). Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11:311–365.

Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., and Niessner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Trigeorgis, G., Snape, P., Nicolaou, M. A., Antonakos, E., and Zafeiriou, S. (2016). Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research*, 6:1453–1484.

Tzimiropoulos, G. (2015). Project-out cascaded regression with an application to face alignment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tzimiropoulos, G., i medina, J. A., Zafeiriou, S., and Pantic, M. (2012). Generic active appearance models revisited. In *11th Asian Conference on Computer Vision (ACCV 2012)*, pages 650–663, Daejeon, Korea.

Tzimiropoulos, G. and Pantic, M. (2013). Optimization problems for fast AAM fitting in-the-wild. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 593–600.

Tzimiropoulos, G. and Pantic, M. (2014). Gauss-Newton Deformable Part Models for Face Alignment In-the-Wild. In *The 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'14*, pages 1851–1858, Columbus, OH, USA.

Uřičář, M. and Franc, V. (2012). Efficient algorithm for regularized risk minimization. In *Proceedings of the 17th Computer Vision Winter Workshop, CVWW'12*, pages 57–64, Ljubljana, Slovenia. CD-ROM.

Uřičář, M., Franc, V., and Hlaváč, V. (2012). Detector of Facial Landmarks Learned by the Structured Output SVM. In *Proceedings of the International Conference on Computer Vision Theory and Applications, VISAPP'12*, volume 1, pages 547–556, Rome, Italy.

Uřičář, M., Franc, V., and Hlaváč, V. (2013). Bundle methods for structured output learning — back to the roots. In *Proceedings of the 18th Scandinavian Conference on Image Analysis, SCIA'13*, Lecture Notes in Computer Science, Berlin, Germany. Springer-Verlag.

Uřičář, M., Franc, V., and Hlaváč, V. (2015a). Facial Landmark Tracking by Tree-based Deformable Part Model Based Detector. In *Proceedings of IEEE International Conference on Computer Vision, 300 Videos in the Wild (300-VW): Facial Landmark Tracking in-the-Wild Challenge & Workshop, ICCVW'15*, Santiago, Chile. IEEE.

Uřičář, M., Franc, V., Thomas, D., Sugimoto, A., and Hlaváč, V. (2015b). Real-time Multi-view Facial Landmark Detector Learned by the Structured Output SVM. In *Proceedings of the 11th IEEE International Conference on Automatic Face and Gesture Recognition Conference and Workshops, BWILD'15*, Ljubljana, Slovenia. IEEE.

Uřičář, M., Franc, V., Thomas, D., Sugimoto, A., and Hlaváč, V. (2016a). Multi-view facial landmark detector learned by the structured output SVM. *Image and Vision Computing*, 47:45–59.

Uřičář, M., Timofte, R., Rothe, R., Matas, J., and Van Gool, L. (2016b). Structured output svm prediction of apparent age, gender and smile from deep features. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Las Vegas, NV, USA, June 26–July 1, 2016*.

Valstar, M. F., Almaev, T., Girard, J. M., McKeown, G., Mehu, M., Yin, L., Pantic, M., and Cohn, J. F. (2015). Fera 2015-second facial expression recognition and analysis challenge. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 6, pages 1–8. IEEE.

Valstar, M. F., Martínez, B., Binefa, X., and Pantic, M. (2010). Facial point detection using boosted regression and graph models. In *The 23rd IEEE Conference on Computer Vision and Pattern Recognition, CVPR'10*, pages 2729–2736, San Francisco, CA, USA.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.

Williams, L. (1983). Pyramidal Parametrics. In *Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '83, pages 1–11, New York, NY, USA. ACM.

Xiong, X. and la Torre, F. D. (2013). Supervised Descent Method and Its Applications to Face Alignment. In *The 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'13*, pages 532–539, Portland, OR, USA.

Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE.

Yu, X., Zhou, F., and Chandraker, M. (2016). Deep deformation network for object landmark localization.

Zhang, J., Shan, S., Kan, M., and Chen, X. (2014a). Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *European Conference on Computer Vision*, pages 1–16. Springer.

Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2014b). Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108. Springer.

Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *The 25th IEEE Conference on Computer Vision and Pattern Recognition, CVPR'12*, pages 2879–2886, Providence, RI, USA.

A. Author's publications

A.1. Publications related to the thesis

A.1.1. Impacted journal papers excerpted by ISI

Čech, J., Franc, V., **Uřičář, M.**, and Matas, J. (2016). Multi-view facial landmark detection by using a 3D shape model. *Image and Vision Computing*, 47:60–70. [25%].

Uřičář, M., Franc, V., Thomas, D., Sugimoto, A., and Hlaváč, V. (2016). Multi-view facial landmark detector learned by the structured output SVM. *Image and Vision Computing*, 47:45–59. [55%].

A.1.2. Conference papers excerpted by ISI

Uřičář, M., Franc, V., and Hlaváč, V. (2012). Detector of Facial Landmarks Learned by the Structured Output SVM. In *Proceedings of the International Conference on Computer Vision Theory and Applications, VISAPP'12*, volume 1, pages 547–556, Rome, Italy. [50%].

Uřičář, M., Franc, V., and Hlaváč, V. (2013). Bundle methods for structured output learning — back to the roots. In *Proceedings of the 18th Scandinavian Conference on Image Analysis, SCIA'13*, Lecture Notes in Computer Science, Berlin, Germany. Springer-Verlag. [45%].

Uřičář, M., Franc, V., and Hlaváč, V. (2013). Facial Landmarks Detector Learned by the Structured Output SVM. In Csurka, G., Kraus, M., Laramee, R. S., Richard, P., and Braz, J., editors, *Computer Vision, Imaging and Computer Graphics. Theory and Application*, volume 359 of *Communications in Computer and Information Science*, pages 383–398. Springer, Heidelberg, Germany. [50%].

Uřičář, M., Franc, V., and Hlaváč, V. (2015). Facial Landmark Tracking by Tree-based Deformable Part Model Based Detector. In *Proceedings of IEEE International Conference on Computer Vision, 300 Videos in the Wild (300-VW): Facial Landmark Tracking in-the-Wild Challenge & Workshop, ICCVW'15*, Santiago, Chile. IEEE. [60%].

Uřičář, M., Timofte, R., Rothe, R., Matas, J., and Van Gool, L. (2016). Structured output svm prediction of apparent age, gender and smile from deep features. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Las Vegas, NV, USA, June 26–July 1, 2016*. [50%].

A.1.3. Other conference papers

Uřičář, M. and Franc, V. (2012). Efficient algorithm for regularized risk minimization. In *Proceedings of the 17th Computer Vision Winter Workshop, CVWW'12*, pages 57–64, Ljubljana, Slovenia. [50 %].

Uřičář, M., Franc, V., Thomas, D., Sugimoto, A., and Hlaváč, V. (2015). Real-time Multi-view Facial Landmark Detector Learned by the Structured Output SVM. In

Proceedings of the 11th IEEE International Conference on Automatic Face and Gesture Recognition Conference and Workshops, BWILD'15, Ljubljana, Slovenia. IEEE. [50%].

Citations of author's work

The 155 known citations (to date April 7, 2017) of the author's work are grouped according to a paper they cite and listed below. The corresponding H-index is 6 on Google Scholar. The respective H-index on Web of Science is 4.

Uřičář, M., Franc, V., and Hlaváč, V. (2012). Detector of Facial Landmarks Learned by the Structured Output SVM. In Proceedings of the International Conference on Computer Vision Theory and Applications, VISAPP'12, volume 1, pages 547–556, Rome, Italy. [cited 107 times]

Ali, H., Tariq, U. U., and Abid, M. (2014). Learning discriminating features for gender recognition of real world faces. *International Journal of Image and Graphics*, 14(03):1450011.

Antoniuk, K. (2016). Discriminative learning from partially annotated examples.

Antoniuk, K., Franc, V., and Hlaváč, V. (2013). Mord: Multi-class classifier for ordinal regression. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 96–111. Springer.

Antoniuk, K., Franc, V., and Hlavác, V. (2014). Interval insensitive loss for ordinal classification. In *ACML*.

Antoniuk, K., Franc, V., and Hlavác, V. (2015). Consistency of structured output learning with missing labels. In *Proceedings of The 7th Asian Conference on Machine Learning*, pages 81–95.

Antoniuk, K., Franc, V., and Hlaváč, V. (2016). V-shaped interval insensitive loss for ordinal classification. *Machine Learning*, 103(2):261–283.

Augustin, T. (2016). Emotion determination in elearning environments based on facial landmarks. In *International Workshop on Learning Technology for Education in Cloud*, pages 122–136. Springer.

Azarmehr, R., Laganiere, R., Lee, W.-S., Xu, C., and Laroche, D. (2015). Real-time embedded age and gender classification in unconstrained video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 57–65.

Azazi, A., Lutfi, S. L., and Venkat, I. (2014a). Analysis and evaluation of surf descriptors for automatic 3d facial expression recognition using different classifiers. In *Information and Communication Technologies (WICT), 2014 Fourth World Congress on*, pages 23–28. IEEE.

Azazi, A., Lutfi, S. L., and Venkat, I. (2014b). Identifying universal facial emotion markers for automatic 3d facial expression recognition. In *Computer and Information Sciences (ICCOINS), 2014 International Conference on*, pages 1–6. IEEE.

Azazi, A., Lutfi, S. L., Venkat, I., and Fernández-Martínez, F. (2015). Towards a robust affect recognition: Automatic facial expression recognition in 3d faces. *Expert Systems with Applications*, 42(6):3056–3066.

Bovée, S., Lézoray, O., and Hamel, P. (2015). Analyse de signaux sociaux non verbaux de vidéos d'entretiens en face à face. In *Journées francophones des jeunes chercheurs en vision par ordinateur*, Amiens, France.

Campr, P., Kunešová, M., Vaněk, J., Čech, J., and Psutka, J. (2014). Audio-video speaker diarization for unsupervised speaker and face model creation. In *International Conference on Text, Speech, and Dialogue*, pages 465–472. Springer.

Campr, P., Pražák, A., Psutka, J. V., and Psutka, J. (2013). Online speaker adaptation of an acoustic model using face recognition. In *International Conference on Text, Speech and Dialogue*, pages 378–385. Springer.

Carvalho, D. R. and Segundo, M. P. (2015). Monocular gaze tracking system for disabled people assistance.

Cech, J., Franc, V., and Matas, J. (2014). A 3d approach to facial landmarks: detection, refinement, and tracking. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2173–2178. IEEE.

Cech, J., Mittal, R., Deleforge, A., Sanchez-Riera, J., Alameda-Pineda, X., and Horaud, R. (2013). Active-speaker detection and localization with microphones and cameras embedded into a robotic head. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 203–210. IEEE.

Çeliktutan, O., Ulukaya, S., and Sankur, B. (2013). A comparative study of face landmarking techniques. *EURASIP Journal on Image and Video Processing*, 2013(1):1.

Černỳ, P. (2014). Odhad tepové frekvence srdce z videa tváře.

Cevikalp, H. and Franc, V. (2017). Large-scale robust transductive support vector machines. *Neurocomputing*, pages –.

Chen, M.-X. (2017). Deep learning on time-frequency representation for heart rate estimation.

Chevallier, L., Vigouroux, J.-R., Goguey, A., and Ozerov, A. (2013). Facial landmarks localization estimation by cascaded boosted regression. In *International Conference on Computer Vision, Imaging and Computer Graphics*, pages 103–115. Springer.

Cui, Y., Zhang, J., Guo, D., and Jin, Z. (2015). Robust facial landmark localization using classified random ferns and pose-based initialization. *Signal Processing*, 110:46–53.

Dahmane, M., Cossette, S., and Meunier, J. (2015). Conditional gabor phase–based disparity estimation applied to facial tracking for person–specific facial action recognition: a preliminary study. *Multimedia Tools and Applications*, 74(17):7111–7130.

DeFelice, E., Ersoy, Y., and Zhou, K. (2012). Expression recognition system: Read me!

Divel, S. and Shunhavanich, P. (2016). Baby face generator.

Drouard, V., Ba, S., and Horaud, R. (2017). Switching linear inverse-regression model for tracking head pose. In *IEEE Winter Conference on Applications of Computer Vision*.

Ellis, J. G., Jou, B., and Chang, S.-F. (2014). Why we watch the news: A dataset for exploring sentiment in broadcast video news. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 104–111. ACM.

Ersoy, Y. (2013). Express recognition exploring methods of emotion detection.

Fadil, C., Alvarez, R., Martínez, C., Goddard, J., and Rufiner, H. (2015). Multimodal emotion recognition using deep networks. In *VI Latin American Congress on Biomedical Engineering CLAIB 2014, Paraná, Argentina 29, 30 & 31 October 2014*, pages 813–816. Springer.

Feng, Y., Huang, W., and Wu, T. (2003). Face swapping. *ACM Transactions on Graphics (TOG)*, 22(3).

Fernández, A., Carús, J. L., Usamentiaga, R., Alvarez, E., and Casado, R. (2015a). Unobtrusive health monitoring system using video-based physiological information and activity measurements. In *Computer, Information and Telecommunication Systems (CITS), 2015 International Conference on*, pages 1–5. IEEE.

Fernández, A., Casado, R., and Usamentiaga, R. (2015b). A real-time big data architecture for glasses detection using computer vision techniques. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 591–596. IEEE.

Fernández, A., García, R., Usamentiaga, R., and Casado, R. (2015c). Glasses detection on real images based on robust alignment. *Machine Vision and Applications*, 26(4):519–531.

Franc, V. (2014). Fasole: Fast algorithm for structured output learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 402–417. Springer.

Gabbouj, M. (2013). *UMAR IQBAL IMPORTANT PERSON DETECTION FROM MULTIPLE VIDEOS*. PhD thesis, TAMPERE UNIVERSITY OF TECHNOLOGY.

Gao, Y., Liu, H., Wu, P., and Wang, C. (2016). A new descriptor of gradients self-similarity for smile detection in unconstrained scenarios. *Neurocomputing*, 174:1077–1086.

Goodarzi, F. and Saripan, M. I. (2015). Real time face pose estimation using geometrical features. In *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 482–487. IEEE.

Gossen, F. (2014). Head pose normalization for recognition of human identities using color and depth data. In *International Symposium on Leveraging Applications of Formal Methods*, pages 97–112. Springer.

Grundström, J. (2015). Face verification and open-set identification for real-time video applications. *Master's Theses in Mathematical Sciences*.

Gu, D. (2016). *Cross models for twin recognition*. PhD thesis, Auckland University of Technology.

Gu, D., Nguyen, M., and Yan, W. (2016). Cross Models for Twin Recognition. *INTERNATIONAL JOURNAL OF DIGITAL CRIME AND FORENSICS*, 8(4):26–36.

Guan, N. C. (2014). Automated landmarks detection on 3d human facial image.

Günther, M., El Shafey, L., and Marcel, S. (2016). Face recognition in challenging environments: An experimental and reproducible research survey. In *Face Recognition Across the Imaging Spectrum*, pages 247–280. Springer.

Holešovskỳ, O. (2015). Face descriptor learned by convolutional neural networks.

Hu, Y., Kautz, J., Yu, Y., and Wang, W. (2015). Speaker-following video subtitles. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(2):32.

Huang, K. (2014). *Face verification using high dimensional feature*. PhD thesis, University of Missouri–Columbia.

Iqbal, U., Curcio, I. D., and Gabbouj, M. (2014). Who is the hero? semi-supervised person re-identification in videos. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 2, pages 162–173. IEEE.

Jin, X. and Tan, X. (2016). Face alignment in-the-wild: A survey. *arXiv preprint arXiv:1608.04188*.

Jou, B., Li, H., Ellis, J. G., Morozoff-Abegauz, D., and Chang, S.-F. (2013). Structured exploration of who, what, when, and where in heterogeneous multimedia news sources. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 357–360. ACM.

Kaewwongsri, K. and Suvonvorn, N. (2015). Automatic checkpoint system using face and card information. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(1):28–33.

Kang, M. W. and Kim, Y.-S. (2015). Development of gender classification scheme using facebook dataset for unrestricted real environments. *International Information Institute (Tokyo). Information*, 18(7):3149.

Kapsouras, I., Tefas, A., Nikolaidis, N., Peeters, G., Benaroya, L., and Pitas, I. (2016). Multimodal speaker clustering in full length movies. *Multimedia Tools and Applications*, pages 1–20.

Kapsouras, I., Tefas, A., Nikolaidis, N., and Pitas, I. (2015). Multimodal speaker diarization utilizing face clustering information. In *International Conference on Image and Graphics*, pages 547–554. Springer.

Karppa, M., Viitaniemi, V., Luzardo, M., Laaksonen, J., and Jantunen, T. (2014). Slmotion-an extensible sign language oriented video analysis tool. In *LREC*, pages 1886–1891.

Laurentini, A. and Bottino, A. (2014). Computer analysis of face beauty: A survey. *Computer Vision and Image Understanding*, 125:184–199.

Lewenberg, Y., Bachrach, Y., Shankar, S., and Criminisi, A. (2016). Predicting personal traits from facial images using convolutional neural networks augmented with facial landmark information. *arXiv preprint arXiv:1605.09062*.

Liew, C. F. and Yairi, T. (2016). Robust face alignment with random forest: Analysis of initialization, landmarks regression, and shape regularization methods. *IEICE TRANSACTIONS on Information and Systems*, 99(2):496–504.

Liu, H., Gao, Y., and Wu, P. (2014). Smile detection in unconstrained scenarios using self-similarity of gradients features. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1455–1459. IEEE.

Liu, P., Li, S., Dai, Q., and Deng, H. (2015). Face detection based on landmark localization. In *Proceedings of the 4th International Conference on Computer Engineering and Networks*, pages 547–554. Springer.

Liu, Y.-L. (2014). Color and motion analysis of facial component for pulse detection.

Luo, C., Gao, W., Jia, Z., Han, R., Li, J., Lin, X., Wang, L., Zhu, Y., and Zhan, J. (2016). Handbook of bigdatabench (version 3.1)—a big data benchmark suite.

Luzardo, M., Karppa, M., Laaksonen, J., and Jantunen, T. (2013). Head pose estimation for sign language video. In *Scandinavian Conference on Image Analysis*, pages 349–360. Springer.

Milborrow, S. (2016). *Multiview active shape models with SIFT descriptors*. PhD thesis, University of Cape Town.

Mohanraj, V., Vaidehi, V., Kumar, R., and Nakkeeran, R. (2016). Face recognition in video using deformable parts model with scale invariant feature transform (dpsift). In *Advanced Computing and Systems for Security*, pages 69–80. Springer.

Nigam, A., Chhalotre, G., and Gupta, P. (2015). Pose and illumination invariant face recognition using binocular stereo 3d reconstruction. In *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 1–4. IEEE.

Noiruxsar, C. and Samanpiboon, P. (2014). Face orientation recognition for electric wheelchair control. *Journal of Automation and Control Engineering*, 2(4).

Orfanidis, G., Tefas, A., Nikolaidis, N., and Pitas, I. (2014). Facial image clustering in stereo videos using local binary patterns and double spectral analysis. In *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, pages 217–221. IEEE.

Orfanidis, G., Tefas, A., Nikolaidis, N., and Pitas, I. (2015). Facial image clustering in stereoscopic videos using double spectral analysis. *Signal Processing: Image Communication*, 33:86–105.

Ozerov, A. (2014). Facial landmarks localization estimation by cascaded boosted regression. In *Computer Vision, Imaging and Computer Graphics–Theory and Applications: International Joint Conference, VISIGRAPP 2013, Barcelona, Spain, February 21-24, 2013, Revised Selected Papers*, volume 458, page 103. Springer.

Paul, A. K., Choo, H.-G., Kim, S., Kim, D., Kim, J., Kim, P., and Yoon, Y. (2015). Multi persons' pupil detection and tracking for digital table top holographic display.

Pedersoli, M., Timofte, R., Tuytelaars, T., and Van Gool, L. (2014). Using a deformation field model for localizing faces and facial points under weak supervision. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3694–3701. IEEE.

Poignant, J., Bredin, H., and Barras, C. (2015a). Multimodal person discovery in broadcast tv at mediaeval 2015. *Proceedings of MediaEval*.

Poignant, J., Fortier, G., Besacier, L., and Quénot, G. (2015b). Naming multi-modal clusters to identify persons in tv broadcast. *Multimedia Tools and Applications*, pages 1–25.

Portaz, M., Budnik, M., Mulhem, P., and Poignant, J. (2016). Mrim-lig at imageclef 2016 scalable concept image annotation task.

Sedmidubsky, J., Mic, V., and Zezula, P. (2015). Face image retrieval revisited. In *International Conference on Similarity Search and Applications*, pages 204–216. Springer.

Setthawong, P. and Vajirasak, V. (2015). Cascading deformable parts model in the facial feature detection of frontal and side profile images. *The International Arab Journal of Information Technology*.

Sharma, R. (2014). Object detection using dimensionality reduction on image descriptors.

Sharma, R. and Savakis, A. (2015). Lean histogram of oriented gradients features for effective eye detection. *Journal of Electronic Imaging*, 24(6):063007–063007.

Sidaty, N. O., Larabi, M.-C., and Saadane, A. (2014). Towards understanding and modeling audiovisual saliency based on talking faces. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on*, pages 508–515. IEEE.

Song, F., Tan, X., Chen, S., and Zhou, Z.-H. (2013). A literature survey on robust and efficient eye localization in real-life scenarios. *Pattern Recognition*, 46(12):3157–3173.

Tang, C., Xu, P., Luo, Z., Zhao, G., and Zou, T. (2015a). Automatic facial expression analysis of students in teaching environments. In *Chinese Conference on Biometric Recognition*, pages 439–447. Springer.

Tang, L. Y., Hannah, B., Jongman, A., Sereno, J., Wang, Y., and Hamarneh, G. (2015b). Examining visible articulatory features in clear and plain speech. *Speech Communication*, 75:1–13.

Tang, Z., Yan, C., Ren, S., Wan, Huagen", e. C.-S., Lu, J., and Ma, K.-K. (2017). *HeadPager: Page Turning with Computer Vision Based Head Interaction*, pages 249–257. Springer International Publishing, Cham.

Tariq, U. U., Ahmad, W., Asif, M. D. A., and Hussain, M. (2013). Gender perception from faces using boosted lbph (local binary patten histograms). *Carpathian Journal of Electronic and Computer Engineering*, 6(1):8.

Tay, N. N. W., Botzheim, J., and Kubota, N. (2016). Joint probabilistic approach for real-time face recognition with transfer learning. *Robotics and Autonomous Systems*, 75:409–421.

Tay, N. N. W., Botzheim, J., Loo, C. K., and Kubota, N. (2014). Robust face recognition via transfer learning for robot partner. In *Robotic Intelligence In Informationally Structured Space (RiiSS), 2014 IEEE Symposium on*, pages 1–8. IEEE.

Timoshenko, D. and Grishkin, V. (2013). Composite face detection method for automatic moderation of user avatars. *Computer Science and Information Technologies (CSIT'13)*.

van Rijn, F. (2014). Automatic detection of facial tics in gilles de la tourette syndrome patients.

Viitaniemi, V., Karppa, M., Laaksonen, J., and Jantunen, T. (2013). Detecting hand-head occlusions in sign language video. In *Scandinavian Conference on Image Analysis*, pages 361–372. Springer.

Villan, A. F., Candas, J. L. C., Fernandez, R. U., and Tejedor, R. C. (2016). Face recognition and spoofing detection system adapted to visually-impaired people. *IEEE Latin America Transactions*, 14(2):913–921.

Wang, N., Gao, X., Tao, D., and Li, X. (2014). Facial feature point detection: A comprehensive survey. *arXiv preprint arXiv:1410.1037*.

Wang, Z. and Yan, J. (2013). Who and where: People and location co-clustering. *arXiv preprint arXiv:1307.8405*.

Wang, Z., Yan, J., and Aghajan, H. (2012a). A framework of personal assistant for computer users by analyzing video stream. In *Proceedings of the 4th Workshop on Eye Gaze in Intelligent Human Machine Interaction*, page 14. ACM.

Wang, Z., Yan, J., Pang, C., Chu, D., and Aghajan, H. (2012b). Who is here: Location aware face recognition. In *Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones*, page 8. ACM.

Xu, W., Shen, Y., Bergmann, N., and Hu, W. (2016). Sensor-assisted face recognition system on smart glass via multi-view sparse representation classification. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12. IEEE.

Yonetani, R., Kawashima, H., and Matsuyama, T. (2013). Learning spatiotemporal gaps between where we look and what we focus on. *IPSJ Transactions on Computer Vision and Applications*, 5(0):75–79.

Yu, X., Huang, J., Zhang, S., and Metaxas, D. N. (2015). Face landmark fitting via optimized part mixtures and cascaded deformable model.

Yu, X., Huang, J., Zhang, S., Yan, W., and Metaxas, D. N. (2013). Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1944–1951.

Zheng, S., Sturgess, P., and Torr, P. H. (2013). Approximate structured output learning for constrained local models with application to real-time facial feature detection and tracking on low-power devices. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE.

Zhou, F., Brandt, J., and Lin, Z. (2013). Exemplar-based graph matching for robust facial landmark localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1025–1032.

**Uřičář, M., Franc, V., and Hlaváč, V. (2013).
Facial Landmarks Detector Learned by the Structured Output SVM. In Csurka, G., Kraus, M., Laramee, R. S., Richard, P., and Braz, J., editors, In Computer Vision, Imaging and Computer Graphics. Theory and Application, volume 359 of Communications in Computer and Information Science, pages 383–398. Springer, Heidelberg, Germany. [cited 6 times]**

Alletto, S., Serra, G., Calderara, S., and Cucchiara, R. (2014a). Head pose estimation in first-person camera views. In *ICPR*, pages 4188–4193.

Alletto, S., Serra, G., Calderara, S., Solera, F., and Cucchiara, R. (2014b). From ego to nos-vision: Detecting social relationships in first-person views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 580–585.

Saravanan, S., Palanivel, S., and Balasubramanian, M. (2014). Facial expression based person authentication. *International Journal of Computer Applications*, 94(13).

Setthawong, P. and Vanijja, V. (2013). Modified deformable parts model for side profile facial feature detection. In *International Conference on Advances in Information Technology*, pages 212–220. Springer.

Varas González, D., Vilaplana Besler, V., Morros Rubió, J. R., Pericás, H., and Javier, F. (2015). Upc system for the 2015 mediaeval multimodal person discovery in broadcast tv task. In *MediaEval 2015 Multimedia Benchmark Workshop*.

Zhang, G., Chen, J., Su, G., and Ou, C. (2016). Comparisons of local methods for face alignment. *IET Computer Vision*.

**Uřičář, M., Franc, V., Thomas, D., Sugimoto, A., and Hlaváč, V. (2015).
Real-time Multi-view Facial Landmark Detector Learned by the Structured Output SVM.
In Proceedings of the 11th IEEE International Conference on Automatic Face and Gesture Recognition Conference and Workshops, BWILD'15, Ljubljana, Slovenia. IEEE. [cited 16 times]**

Antipov, G., Baccouche, M., Berrani, S.-A., and Dugelay, J.-L. (2016). Apparent age estimation from face images combining general and children-specialized deep learning models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Boutellaa, E., Boulkenafet, Z., Komulainen, J., and Hadid, A. (2016). Audiovisual synchrony assessment for replay attack detection in talking face biometrics. *Multimedia Tools and Applications*, pages 1–15.

Comaschi, F., Stuijk, S., Basten, T., and Corporaal, H. (2016). Robust online face tracking-by-detection.

Egger, B., Schneider, A., Blumer, C., Morel-Forster, A., Schönborn, S., and Vetter, T. (2016). Occlusion-aware 3D morphable face models. In *Proceedings of the British Machine Vision Conference 2016, BMVC'16*.

Fernández, A., Usamentiaga, R., Carús, J. L., and Casado, R. (2016). Driver distraction using visual-based sensors and algorithms. *Sensors*, 16(11):1805.

Fogelton, A. and Benesova, W. (2016). Eye blink detection based on motion vectors analysis. *Computer Vision and Image Understanding*, 148:23–33.

Gonzalez, M. and Collet, C. (2016). Face tracking using a combination of colour and pattern matching based on particle filter.

Jia, S., Lansdall-Welfare, T., and Cristianini, N. (2016). Gender classification by deep learning on millions of weakly labelled images. In *IEEE Workshop Data Mining in Human Activity Analysis, ICDMW'16*.

Le Borgne, H., Gadeski, E., Chami, I., Tran, T. Q. N., Tamaazousti, Y., Ginsca, A. L., and Popescu, A. (2016). Image annotation and two paths to text illustration.

Liba, O. (2015). Face and photograph augmentation based on a custom theme.

Liu, Y., Zhou, X., Li, Y., Shao, X., and Zhou, X. (2016). Robust facial landmark detection using mixture of discriminative visibility-aware models. *IET Image Processing*.

Merget, D., Eckl, T., Schwoerer, M., Tiefenbacher, P., and Rigoll, G. (2016). Capturing facial videos with kinect 2.0: A multithreaded open source tool and database. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–5. IEEE.

Merget, D., Tiefenbacher, P., Babaee, M., Mitov, N., and Rigoll, G. (2015). Photorealistic face transfer in 2d and 3d video. In *German Conference on Pattern Recognition*, pages 400–411. Springer.

Steger, A., Timofte, R., and Van Gool, L. (2016). Failure detection for facial landmark detectors. *arXiv preprint arXiv:1608.06451*.

Steiner, H., Kolb, A., and Jung, N. (2016). Reliable face anti-spoofing using multispectral swir imaging. In *Fierrez, Li et al. (Eds.): 2016 International Conference on Biometrics (ICB). Proceedings, 13-16 June 2016, Halmstad, Sweden*.

Uřičář, M., Franc, V., and Hlaváč, V. (2015).
Facial Landmark Tracking by Tree-based Deformable Part Model Based Detector.
In Proceedings of IEEE International Conference on Computer Vision, 300 Videos in the Wild (300-VW): Facial Landmark Tracking in-the-Wild Challenge & Workshop, ICCVW'15, Santiago, Chile. IEEE. [cited 8 times]

Alp Güler, R., Trigeorgis, G., Antonakos, E., Snape, P., Zafeiriou, S., and Kokkinos, I. (2016). Densereg: Fully convolutional dense shape regression in-the-wild. *ArXiv e-prints*.

Chrysos, G. G., Antonakos, E., Snape, P., Asthana, A., and Zafeiriou, S. (2016). A comprehensive performance evaluation of deformable face tracking "in-the-wild". *CoRR*, abs/1603.06015.

Fernández, A., Usamentiaga, R., Carús, J. L., and Casado, R. (2016). Driver distraction using visual-based sensors and algorithms. *Sensors*, 16(11):1805.

Li, M., Jeni, L., and Ramanan, D. (2017). Brute-force facial landmark analysis with a 140,000-way classifier.

Liba, O. (2015). Face and photograph augmentation based on a custom theme.

Liu, Q., Yang, J., Deng, J., and Zhang, K. (2016). Robust facial landmark tracking via cascade regression. *Pattern Recognition*, pages –.

Sheih, A. (2016). Simulating imitative learning in a humanoid robot for the purpose of language acquisition.

Shen, J., Zafeiriou, S., Chrysos, G., Kossaifi, J., Tzimiropoulos, G., and Pantic, M. (2015). The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of IEEE International Conference on Computer Vision, 300 Videos in the Wild (300-VW): Facial Landmark Tracking in-the-Wild Challenge & Workshop (ICCVW'15)*.

Čech, J., Franc, V.,Uřičář, M., and Matas, J. (2016). Multi-view facial landmark detection by using a 3D shape model Image and Vision Computing, 47:60–70. [cited 6 times]

Alp Güler, R., Trigeorgis, G., Antonakos, E., Snape, P., Zafeiriou, S., and Kokkinos, I. (2016). Densereg: Fully convolutional dense shape regression in-the-wild. *ArXiv e-prints*.

Kossaifi, J., Tzimiropoulos, Y., and Pantic, M. (2016). Fast and exact newton and bidirectional fitting of active appearance models. *IEEE Transactions on Image Processing*, PP(99):1–14.

Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2016). 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 47:3–18.

Soukupová, T. (2016). Eye blink detection using facial landmarks.

Tuzel, O., Marks, T. K., and Tambe, S. (2016). Robust face alignment using a mixture of invariant experts. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 825–841, Cham. Springer International Publishing.

Zafeiriou, S., Tzimiropoulos, G., and Pantic, M. (2016). 300 w: Special issue on facial landmark localisation "in-the-wild". *Image and Vision Computing*, 47:1–2.

Uřičář, M., Franc, V., Thomas, D., Sugimoto, A., and Hlaváč, V. (2016). Multi-view facial landmark detector learned by the structured output SVM Image and Vision Computing, 47:45–59. [cited 7 times]

Alp Güler, R., Trigeorgis, G., Antonakos, E., Snape, P., Zafeiriou, S., and Kokkinos, I. (2016). Densereg: Fully convolutional dense shape regression in-the-wild. *ArXiv e-prints*.

Dou, Y., Zhou, M., Wang, J., and Qiang, J. (2016). Facial expression recognition based-on saliency guided support vector machine. In *Computational Intelligence and Design (ISCID), 2016 9th International Symposium on*, volume 2, pages 389–393. IEEE.

Jin, X. and Tan, X. (2016). Face alignment in-the-wild: A survey. *arXiv preprint arXiv:1608.04188*.

Kossaifi, J., Tzimiropoulos, Y., and Pantic, M. (2016). Fast and exact newton and bidirectional fitting of active appearance models. *IEEE Transactions on Image Processing*, PP(99):1–14.

Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2016). 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 47:3–18.

Wang, L., Zhang, K., Liu, X., Long, E., Jiang, J., An, Y., Zhang, J., Liu, Z., Lin, Z., Li, X., et al. (2017). Comparative analysis of image classification methods for automatic diagnosis of ophthalmic images. *Scientific Reports*, 7.

Zafeiriou, S., Tzimiropoulos, G., and Pantic, M. (2016). 300 w: Special issue on facial landmark localisation "in-the-wild". *Image and Vision Computing*, 47:1–2.

Uřičář, M., Timofte, R., Rothe, R., Matas, J., and Van Gool, L. (2016). Structured output svm prediction of apparent age, gender and smile from deep features.
In 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Las Vegas, NV, USA, June 26–July 1, 2016. [cited 6 times]

Agustsson, E., Timofte, R., Escalera, S., Baro, X., Guyon, I., and Rothe, R. (2017). Apparent and real age estimation in still images with deep residual regressors on appa-real database.

Bekhouche, S., Ouafi, A., Dornaika, F., Taleb-Ahmed, A., and Hadid, A. (2017). Pyramid multi-level features for facial demographic estimation. *Expert Systems with Applications*.

Dehghan, A., Ortiz, E. G., Shu, G., and Masood, S. Z. (2017). Dager: Deep age, gender and emotion recognition using convolutional neural network. *arXiv preprint arXiv:1702.04280*.

Ranjan, R., Sankaranarayanan, S., Castillo, C. D., and Chellappa, R. (2016). An all-in-one convolutional neural network for face analysis. *CoRR*, abs/1611.00851.

Rothe, R., Timofte, R., and Van Gool, L. (2016). Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, pages 1–14.

Torfason, R., Agustsson, E., Rothe, R., and Timofte, R. (2016). From face images and attributes to attributes. In *Asian Conference on Computer Vision*.