Image and Vision Computing 42 (2015) 47–61

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Empirical analysis of cascade deformable models for multi-view face detection☆

Javier Orozco [a,*], Brais Martinez [a], Maja Pantic [a,b]

[a] Imperial College, Department of Computing, London, UK
[b] University of Twente, EEMCS, Twente, Netherlands

ARTICLE INFO

Article history:
Received 11 June 2014
Received in revised form 18 July 2015
Accepted 24 July 2015
Available online 14 August 2015

Keywords:
Multi-view face detection
Cascade deformable models
FDDB database
AFLW database
HPID database
COFW dataset

ABSTRACT

We present a multi-view face detector based on Cascade Deformable Part Models (CDPM). Over the last decade, there have been several attempts to extend the well-established Viola&Jones face detector algorithm to solve the problem of multi-view face detection. Recently a tree structure model for multi-view face detection was proposed. This method is primarily designed for facial landmark detection and consequently a face detection is provided. However, the effort to model inner facial structures by using a detailed facial landmark labelling resulted on a complex and suboptimal system for face detection. Instead, we adopt CDPMs, where the models are learned from partially labelled images using Latent Support Vector Machines (LSVM). Furthermore, LSVM is enhanced with data-mining and bootstrapping procedures to enrich models during the training. Furthermore, a post-optimization procedure is derived to improve the performance. This semi-supervised methodology allows us to build models based on weakly labelled data while incrementally learning latent positive and negative samples. Our results show that the proposed model can deal with highly expressive and partially occluded faces while outperforming the state-of-the-art face detectors by a large margin on challenging benchmarks such as the Face Detection Data Set and Benchmark (FDDB) [1] and the Annotated Facial Landmarks in the Wild (AFLW) [2] databases. In addition, we validate the accuracy of our models under large head pose variation and facial occlusions in the Head Pose Image Database (HPID) [3] and Caltech Occluded Faces in the Wild (COFW) datasets [4], respectively. We also outline the suitability of our models to support facial landmark detection algorithms.

1. Introduction

Face detection is invariably the first step in any automatic face analysis system. With the rapid increase of computational power and modern digital signal processing, face detection is a handy and a customary feature present in many human sensing applications. Still, the key aspect of performance is not only the ability to detect the face quickly, but also reliability and precision. It is indeed common that further processes are initialised upon the face detection output, including face alignment, face modelling, face relighting, face recognition, face authentication, head pose estimation, facial expression recognition, gender/age recognition, and many more [5].

For the past decade, face detection has relied on the influential Viola&Jones (VJ) algorithm [6]. Near-frontal face detection suddenly became feasible as the VJ algorithm provides real-time performance for head pose variation up to 30° of yaw and 15° of pitch rotations.

Driven by the imminent necessities of technological progress, face analysis recently abandoned the typical controlled scenarios of the lab-produced databases to tackle real world challenges. That is to say, face detection must evolve from restricted settings where near-frontal faces, clean backgrounds, perfect illumination and occlusion-free faces are acquired. This is why the latest challenges in face analysis arise from unconstrained imagery collected over the internet, widely known as the "in-the-wild" databases [2,1].

The VJ algorithm was early exhibited as largely insufficient to handle the head pose rotations in databases like the Multi-Pie database [7] and the plethora of in-the-wild databases that followed. It is within this context that Multi-View Face Detection (MVFD) rapidly raised in practical importance.

A first attempt to extend the VJ algorithm to MVFD was proposed by Viola and Jones [8]. They proposed a two-stage MVFD, where the face pose is initially estimated, followed by face detection according to pose-wise models. Other subsequent works proposed different modifications to the VJ detector. The most relevant proposals include the use of different cascade architectures, variants of Boosting and modified Haar features.

For example, [9] proposed the use of a pyramid of classifiers to deal with MVFD, where lower levels of the pyramid would be increasingly

☆ This paper has been recommended for acceptance by Qiang Ji.
* Corresponding author. Tel.: +44 20 7594 8336; fax: +44 20 7581 8024.
E-mail address: drjo2009@gmail.com (J. Orozco).
URL: http://www.ibug.doc.ic.ac.uk/people/jorozco (J. Orozco).

http://dx.doi.org/10.1016/j.imavis.2015.07.002
0262-8856/

specific to a head pose. FloatBoost was used instead of AdaBoost to avoid the greedy search, and an extension of Haar features was used. Similarly, [10] used Real AdaBoost to train pose-wise experts, while a nested-structured cascade was proposed to replace the cascade architecture of VJ algorithm. Alternatively, [11] proposed a tree-shaped organization of the cascaded search in combination with a variant of Boosting, called Vector Boosting, which allowed classes to share features. This variant aimed at alleviating the impact of class overlap when training pose-wise experts for MVFD. However, most of these works were strictly incremental over the VJ algorithm, and offered no breakthrough.

Papers that are more recent have proposed the substitution of Haar-like features. For example, [12,13] attained a large performance improvement by using SURF features, which are equally computable by means of the integral image. Face detection is still attained through a cascade of SURF weak classifiers, trained with billions of samples within one hour. This work resulted on the best performance to date over the FDDB benchmark database [1]. Alternatively, [14] proposed the use of Local Gradient Patterns to build a feature pyramid while maintaining the cascaded AdaBoost as the learning algorithm. The large performance gain of these methods suggests that the representation power of Haar-like features is a bottleneck of the classical VJ framework. However, these articles do not specifically tackle MVFD.

New works have emerged applying the vast advances on generic object detection to the specific problem of face detection. For example, [15] proposed to apply methods based on local scale-invariant keypoint features to solve the problems of pose-invariant face detection. A similar idea was followed in [16], where SIFT features are detected within the images and then used to score similarity between two images as the number of positive key-points matching. This work reported results on the FDDB database [1] but did not outperform the results reported in [17], where a Gaussian Process Regression scheme was applied to adapt the VJ pre-trained model to the "in-the-wild" domain. Very similar approach was used in [18], where authors trained a Gaussian Mixture Model to also adapt the VJ model to the "in-the-wild".

Recently, a major breakthrough for MVFD was obtained when Zhu and Ramanan [19] proposed to apply another object detection framework, the Deformable Parts Model (DPM), for joint face detection, pose estimation and facial landmark detection. Specifically, they proposed a model formed by 68 part filters per pose, each one corresponding to a facial landmark. Their spatial relations are modelled using a Tree Structure Model (hereafter we will refer to this method as TSM). The absence of loops in the shape model means that minimization can be attained through dynamic programming. Finally, the model was composed of 13 head pose-wise experts, corresponding to the poses present on the Multi-Pie database. This approach showed a much better performance than VJ-like methodologies for MVFD. Essentially, a finer face representation, modelling inner facial structures, Histogram of Oriented Gradients (HOG) features [20] and view-dependent models, lead this method to a better discrimination power.

However, the way inner facial structures are modelled appears optimal for facial landmark detection while being suboptimal when only face detection is intended. Firstly, it requires an exhaustive facial landmark labelling, which hugely reduces the amount of training data that can be used. Secondly, the large amount of experts and parts makes the algorithm too slow for face detection in practical applications. The resolution required is higher as the part filters rely on local statistics for a successful detection. Finally, the TSM model lacks a holistic face filter that could speed-up the face detection at lower resolutions and improve its robustness to partial occlusions. We argue that the baseline framework of DPMs as defined in [21] is more suitable for MVFD than the TSM, which is actually derived from [21].

Recent contributions to the literature extended the TSM framework [19], to propose structural models for body and face detection [22–24]. These models pursue a double objective, detection of faces, facial parts and/or facial landmarks. Such an ontological dual function also requires more complete facial annotations beyond simple face bounding boxes. Moreover, the method proposed in [22] makes use of contextual information by combining the results of an upper body detector to improve the face detection performance. This increases the complexity of the annotated data required to train such model.

The star-structured model of the original DPM has shown excellent detection performance on difficult benchmarks such as the PASCAL datasets [25]. When using star models, facial appearances are modelled using multi-scale DPMs. Further performance improvement is attained by combining Latent Support Vector Machines (LSVM) and data-mining procedures. Finally, a Cascade Deformable Part Model (CDPM) [26] can speed up over 20 times the DPM's detection without sacrificing detection accuracy.

In this paper, we present an empirical analysis of CDPMs to address the reliability problem of MVFD. First, we describe a data-mining process to incrementally learn DPMs from partially labelled data using the LSVM algorithm. Second, we derive a post-optimization procedure for the CDPMs training that improves significantly its performance. As a result, we obtain a face detector that outperforms the state-of-the-art face detectors on three challenging "in-the-wild" datasets such as the Face Detection Database (FDDB) [1], the Annotated Facial Landmarks in the Wild [2], and the Caltech Occluded Faces in the Wild (COFW) [4]. Additionally, detailed experimental results on the Head Pose Image Database [3] are discussed. We also provide analyses regarding average face detection times and the accuracy of our MVFD for the initialization of facial landmark detection.

2. Multi-view face detection

One of the most efficient and remarkable frameworks in object detection has been presented by Felzenszwalb et al. [21]. This method proposes to build pictorial structures composed by a set of dual resolution image filters, which are a global object model and a set of parts representing object sub-structures, arranged according to a deformable spatial configuration. Here, we describe an empirical analysis of this DPM to address the problem of efficient MVFD.

First, we present the DPM and detail the face detection hypothesis. Second, we explain the process of discriminatively learning a DPM from weakly labelled data using the LSVM algorithm [21]. At this point, we introduce two post-processing procedures, data-mining and bootstrapping, which allow us to refine training sets while increasing the robustness of the model. Finally, we describe cascaded search strategy, where early stages in the cascade use a basic DPM face detector to rapidly scan the image while speeding up face detection without any performance loss.

2.1. Deformable part models

Following the original DPM's framework [21], let us define a DPM with n parts as $\beta = \{F_0, P_1, \ldots, P_n, b\}$, where F_0 is a coarse-scale global *Root-Filter*, P_i is a *Part-Filter* model for the i^{th} part and b is a bias term. Part filters are defined as $P_i = \{F_i, v_i, w_i\}$, where F_i is a fine-scale part filter at twice the resolution of the root filter. The spatial distribution of part filters is defined relative to the root filter by both v_i and w_i, the anchor and deformation penalty, respectively.

DPM filters are matrices designed to weight the sub-windows of a pyramidal representation of an image. We employ a variant of the HOG features introduced by Dalal and Triggs [20] to represent the facial appearance. These features have shown to be robust for object detection under challenging conditions such as image noise, scale variation and occlusions [25].

Given both a DPM and a HOG feature pyramid of a testing image x, a binary convolution function, $\Phi(x, \beta)$, scans the responses of the model β onto the testing image. The score of a filter with respect to a sub-window of a HOG pyramid is the dot product of the weight vector and the features comprising the sub-window. Thus, the scoring

function combines the appearance fitness and a penalization of spatial deformation as follows:

$$S_\beta(x) = \Phi(x, F_0) + \sum_{i=1}^{n} \max_{\delta_i \in \Delta} \Phi(x, P_i, \delta_i) - w_i(\delta_i) \quad (1)$$

where $\Phi(x, F_0)$ is the root filter response, δ_i gives the displacement of part filters relative to its anchor and the root's position. In this model, each part is expected to keep a specific relative position respect to the root filter, called the anchor point. The part can move away from its anchor point, but it incurs in a penalization, $w_i(\delta_i)$, when doing so. This penalization might however be outweighed by the improved matching of the part filter. Thus, $\Phi(x, P_i, \delta_i) - w_i(\delta_i)$ scores the responses of the part filters under the displacement from the anchor point, δ_i, and the deformation cost associated with the displacement, w_i. We model the deformation as a symmetric two-dimensional Gaussian mask superimposed on the target sub-window, with mean location being the anchor point.

A face detection model is implemented as a mixture of DPMs, in which each DPM's component is designed to respond only to a subset of the possible appearances and deformations. In our case each subset corresponds to a distinct range of head poses. Fig. 1 shows a DPM example that comprises four mixture components representing near-frontal and profile faces, left and right (only the right view components are displayed). Each mixture component has a root filter (Fig. 1.(a)) and six independent part filters, (Fig. 1.(b)), this is known as $4 * (Roots + 6Parts)$ DPM. Fig. 1(top) shows a face detection example of this DPM, where the red bounding boxes correspond to the maximum combined scored, Eq. (1), and blue boxes display the best configuration of the model part filters.

Observe that this DPMs are trained with weakly labelled data, i.e. only the face bounding box is known in opposition to previous works [19,24,23,22]. Consequently, the part filters composing a view-based mixture detector do not correspond to any face part or facial landmark.

2.1.1. DPM training

Training a robust face detector for the "in-the-wild" images requires a large amount of data from a variety of databases. Ideally, we want to learn from both lab-designed and "in-the-wild" databases, but the main challenge is the lack of consensus in the annotations. Therefore, we deal with this issue by adopting a multi-instance learning formulation, Latent Support Vector Machines (LSVM). This consists of training an initial face model using a partially labelled dataset, with homogeneous bounding box annotations. Afterwards, new latent variables are collected in order to extent the primary training set.

Now, let us define a classifier that scores an example image x with the following function:

$$S_\beta(x) = \max_{z \in Z(z)} \beta \cdot \Phi(x, z) \quad (2)$$

where $Z(x)$ defines a set of possible latent variables for an example x, scored using the DPM β and the scoring function in Eq. (1). In our case, these latent variables are obtained by evaluating all DPM view-

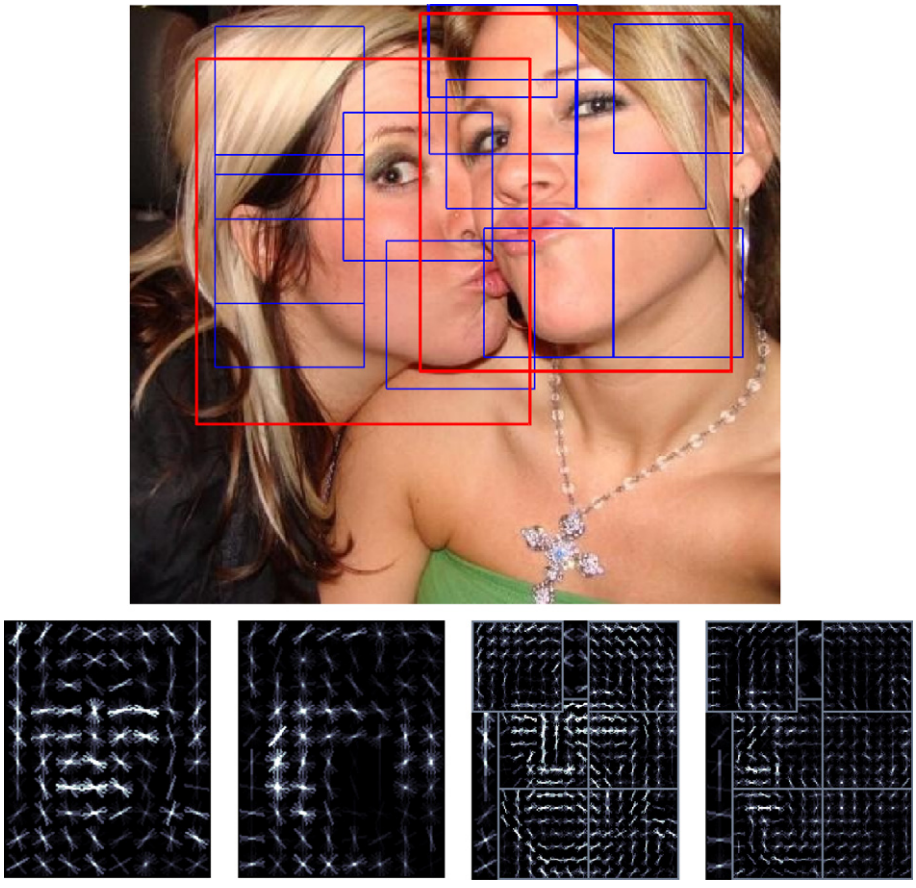

Fig. 1. Face detections obtained with a four components model. Each component is defined by (a) root filters (red-boxes) and high resolution (b) part filters (blue-boxes). These view components of a 4-Roots DPM cover faces on near frontal and profile head poses (only the right view components are displayed).

based components on the hypothesis x. The detection with maximum score is kept and a binary label is assigned to x upon a minimum detection score threshold. In a similar fashion to training an SVM, we use the LSVM algorithm [21] to train a DPM for face detection while obtaining β. To this end, a face DPM is discriminatively trained with labelled examples by minimizing the following loss function via a coordinate descent algorithm:

$$L_{\mathcal{D}}(\beta) = \frac{\|\beta\|^2}{2} + c\sum_{i=1}^{k} \max(0,\, 1 - y_i \cdot S_\beta(x_i)) \tag{3}$$

here, $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n))\}$ is the training set and $y_i \in \{-1, 1\}$ are the binary class labels. $\max(0,\, 1 - y_i \cdot S_\beta(x_i))$ is the standard hinge loss and c is a regularization term.

In general, training an LSVM requires optimizing a non-convex function. Still, there are two strategies to ease the LSVM optimization as proven by Felzenszwalb et al. [21]. First, the training of LSVM is made convex by specifying the latent information for the positive training examples, while the negative training examples remain fixed. Second, $S_\beta(x)$ is made linear in β by collecting only one latent variable for each positive example, $|Z(x)| = 1$. Bear in mind that at this point we are training linear SVM as a special case of LSVM, using latent variables. Consequently, we obtain the perfect scenario to use of-the-shelf optimization algorithms and large training datasets.

2.1.1.1. Root and part filters. DPM combines mixture models able to deal with facial appearance variation due to head pose and facial expressions. Hence, root filters allow discriminating faces from the background while deformable part filters can adapt to expressive faces and head pose variations.

Usually face images are labelled with bounding boxes, which enable training of rigid face detection models. A more complete labelling might be used such as the facial landmarks used by Zhu and Ramanan [19]. However, such level of detail combined with the definition of fine inner facial structures make the TSM a suboptimal face detector. Instead, we first train a DPM that contains a mixture of view-based root filters learned from labelled data. Subsequently, the mixture of root filters is used to acquire latent examples that serve as training set for new root and part filters. The initial structure of the part filters is obtained by applying Gaussian Mixture Models (GMM). Next, the gradient descent process of LSVM allows finding the best possible location of the parts relative to the root filter such that the detection score is a maximum of Eq. (1).

Afterwards, finding the optimal DPM requires to alternate between the acquisition of latent examples and retraining LSVM until the best performance measure is achieved on a validation set.

To train a face detector with high performance, LSVM relies upon the precision of the root filters to extend the training set with new detected faces. This allows incrementally learning root and part filters as latent variables. Felzenszwalb et al. applied this data mining strategy over the positive training samples to learn non-deformable objects. However, face detection presents additional challenges due to a large diversity of head poses and facial expressions. To deal with this appearance variations and high deformability of faces, we propose to split the positive training set, D_p, into easy and hard positives, D_{ep} and D_{hp}, correspondingly.[1] Likewise, the negative training set, D_n, is extended with a set of hard negatives, $\mathcal{Z}_{hn}$, which are positively scored detections collected from outside the annotated face bounding boxes.

Initially, a mixture of coarse root filters is discriminatively trained using easy positives and negative examples, D_{ep} and D_n, respectively. Note that this step of the LSVM is reduced to the simple case of training a binary SVM for view-based mixture component. Once all root filters are obtained, the scoring function, $S_\beta(x)$, is globally normalized according to all easy positive examples. This is in order to enable comparisons of detections from different components.

To incrementally learn DPMs based on latent variables, the root filters obtained from the easy positives are used to score the training set D_{ep}. Thus, the corresponding set of latent positives, $\mathcal{Z}_{ep}$, is obtained. Then, LSVM is applied again to discriminatively train a new mixture of root filters based on the latent easy positives and negative examples, $\mathcal{Z}_{ep}$ and D_n, respectively. Here, LSVM seeks to minimize the objective function $L_{\mathcal{Z}_{ep}}(F_0)$ in Eq. (3).

As mentioned above, each mixture component consists of a root filter and a set of part filters, which are designed to cope with both face appearance and facial deformation, correspondingly. Therefore, part filters are trained while keeping the previously learned root filters. In fact, root filters are used to score both easy and hard positive examples to produce a complete set of latent positives, $\mathcal{Z}_p = \{\mathcal{Z}_{ep} \cup \mathcal{Z}_{hp}\}$. Subsequently, LSVM is applied to discriminatively learn part filters using Z_p and D_n.

Differently than learning root filters, the learning of part filters is a constrained optimization process as follows:

$$\begin{aligned} &L_{\mathcal{Z}_p, \mathcal{D}_n}(\beta) = \min\left\{\frac{\|\beta\|^2}{2} + c\sum_{i=1}^{k} \max(0,\, 1 - y_i \cdot S_\beta(z_i))\right\} \\ &s.t. \\ &S_\beta(z) = \max_{z \in Z_p(z)} \{\beta \cdot \Phi(z)\}. \end{aligned} \tag{4}$$

Furthermore, the HOG features of latent part locations, $z_p \in \mathcal{Z}_p$, are computed at twice the resolution of the root filters. Thereby, part filters are built using higher resolution features computed over highly scored latent positive examples. This combination of dual feature resolution and latent variables enables root filters to capture coarse resolution edges such as the face's boundary while part filters capture details such as eyes, nose and mouth.

2.1.2. Data-mining and bootstrapping

Face detectors are normally trained with a large number of negative examples. For a feasible discriminative training, the most common practice is to use all positive data and hard negative instances. Yet, to avoid computational overloads, bootstrapping methods propose to train a model with an initial subset of negative examples, and then collect additional negative examples that are incorrectly classified by the initial model. Then, an iterative process is established repeating the extraction of hard negatives and re-training the model until optimal stopping criteria are met.

Motivated by the data-mining procedure described in [21] and the necessity to exploit large number of face images, here we combine both data-mining and bootstrapping. We define a margin-sensitive clustering procedure to extend both negative and positive examples into easy and hard latent instances. Given the training set of positive and negative labelled data, $\mathcal{D} = \{\mathcal{D}_p \cup \mathcal{D}_n\}$, we define easy and hard examples relative to a face DPM, β, as follows:

$$\begin{aligned} \mathcal{Z}_{hp}(\beta) &= \left\{(x,y) \in D_p \middle| y_i^+ \cdot S_\beta(x) < 1\right\} \quad , Hard-Positives \\ \mathcal{Z}_{ep}(\beta) &= \left\{(x,y) \in D_p \middle| y_i^+ \cdot S_\beta(x) > 1\right\} \quad , Easy-Positives \\ \mathcal{Z}_{hn}(\beta) &= \left\{(x,y) \in D_p \middle| y_i^- \cdot S_\beta(x) < 1\right\} \quad , Hard-Negatives. \end{aligned} \tag{5}$$

It can be seen that $\mathcal{Z}_{hp}$ and $\mathcal{Z}_{hn}$ are positive and negative latent examples incorrectly classified by β, that is, data within the SVM margin. Instead, $\mathcal{Z}_{ep}$ are correctly classified examples, so they fall outside the margins with a high detection score. Let $x_p \in \mathcal{D}_p$ and $z_p \in \mathcal{Z}_p$ be two bounding boxes of a positive annotation and the corresponding detection, respectively. As determined in the PASCAL VOC [25], we measure the *Overlapping* percentage as follows:

$$Overlapping = \frac{area(x_p \cap z_p)}{area(x_p \cup z_p)}. \tag{6}$$

[1] The initial split could be based on a priori knowledge of the training data, e.g. images taken under controlled illumination and clean backgrounds are easier to learn.

An initial model β_0 is trained with LSVM using only annotated easy positives and negatives, $\mathcal{D}_{ep}$ and $\mathcal{D}_n$, respectively. Subsequently, as detailed in Section 2.1.1, LSVM is used iteratively alternating between caching a set of "good" training samples and updating the cache. For the LSVM training problem, we determine as latent positive example, z_p, a detection window such that the *Overlapping* with x_p is greater than 50%. However, an *Overlapping* of 70% determines whether the z_p instance belongs to either $\mathcal{Z}_{hp}$ or $\mathcal{Z}_{ep}$. This allows to apply data-mining in the positive examples at slow learning rate but with highly scored new latent examples.

Like with positive examples, there is a set of hard negatives, $\mathcal{Z}_{hn}$, which are highly scored detections collected from $\mathcal{D}_p$. Thus, a detected window is considered as hard negative if the *Overlapping* with the annotation x_p is lower than 50% but with a high detection score, e.g. $S_\beta(x) \geq 0$.

Accordingly, a bootstrapping stage is performed by updating the cache with latent hard negative examples, $\mathcal{Z}_{hn}$. This post-processing procedure is intended to maximize the correlation between the precision–recall and the score function of the face detector. In addition, both data-mining and bootstrapping procedures contribute to refine the SVM margins while reducing highly scored detections around accurate face detections.

This whole data-mining/bootstrapping procedure is repeated upon convergence to the optimal precision–recall computed over a validation set.

2.2. Cascade deformable part models

Felzenszwalb et al. also provide a Star-Cascade (SC) algorithm [26] in order to speed-up the DPM detection without a loss in accuracy. Contrary to the TSM [19], a DPM with start model structure already outperforms the TSM at the first stage of the cascaded classification. The mixture of root filters of the DPM are more efficient proxies than the small facial part-like features of the TSM, resulting in a much larger reduction of the computational cost.

To circumvent the bottle-neck of DPMs, a Cascade DPM (CDPM) is trained to find likely object locations that are later validated by the DPM. Although this procedure is not specific to the star model, the CDPM consists of a tailored root filter capable of scanning the image at low resolution whereas part filters are used at high resolution over the locations provided by CDPM's root filter.

In our model, we tailor the root filter model and the corresponding parts to be hierarchically applied, resulting in $n + 2$ stages, where n is the number of part filters. The SC algorithm learns a global threshold, τ, which is used to score the most likely locations with the CDPM's root filter, $S_c(F_0) \geq \tau$. This score is accumulated throughout the stages of a cascade. If $S_c(F_0)$ with the first i parts is lower than a threshold τ_i, the root location is not evaluated for the rest of the cascade. This is known as *hypothesis-pruning*. SC will also skip locations if the deformation w_i is above a threshold τ'_i. Finally, the SC algorithm will use the CDPM for hypothesis-pruning at early stages as a proxy to highlight faces from the background. Once a candidate location is found, we compute the actual filter convolution of the underlying image features with the face DPM including both roots and part filters. This additional stage in the cascade allows to suppress all but the best detections in a faster manner.

Further speed-up can be attained by using PCA-HOG features to encode the appearance of root and part filters of the CDPM. This allows obtaining simplified cascade models with no noticeable loss of information as demonstrated in [26]. That is, the CDPM's filters are projected onto a fix number of eigenvectors achieving a faster face detection while reducing memory requirements. Here, our face CDPMs are trained with filters of 5-PCA-HOG features learned using the corresponding DPM and latent positive examples.

Lower dimensional features may improve the precision as consequence of applying PCA, but at the expense of recall loss. Therefore, we propose to lessen this effect with a post-optimization procedure, which improves the CDPM's performance, i.e. precision–recall. We use both labelled and latent (easy and hard) positive examples to build an eigenspace of HOG features. Next, we follow the same steps as in [26] to compute the 5-PCA-HOG features corresponding to the CDPM's root filters.

3. Training details

Here, we detail the training procedure followed by both VJ-MVFD and our DPM-MVFD. We used 35, 738 publicly available face images, see Table 1. Images from video sequences were clustered into different views (head poses) using the 3D head pose estimation given by the tracking system in [32]. The training set only contained faces with pitch and roll angles within the range of $\pm 20°$.

3.1. MVFD with Viola and Jones

As baseline, we trained a VJ-MVFD because the work in [8] is not publicly available. The training has been carried out using the OpenCV library [33], using a Gentle AdaBoost classifier, the upright Haar-like features and a tree-based cascade structure for an efficient search [34]. We trained a 6-Views MVFD for near-frontal, [0°, 30°], half-profile, (30°, 60°], and full profile, (60°, 90°] faces. The training set of 35, 738 face images from Table 1 was extended to 100, 000 positive examples by flipping the images and applying random distortions. Our training of the VJ-MVFD took approximately four weeks per pose. Bear in mind that this haar-cascade training has several parameters suitable for õoptimization such as number of stages, type of haar-features, minimum hit rate and maximum false alarm rate. However, the major improvement in performance comes from the appropriate combination of pose-specific components.

To detect a face, the VJ-MVFD runs all pose-specific detectors in parallel. Next, detections are merged by first using a disjoint-set data structure function [33] to cluster the detected rectangles according to their size and location. Then, clusters with a small number of rectangles are eliminated. Finally, a non-maximum suppression function is used to merge the remaining detections. The detections are scored as the maximum response of the Haar-like features among the view-specific detectors.

3.2. MVFD with CDPM

We trained four different CDPM face detectors to assess the performance depending on the number of root and parts filters. A $4 * (Roots + 6Parts)$ CDPM was trained using images for near-frontal, [0°, ± 30°], and profile, (±30°, ± 90°], faces. Face images were flipped, which allows building symmetric view-based models but asymmetric filters, as both left and right views are trained independently. A second model was trained with $8 * (Roots + 6Parts)$ following the annotation structure provided with the Multi-Pie database [7]. For this model, we clustered the images according to the views, (±0°, ± 30°], (±30°, ± 45°], (±45°, ± 60°] and (±60°, ± 90°].

Table 1
Composition of the training datasets containing 35, 738 positive examples.

Database	# Examples
AFLW [2]	10096
Cohn–Kanade [27]	3130
DaFeX [28]	996
FGnet [29]	1962
MMI [30]	1150
Mind Reading [31]	6552
MultiPIE [7]	7952
Head Pose Database [3]	3900

The same views were used to train a $8 * (Roots + 20Parts)$ CDPM. Lastly, we trained a $13 * (Roots + 6Parts)$ by splitting the head rotation of $[-90°, +90°]$ on every 15°, so that is 13 views were obtained.

Using the 35, 738 face images (labelled with bounding boxes) as listed in Table 1, we first learned the root filters of a DPM using the LSVM algorithm as explained in Section 2.1. To avoid scatter root models and benefit from less noisy annotations, we initially train them with easy positives, $\mathcal{D}_{ep}$. AFLW images were used as latent hard positives, $\mathcal{Z}_{hp}$. Hence, we disregarded the provided annotations, and latent bounding boxes were extracted instead by applying the data-mining process explained in Section 2.1.2.

We adopt the PASCAL VOC precision–recall protocol for object detection [25]. A hypothesis is considered as a correct detection if the annotation and the estimation are at least 50% overlapped, Eq. (6).

3.2.1. DPM design

The root filters of our DPM models were designed using HOG features extracted from 10×8 pixels cells to match the head aspect ratio of the 75% of the annotated faces. Instead, part-filters are designed with HOG features extracted from square pixel cells of size 6×6. Our DPM models were trained using 32-dimensional HOG features, which were originally proposed by Felzenwalb et al. [21]. Subsequently, 50% of the annotated data are used to train an eigenspace of root-filters of these HOG features. Next, the first 5 eigenvectors of the trained eigenspace are taken as basis to represent the main structure of our CDPM.

3.2.2. Data-mining in action

To train our DPM models with weak labelled and using data-mining and bootstrapping, we found out that the best strategy was alternating these two process in 2×1 stages. As explained in Section 2.1.2, we threshold the latent detections on our training set to distinguish easy positives $\mathcal{Z}_{ep}$, hard positives $\mathcal{Z}_{hp}$ and hard negatives $\mathcal{Z}_{hn}$. This alternating procedure is described as follows:

1. Data-mining *easy—positives*
   1.1. Obtain latent detections using previous model.
   1.2. Select easy positives using overlapping threshold, Eq. (6).
   1.3. Train DPM using LSVM and $\mathcal{Z}_{ep}$ while keeping support vectors from previous model.
2. Data-mining *hard—positives*
   2.1. Obtain latent detections using previous model.
   2.2. Select hard positives using overlapping threshold, Eq. (6).
   2.3. Train DPM using LSVM and $\mathcal{Z}_{hp}$ while keeping support vectors from previous model.
3. Bootstrapping *hard—negatives*
   3.1. Obtain latent detections using previous model.
   3.2. Select hard negatives using overlapping threshold, Eq. (6).
   3.3. Train DPM using LSVM using all above data $\mathcal{Z}_{ep} + \mathcal{Z}_{hp} + \mathcal{Z}_{hn}$ to obtain a new set of support vectors, which generalize for the extended dataset.

The above steps (1) and (2) adopt similar strategies as on-line learning methods by adding new support vectors to previous models. This allows increasing the generalization strength of the model based on positive training samples. Finally, a fresh model is trained in step (3) after obtaining all the latent training samples from positives and negatives.

The training of each view-based DPM component is initialized with at least 400 to 500 positive annotated samples, which ensures an AP grater than 90% for the first root filters. Then, a first round of the above (1) to (3) steps is conducted for each view-based component while using data corresponding to that view only. However, to achieve highly discriminative view-based components, posterior rounds are advanced using all remaining data, where the view component with highest scored detection will retain that latent sample for further retraining.

To obtain a robust mixture of DPMs, we repeat this alternating data-mining and bootstrapping steps at least five times. This has been determined by using a separate validation sample to assess the convergence of the improvement in AP. Additional statistics also ensure that more than 95% of our positive samples have been used while reducing the false positives rate.

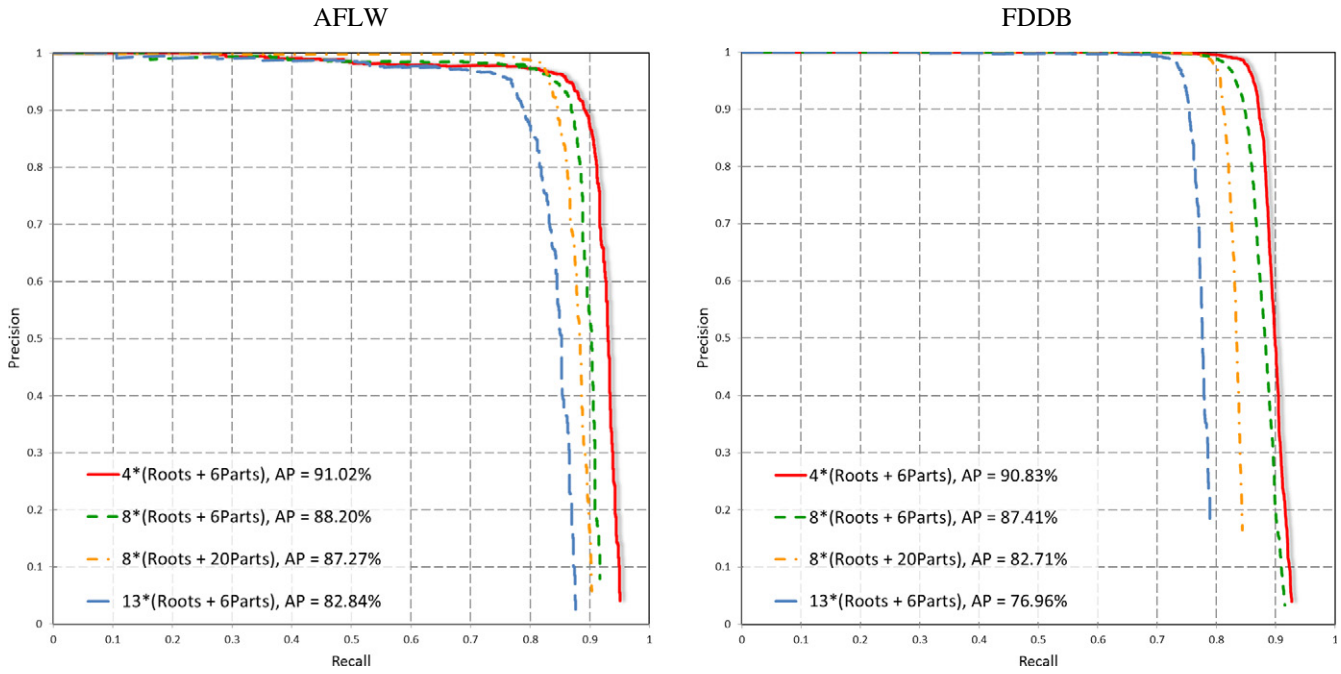

Fig. 2. Precision–Recall curves of four face detection CDPMs tested on the AFLW and the FDDB datasets. Each model is named according to the number of Roots and Parts composing it. In addition, the Average Precision (AP) is reported besides the curves.

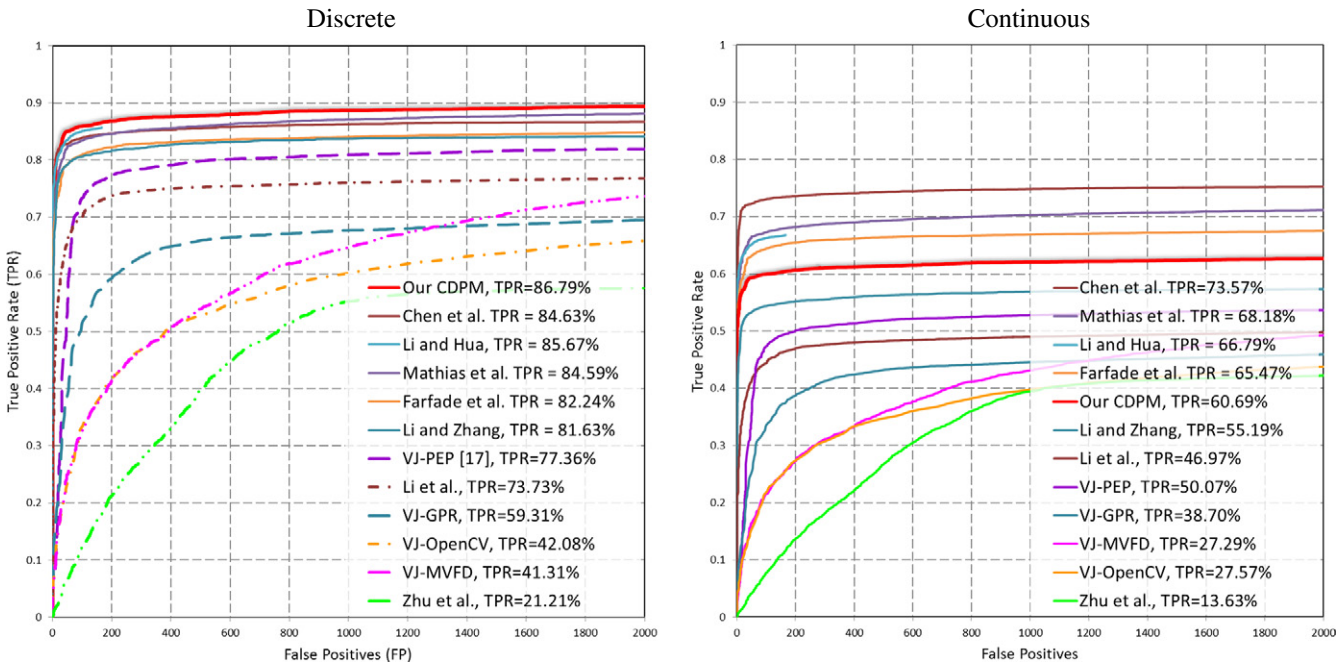

Fig. 3. ROC curves for different methods tested on the FDDB. We report the face detection performance of "Our CDPM" according to the scheme proposed, showing discrete and continuous ROC curves, respectively. The *TPR* is reported for $FP = 200$ along with each method. Our CDPM outperforms for a margin of 2% the best result of the state-of-the-art (left). Our CDPM is outperformed in the continues ROC (right) by a method that is boosted by the simultaneous face detection and face alignment [24].

3.2.3. Training times

The full training process of a DPM with four view-based components and six filters can take about a week. The process is slow because involves running face detection on 35,000 images, three times for each of the rounds described in Section 3.2.2. This calculation is done assuming the availability of a pool of 20 workers @ 3.60 GHz, 16 GB of RAM and Linux 64 bits. The training cost can increase upon the number of root and part filters, where the latter drive the highest computation overload as they require a HOG pyramid at double resolution of the root filters features. However, we believe that both training and running cost can be drastically reduced by optimizing the HOG computation and the use of GPU power.

4. Experimental results

In this section, we describe the experimental results obtained with our face detector. We start comparing different CDPMs by varying the number of mixture components and part filters. Next, we discuss the performance of our best model on two "in-the-wild" databases, FDDB and AFLW. Subsequently, we validate the performance of our CDPM-MVFD according to head pose variation in the HPID database. We also include a face detection test under different levels of occlusion. This test is performed in an "in-the-wild" dataset that has been recently released, the COFW database. Finally, we present how our MVFD performs when used to initialize a facial landmarking algorithm.

4.1. DPM trade-off

In order to find the optimal combination of components and part filters, we have conducted some preliminary experiments. Fig. 2 displays the face detection results using four different CDPMs. To compare the performance of different face detectors, we compute the average precision as established by the PASCAL VOC [25].

It can be seen how the $4 * (Roots + 6Parts)$ CDPM face detector is outperforming the remaining models with an Average Precision (AP) of 91.02% on the AFLW database [2]. The four models performed comparably when tested on the FDDB database [1]. As can be seen, the precision drops with either higher number of roots or parts. This indicates that a higher number of views leads to suboptimal face detectors such as the $13 * (Roots + 6Parts)$ model. This is probably because higher number of pose-wise components require larger amounts of training data. However, the $8 * (Roots + 20Parts)$ model achieves more precise detections but at lower recall. Consequently, finding an appropriate trade-off between the number of roots and

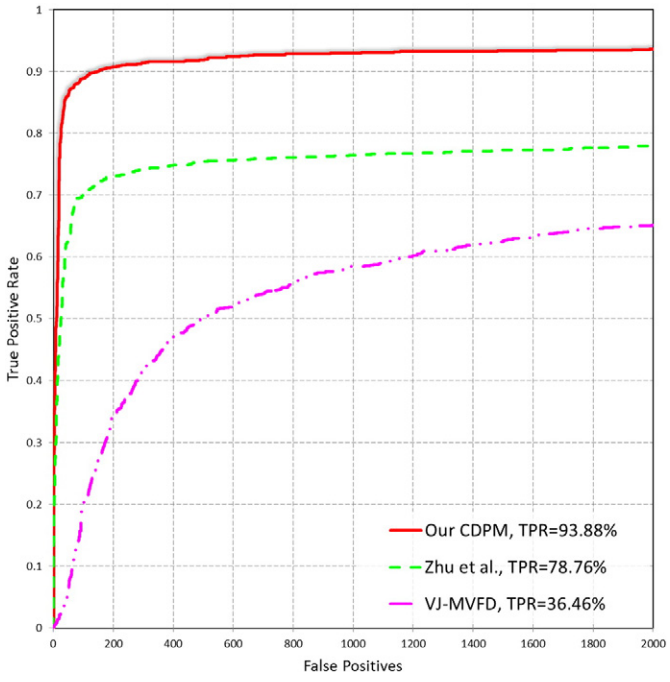

Fig. 4. Performance on the AFLW. We report the discrete ROC curves corresponding to "Our CDPM", our implementation of the VJ-MVFD and the TSM face detector [19]. They are compared according to the TPR with at most $FP = 200$.

parts is required and this should be constrained by the amount of available training data.

4.2. Experiments on FDDB

The FDDB database [1] is the latest benchmark dataset for face detection in real world scenarios. It contains 2845 images and 5171 faces acquired under unconstrained conditions. This dataset is released with a standard performance evaluation scheme proposed by Jain et al. [1]. Bear in mind that faces are annotated with ellipses instead of rectangular bounding boxes. This is atypical since most object detection methods, including DPMs, learn and estimate rectangular bounding boxes. Thus, it is an extra challenge to achieve the right overlapping with the annotated ellipse according to Eq. (6).

In Fig. 3, we report the MVFD performance of "Our CDPM" on the FDDB using the $4*(Roots+6Parts)$ CDPM. The performance of our implementation of VJ-MVFD is also reported in both discrete and continuous ROC curves. We compare the different methods according to the maximum True Positive Rate (TPR) for a number of False Positives (FP) as high as 200. In addition, we include the MVFD performance of the TSM method [19] at a small number of false positives such as 200, together with the top five face detectors reported on the FDDB [35]. The result of the VJ-OpenCV implementation [33] for frontal faces is also included, see Fig. 3.

Table 2
Average detection time and the corresponding average precision on images of the FDDB database.

Face detector	Average time (s)	Average precision (%)
$4Roots+6Parts$	0.463	90.83
$8Roots+6Parts$	0.755	87.41
$8Roots+20Parts$	2.391	82.71
$13Roots+6Parts$	4.367	76.96
VJ-MVFD	0.520	73.44
TSM	26.063	49.52

It can be seen from Fig. 3, in the discrete ROC curve, that our CDPM achieves the highest performance on the FDDB discrete ROC. Our CDPM achieved a TPR greater than all methods at any rate of false

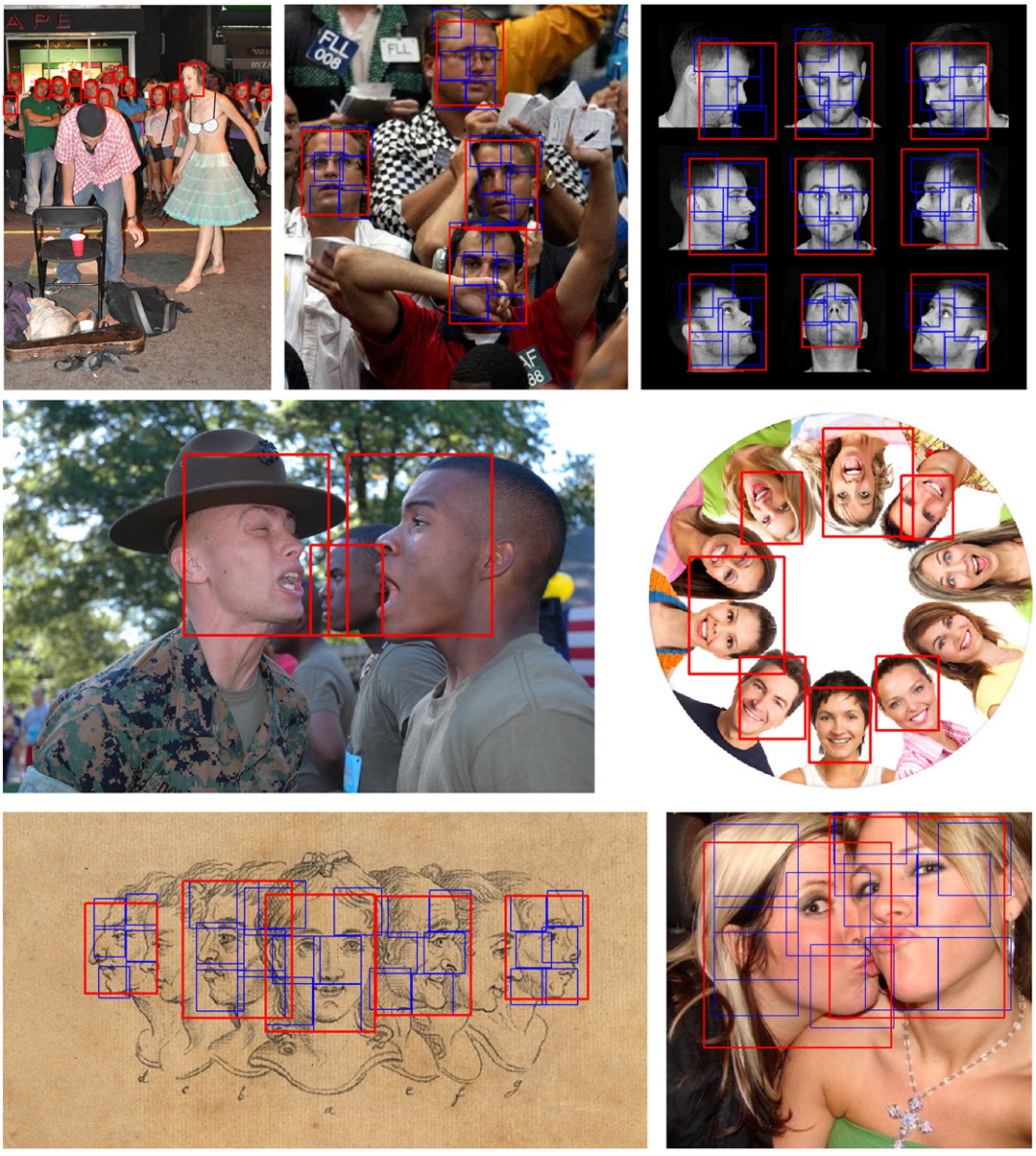

Fig. 5. Multi-view face detection with CDPM. These examples were obtained with the $4*(Roots+6Parts)$ CDPM on images from the FDDB and AFLW databases. Red bounding boxes determine the best location for the root filters whereas blue boxes are used for the part filters detections.

Fig. 6. MVFD with CDPM on HPID. We tested the performance of our $4 * (Roots + 6Parts)$ CDPM on images from the HPID databases. These images are annotated with pan and tilt head rotations of $\pm 90°$, $\pm 75°$, $\pm 60°$ and $\pm 45°$.

positives. However, our CDPM achieves the second best performance in the continuous ROC curve. Our CDPM is only outperformed by the method of Chen et al. [24], which is boosted in the continues protocol by the simultaneous face detection and face alignment. At this point, our CDPM improves the TSM for more than 60% and VJ-MVFD for more than 45%. Our CDPM also outperforms the results obtained by Li et al. using SURF features [12,13]. To date, the best performances reported on the FDDB are the face detectors by Chen et al. [24] and Yen et al. [36], but our CDPM also outperforms these methods in the discrete scoring protocol by more than 2% and 5%, respectively. However, our CDPM is outperformed by these two methods when using the protocol of the continuous ROC curves.

In order to deal with the aforementioned challenge of overlapping rectangular bounding boxes with the ellipse annotations by the FDDB, the latest methods reported to this face benchmark used the same type of annotations [36–38]. This explicitly makes their models more efficient under the continuous ROC protocol because the features response maps help to optimize the face detectors for in plane rotations. Consequently, these face detectors give higher score to hypothesis faces with better alignment to the ellipse annotations. This was also implicitly learnt in the Joint Cascade Face Detection and Alignment method [24]. Thereby, these three methods [36–38] outperform our CDPM in the continuous ROC protocol while our CDPM outperforms them in the discrete ROC protocol, see Fig. 3. This means, they have better alignment to the ellipse annotations but no higher face detection rate.

Observe that our CDPM of $4 * (Roots + 6Parts)$ can recall up to 92.96% of the faces in the FDDB. By contrast, the TSM [19] method can at most recall 59.16% of the faces while getting the lowest TPR. Furthermore, we could confirm that our implementation of the VJ-MVFD can just perform as well as the VJ-OpenCV, which uses only a frontal classifier without filtering neighbouring detections. Unfortunately, the VJ-MVFD needed about two months of training and tuning parameters,

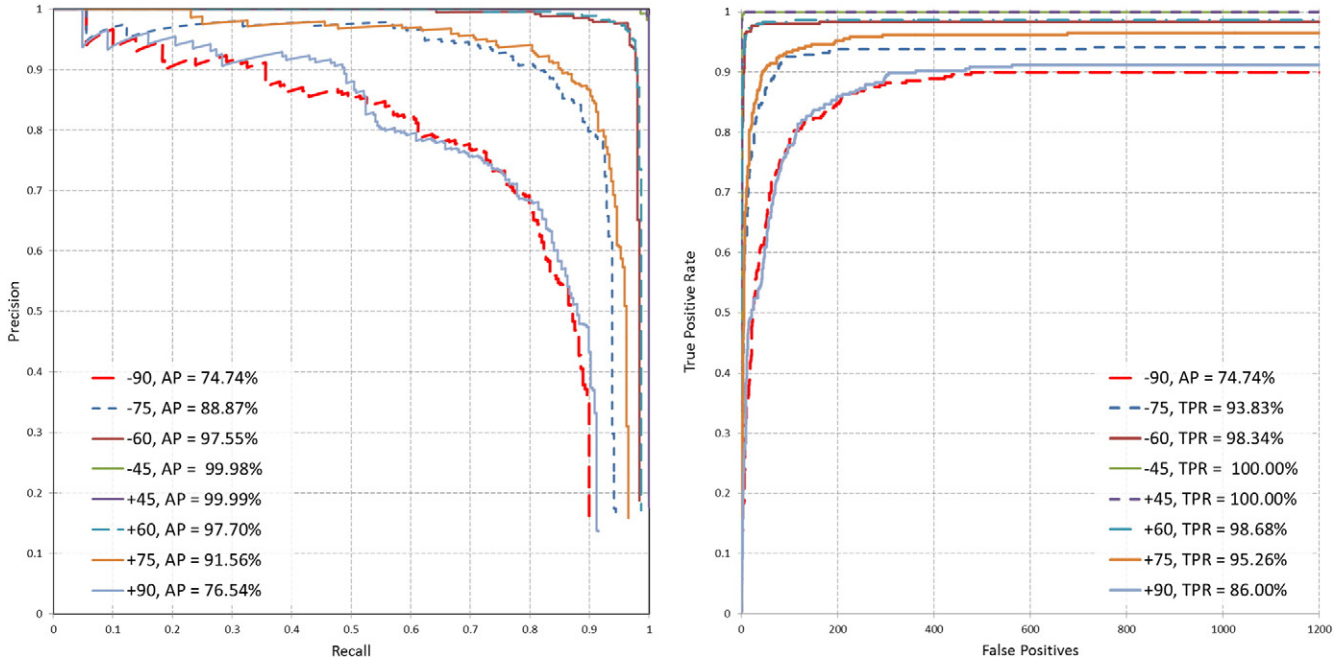

Fig. 7. Precision–Recall curves for the $4 * (Roots + 6Parts)$ CDPM-MVFD tested on the HPID database. Precision curves are displayed according to eight discrete pan head rotations. Likewise, ROC curves are also shown for the same head poses. The average precision is reported in the left graph, whereas the *TPR* is reported for at most $FP = 200$, right graph. To the best of our knowledge, this is the first validation of a MVFD method on annotated head pose datasets.

yet the results are very poor. This means that Haar-like features are not robust for detecting profile and partially occluded faces such those often included in the "in-the-wild" databases.

4.3. Experiments on AFLW

The AFLW database contains 24, 686 faces in 21, 328 images, with manually annotated facial landmarks. Like the FDDB database, the AFLW offers challenging faces in the wild but with a larger variety of backgrounds and resolutions. The database is released in three folders, such that testing images are taken from the first two folders and training images from the third folder. The testing set contains 14, 675 images and 17, 166 annotated faces.

In this experiment, we compare the face detection results of "Our CDPM" with the VJ-MVFD and the TSM face detector. Our CDPM is composed by 4 roots and 6 parts, which have been learned by collecting latent hard positives from the AFLW training images among other data. Like in the previous section, we measure performance according to the discrete ROC curves at the maximum TPR with at most $FP = 200$ (see Fig. 4). Again, our CDPM outperforms both VJ-MVFD and TSM face detectors with margins of 57% and 15%, respectively. Furthermore, our MVFD can recall up to 95.08% of the faces on the AFLW, while the VJ-MVFD and TSM can recall 65.12% and 78.01%, respectively.

The AFLW database contains faces with larger head pose variations and higher resolution than the FDDB. Therefore, the VJ-MVFD fails on profile faces whereas the TSM can deal with pose variation, but only with high resolution images. In contrast to the AFLW, the major challenges of the FDDB database are the low resolution and occluded faces. Consequently, the TSM performed better on the AFLW while VJ-MVFD performed better on the FDDB. In addition, both AFLW and FDDB differ regarding on the type of face annotations. While the FDDB is annotated with ellipses covering most of the facial skin, the AFLW is annotated with bounding boxes enclosing the facial landmarks (eyebrows, lips, nose and chin). We trained our CDPMs to achieve high TPR as well as high precision–recall, which was also shown in Fig. 2. See Fig. 5 for a some qualitative results of face detection examples obtained with the best of our MVFD models, the $4 * (Roots + 6Parts)$ CDPMs.

Fig. 8. MVFD on the COFW database. The face detection performance of our $4 * (Roots + 6Parts)$ CDPM was measured according to different levels of occlusion. This database consist of "in-the-wild" images labelled with facial landmarks and their corresponding occlusion label.

4.3.1. Face detection speed

Although we are not aiming at real-time performing face detection, we have obtained a MVFD that is comparable in speed to the VJ-MVFD. We tested our four CDPM face detectors, our implementation of the VJ-MVFD and the TSM in the 2845 FDDB images, which have an average resolution of 377 × 399 pixels. It is common to assess face detection speed in QVGA (320 × 240 pixels) images, but we rather prefer to avoid scaling the images in order to keep correspondence with the performance measurements. These experiments were run on a PC with Intel Xeon CPU E5-1620 @ 3.60 GHz, 16 GB of RAM, running Linux 64 bits. Caveat, Our CDPMs and the TSM have a bottle-neck convolution operation between a HOGs pyramid and the model. However, we use a convolution function provided by Felzenszwalb et al. [21], which makes both methods faster in Linux.

As can be seen from Table 2, the average detection time by our 4 ∗ (*Roots* + 6*Parts*) CDPM is comparable to the attained by VJ-MVFD. This face detection comparison is done in Matlab and we understand that VJ-MVFD is faster in C++ OpenCV. Though our CDPMs can also run in C++ using the OpenCV implementation for DPMs, both DPMs or CDPMs are faster in Matlab.

The average detection time is also affected by increasing the number of roots/parts in a CDPM. This also confirms that the TSM is not an efficient model when only face detection is sought, as it uses 13 mixture components and 68 part filters per component.

4.4. Experiments on HPID

To the best of our knowledge, there is no state-of-the-art face detector that has been rigorously validated w.r.t. the head pose variations. The TSM face detector [19] reported the performance of the model at face detection, head pose estimation and facial landmark detection, but separately.

Although both FDDB and AFLW face databases cover a wide spectrum of head rotations (in-plane and out-of-plane), it is difficult to report our face detection performance according to a specific range of head poses. AFLW was already segmented for training into subsets of discrete head poses, but the testing samples of this dataset have not been segmented as such. Note that the training process only requires a weakly labelled dataset, thereby, accuracy in head pose segmentation is not required to initialize the training of the first root filters.

Consequently, to evaluate face detection performance on a dataset with annotated head poses, we validate the performance of our 4 ∗ (*Roots* + 6*Parts*) CDPM on the Head Pose Image Database (HPID) [3]. This database contains 2790 monocular face images of 15 people photographed under pan and tilt rotations of ±90° with angle variations of ±15°. For this study, we use all images except the ones with annotated tilt angles beyond ±30° and pan-tilt angles below ±15°, which correspond to the trivial case of near frontal faces. Fig. 6 shows prototypical examples of faces in the HPID.

Fig. 7 presents a collection of eight precision–recall curves corresponding to pan head rotations. As can be seen from Fig. 6, these images are not particularly challenging due to the uniform background. Instead, the complexity arises by extreme head poses, and such challenge is certainly reflected in a slight precision decay of our CDPM at head poses of ±90°. Still, the precision remains around 90% for head poses of ±75°. Fig. 7 also shows the corresponding ROC curves for the same subsets of images, which confirms the high recall of our CDPM.

Note that our CDPM reports a different performance for symmetric head poses, which is expected given that our models are trained via LSVM combined with data-mining and bootstrapping (Section 2.1.2).

4.5. Experiments on COFW

The most common and challenging test in object detection is the reliability of the method to detect partially visible objects. Felzenszwalb et al. [21] proved the robustness of DPMs to detect partially occluded objects. However, the robustness to occlusions of multi-view face detectors has not been validated yet.

Burgos et al. [4] proposed a facial landmark detector with outstanding performance even when some of the facial features are not visible. To prove their concept, they annotated a collection of face images "in-the-wild", the COFW database. COFW contains 1852 images annotated with 68 facial landmarks and binary labels indicating whether a landmark is occluded. Authors made clear that their face alignment results are reproducible by simulating the output of a face detector. This allows them to overcome the issue of low precision and recall of available face detection methods.

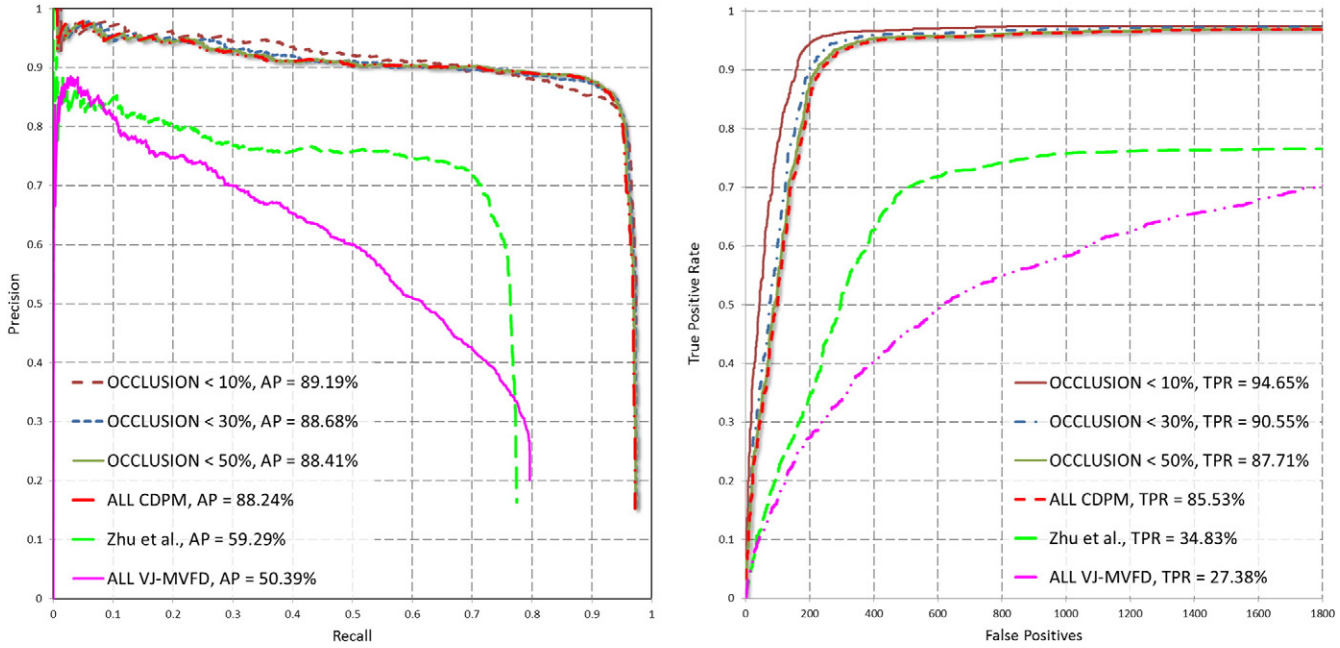

Fig. 9. CDPM-MVFD performance on the COFW database. Both Precision–Recall and ROC curves are disseminated according to the percentage of face occlusion. It is not possible to conclude whether our CDPM performs better upon the level of occlusion given that the imagery varies in resolution indistinctly.

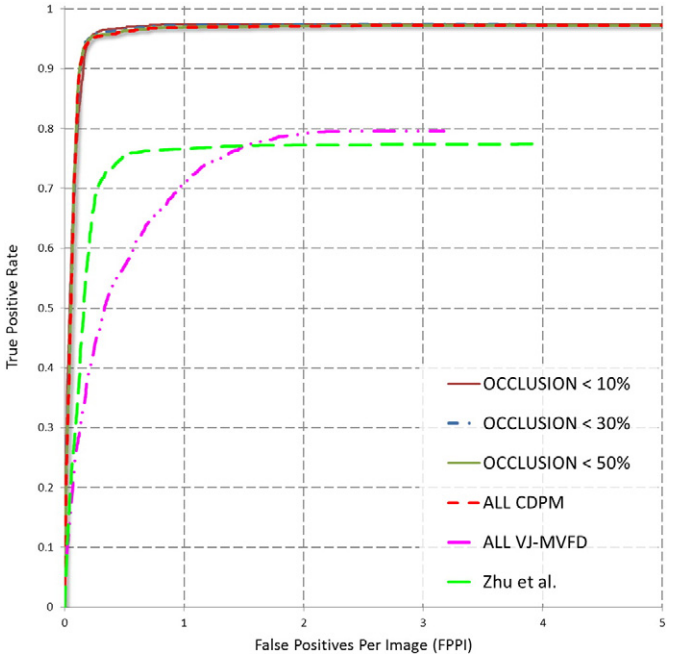

Fig. 10. TPR vs. FPPI in the COFW database. Both our CDPM and Zhu et al. [19] reach the maximum TPR with at most 1 FPPI, whereas our VJ-MVFD achieves the peak with more than 2 FPPI. Thus, our CDPM achieves a TPR of 97% at 1 FPPI.

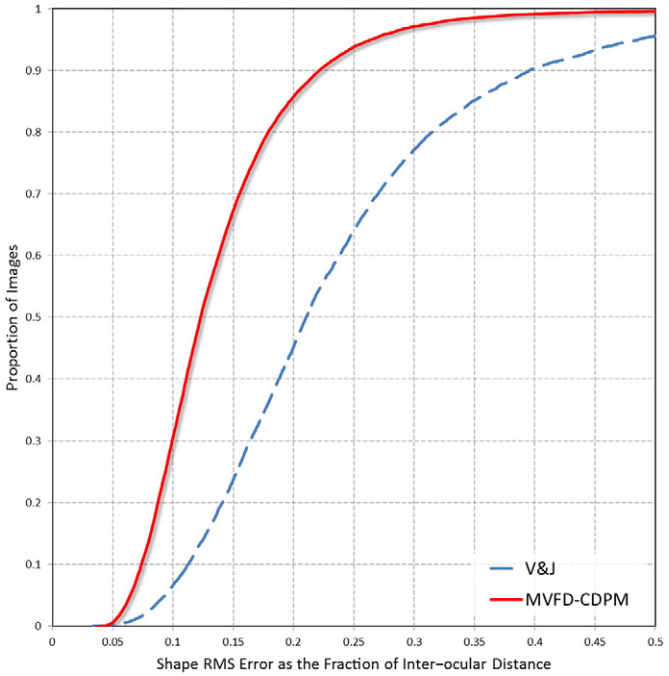

Fig. 12. Facial landmark initialization accuracy on the 300-W database. The component-wise mean shape and the ground truth are compared by measuring the shape RMS error as the fraction of interocular distance.

In this experiment, we tested our $4 * (Roots + 6Parts)$ CDPM in the COFW images (see an example gallery in Fig. 8). This shows that our CDPM can recall faces in the wild even when they render different levels of facial occlusion.

Specifically, it is possible to measure the face detection performance in this database by categorizing the results according to certain levels of occlusion. Fig. 9 shows the precision–recall curve corresponding to our CDPM without filtering any level of facial occlusion, "ALL CDPM". Subsequently, we report the precision–recall and the average precision for levels of occlusion such as <30%, >30%, <50% and >50%. Furthermore, we also present results for the TSM and our implementation of the VJ-MVFD in the COFW database. Observe that our CDPM outperforms by a large margin any of these state-of-the-art face detectors. Fig. 9 also shows the ROC curves of both our CDPM, TSM and VJ-MVFD. This confirms the outstanding results of our CDPM under different levels of facial occlusions.

Note that increasing the occlusion levels does not result in a decreased performance of our CDPM. This is because the scoring function of the face detector varies according to occlusion but also upon other factors such as image resolution, facial expressions, head pose, etc. These challenging image conditions can be found within any subset of the COFW database.

For a better appreciation of the robustness of our CDPM under occlusions, we also report the the true positive rate as function of the False Positives Per Image (FPPI) as commonly done in the PASCAL VOC [25], see Fig. 10. Its can be seen in this figure that our CDPM achieves a top TPR of 97% with at most 1 FPPI. Instead, Zhu et al. [19] only achieves a TPR of 77% at the same 1 FPPI, whereas our VJ-MVFD reaches a TPR of 80% at 3 FPPI.

4.6. CPDM-MVFD and face alignment

A recent challenge on facial landmark detection [39] managed to obtain the contribution of six participants. The 300-W challenge presented a collection of facial images in images in the wild, which contains similar issues to the ones we already tested against with the AFLW, FDDB and COFW databases. Using a semi-automatic methodology [40], the 300-W images were annotated with 68 facial landmarks related to eyebrows, eyes, nose, mouth and edge contours.

Facial landmark detectors are typically initialized based on the face detection output. To this end, a mean shape is fitted to the detected face bounding box. The mean shape model is usually learned by computing the mean of facial landmarks normalized by the bounding box given by the face detector. Thereby, the face detector output is the key to initialize the search of facial landmark locations.

Authors in [40] used a generic model of the TSM for face detection, which was retrained with images of faces in the wild. Thus, 300-W database was released with a face bounding box and facial landmarks annotations. Although this face detector was used to annotate images

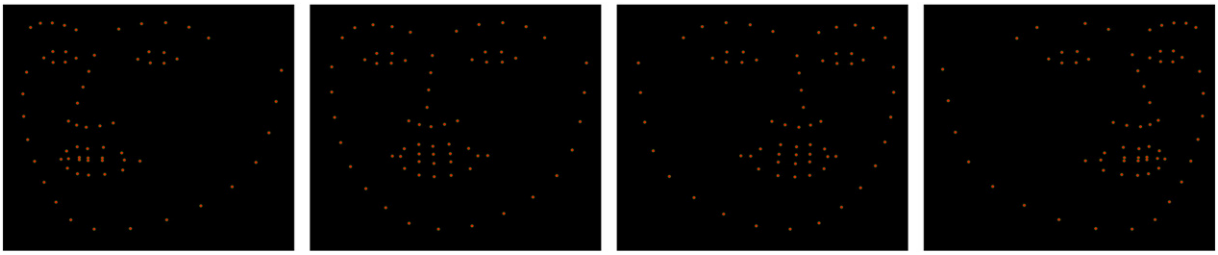

Fig. 11. Means shapes relative to the components of the $4 * (Roots + 6Parts)$ CDPM.

Fig. 13. Suitability of our CDPM for facial landmark detection. Mean shapes relative to the components of our $4 * (Roots + 6Parts)$ CDPM are fitted onto the detected bounding box accordingly. A component-wise initialization seems leading to faster and more accurate results, given that all face alignment methods are sensitive to initialization.

in the wild, it was not made available and there is no public reference to it. Furthermore, it is computationally demanding.

In this experiment, we outline the applicability of our MVFD to support the initialization of facial landmark detectors. To this end, we used the 300-W [39] dataset. First, all images were analysed by our $4 * (Roots + 6Parts)$ CDPM in order to detect the faces. Second, the CDPM detections were cross-referenced with the annotations in order to retain only the annotated faces in the 300-W database. Third, we split the 300-W images into four subsets corresponding to the CDPM face model components (root filter). Thus, the component with the highest detection score determines the subset. Finally, four mean shape models were learned from each subset of detections. The 68 facial landmarks are first centred w.r.t the estimated bounding box and then normalized by the width and the height of the same detection.

Fig. 11 shows the four mean shapes relative to the components of the $4 * (Roots + 6Parts)$ CDPM. As noted before, given that our MVFD is trained with latent examples, the four model components are not symmetric. Hence, the mean shapes are not symmetric either.

To test the suitability of our face detector for facial landmark detection, we adapt one of the four means shapes in Fig. 11 upon the CDPM component that produces the highest detection score. Subsequently, we compare the adapted mean shape and the ground truth by measuring the shape RMS error as the fraction of interocular distance, see Fig. 12.

In Fig. 12, we compare the facial landmark detection accuracy between a single mean shape model trained with VJ-MVFD and four mean shape models trained with our CDPM. It is understandable that none of the models will be comparable to the results reported in [39]. However, we aim to prove that our CDPM seems leading to a more accurate facial landmark detection. In one hand, a single mean shape of VJ-MVFD trained with face images exhibiting large head poses will lead to a sparse initialization model, i.e. large variance. Instead, the same sparse data will be segmented into four mean shapes when using our CDPM. Caveat, the robustness of each mean shape depends on the amount of available annotated data, which is an issue in the case of large head poses. On the other hand, our CDPM will ease the initialization for facial landmark detection on faces deploying a large head pose. This is still a limitation of any of the existing facial landmark detection methods.

Fig. 13 shows a gallery of examples of the CDPM mean shapes fitted to the detected bounding box. Given the proximity of the mean shapes to both face's centre and head pose, a faster facial landmark detection will be expected. This is assuming that the facial landmark detector has learned features from faces exhibiting large head poses. Otherwise, the best solution at large head poses will be close to the corresponding mean shape.

4.7. Experiments on ESOGU

The authors in [23] presented a face and facial landmark detection model, which also adopts the framework of structured models and DPMs to deal with this dual detection function. Root detectors of LBP-HOGs are used in cascade to perform face detection. Subsequently, a star model is hand crafted, which includes the face detection root filter and part filters dedicated to detecting eyes and mouth. Such approach is appealing when a richer face description is required, but it requires an more detailed annotated database. Furthermore, due to the similarity to the TSM models [19], this work may be also sensitive to low resolution images.

Consequently, the authors introduced a new "in-the-wild" image database collected from the web, ESOGU.[2] This database is split into two subsets of 285 and 382 images with a total of 2042 face bounding boxes annotated. One important characteristic of this dataset is the

[2] Available at http://mlcvdb.ogu.edu.tr/facedetection.html.

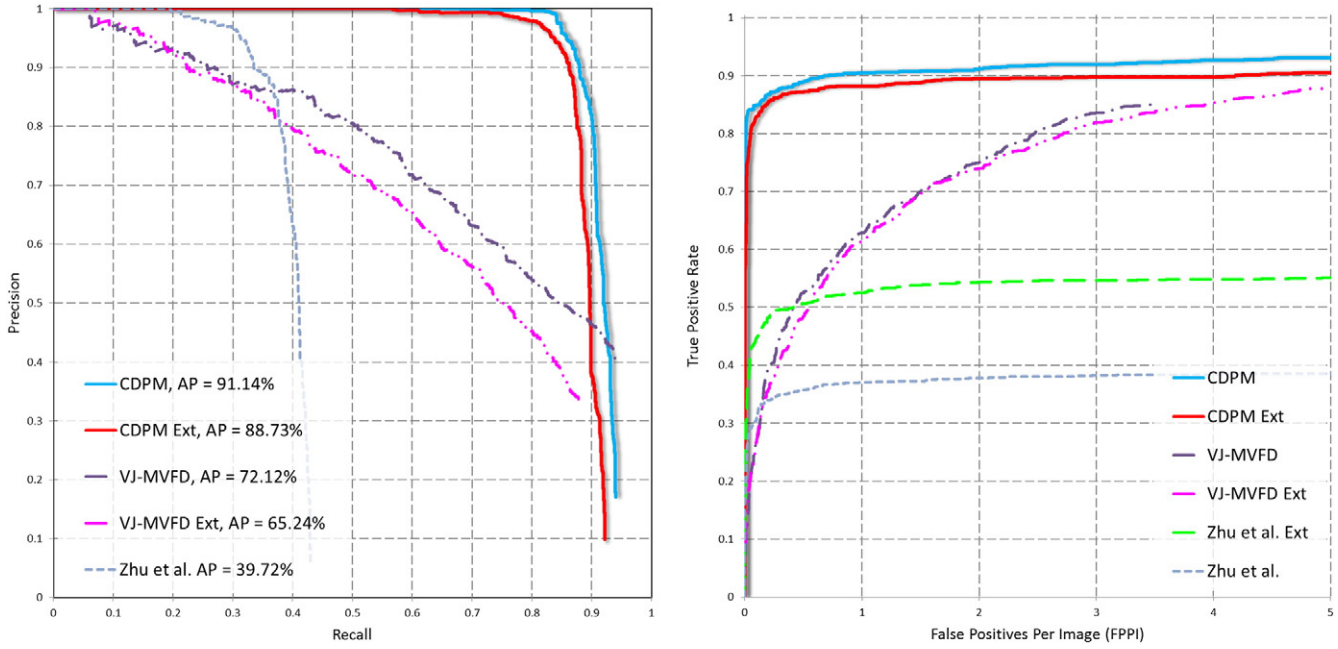

Fig. 14. Precision–Recall curves for the $4 * (Roots + 6Parts)$ CDPM-MVFD tested on the ESOGU database. Three models are reported for the two set of ESOGU images, 285 and 382 (Ext). The average precision is reported in the left graph, whereas the right graph shows the TPR as function of the FPPI. Our CDPM outperforms the other models in both ESOGU image sets.

inclusion of small-sized, lower resolution faces within the images. It is thus an important benchmark to highlight how a face detection-specific model yields a dramatic performance boost respect to equivalent models for combined face detection and landmarking.

We measured the face detection performance using our $4 * (Roots + 6Parts)$ CDPM, our VJ-MVFD and Zhu et al. [19] models. Fig. 14 shows the face detection results of these three models for the two ESOGU image sets. In terms of average precision (AP), our CDPM achieves 91.14% in the original ESOGU (285 images) and 87.73% in the ESOGU extension (referred as "Ext" an containing 382 images). Hence, our CDPM clearly outperforms the method proposed in [23], which reported an *AP* of 83.76% in ESOGU extended dataset.

5. Conclusions

We have presented a Multi-View Face Detector (MVFD) algorithm that is robust and accurate for "in-the-wild" scenarios. We adopted the object detection framework of Felzenszwalb et al. [26] to learn MVFD-DPM as well as the cascade version of it, CDPM-MVFD. We trained our models with weakly labelled data via Latent Support Vector Machines (LSVM). Hence, we showed the feasibility of learning models from a reduced number of labelled data. In order to increase the robustness of our MVFD, we combined LSVM with bootstrapping and data mining procedures. This post-processing procedures facilitate the incremental learning of models by progressively extending both positive and negative training sets.

We experimentally showed the performance of different variants of our CDPM depending on factors as the number of mixture components and part filters. This benchmark showed that models learned from a more detailed labelling or more granular part filters lead to lower precision.

We presented lengthy empirical performance analysis for face detection on a range of unconstrained and challenging databases. We compared our face detector against state-of-the-art methods. We showed that our CDPM model outperforms other state-of-the-art methods for face detection in in-the-wild scenarios by a large margin.

Furthermore, we also compared our CDPM-MVFD against the latest state-of-the-art face detectors tested on FDDB [1] benchmark dataset. In this context, our face detector also significantly outperforms these state-of-the-art methods by a large margin. Additionally, we compare against state-of-the-art methods such as the TSM by Zhu et al. [19] and our implementation of a VJ-MVFD [41] on the AFLW.

We also presented a specific per-head-pose face detection performance. To this end, we used the HPID [3] database, achieving an average precision over 95% for head poses up to $\pm 60°$, whereas head poses up to $\pm 90°$ can be detected with an average precision of 75%.

We showed that the cascade search of our face detection models is much faster than that reported by Zhu et al. [19], and of comparable speed to that of VJ-MVFD, achieving close to real-time performance.

Since face detection is often followed by facial landmark detection, we showed how our face detector can support a facial landmark detection process. By simply fitting a mean shape model relative to the face detection bounding box, our CDPM shows promising results in the 300-W database. We compare the accuracy of a single mean shape trained with the VJ-MVFD against four component-wise mean shapes trained with our CDPM. According to the standard MRS error, our face detector achieved higher accuracy than VJ-MVFD. This result indicates that a view-based facial landmark detector using a CDPM-MVFD will have better chances of outstanding results.

Lastly, we provide Matlab code for reproducing our experiments. It can be found at http://ibug.doc.ic.ac.uk/resources.[3]

References

[1] V. Jain, E. Learned-Miller, Technical Report, FDDB: A Benchmark for Face Detection in Unconstrained Settings, University of Massachusetts, Amherst, 2010.
[2] M. Koestinger, P. Wohlhart, P. Roth, H. Bischof, Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization, ICCV, IEEE 2011, pp. 2144–2151.
[3] N. Gourier, D. Hall, J. Crowley, Estimating face orientation from robust detection of salient facial structures, CVPR, IEEE 2004, pp. 1–9.
[4] X.P. Burgos-Artizzu, P. Perona, P. Dollár, Robust face landmark estimation under occlusion, ICCV, IEEE 2013, pp. 57–64.
[5] C. Zhang, Z. Zhang, A survey of recent advances in face detection, Technical Report , Microsoft Research, 2010.
[6] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, CVPR, IEEE, vol. 1 2001, pp. 1–511.

[3] This will be available once this manuscript is accepted for publication, however it will be provided to reviewers as supporting material.

[7] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-pie, Image Vis. Comput. 28 (2010) 807–813.
[8] M. Jones, P. Viola, Fast multi-view face detection, Mitsubishi Electric Research Lab TR-20003-96 3, 2003.
[9] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, ECCV, IEEE 2006, pp. 117–121.
[10] B. Wu, H. Ai, C. Huang, S. Lao, Fast rotation invariant multi-view face detection based on real adaboost, FG, IEEE 2004, pp. 79–84.
[11] C. Huang, H. Ai, Y. Li, S. Lao, Vector boosting for rotation invariant multi-view face detection, ICCV, IEEE, vol. 1 2005, pp. 446–453.
[12] J. Li, T. Wang, Y. Zhang, Face detection using SURF cascade, ICCV-W, IEEE 2011, pp. 2183–2190.
[13] J. Li, Y. Zhang, Learning SURF cascade for fast and accurate object detection, CVPR, IEEE 2013, pp. 3468–3475.
[14] B. Jun, D. Kim, Robust face detection using local gradient patterns and evidence accumulation, Pattern Recogn. 45 (2012) 3304–3316.
[15] M. Toews, T. Arbel, Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 1567–1581.
[16] S. Anvar, W.-Y. Yau, E.K. Teoh, Multiview face detection and registration requiring minimal manual intervention, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 2484–2497.
[17] V. Jain, E. Learned-Miller, Online domain adaptation of a pre-trained cascade of classifiers, CVPR, IEEE 2011, pp. 577–584.
[18] H. Li, G. Hua, Z. Lin, J. Brandt, J. Yang, Probabilistic elastic part model for unsupervised face detector adaptation, ICCV, IEEE 2013, pp. 1–8.
[19] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, CVPR, IEEE 2012, pp. 2879–2886.
[20] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, CVPR, IEEE, vol. 1 2005, pp. 886–893.
[21] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1627–1645.
[22] J. Yan, X. Zhang, Z. Lei, S.Z. Li, Face detection by structural models, Image Vis. Comput. 32 (10) (2014) 790–799.
[23] H. Cevikalp, B. Triggs, V. Franc, Face and landmark detection by using cascade of classifiers, FG, IEEE 2013, pp. 1–7.
[24] D. Chen, S. Ren, Y. Wei, X. Cao, J. Sun, Joint cascade face detection and alignment, ECCV, Springer 2014, pp. 109–122.
[25] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL VOC2012 Results, 2012.
[26] P. Felzenszwalb, R. Girshick, D. McAllester, Cascade object detection with deformable part models, CVPR, IEEE 2010, pp. 2241–2248.
[27] T. Kanade, J. Cohn, Y. Tian, Comprehensive database for facial expression analysis, FG, IEEE 2000, pp. 46–53.
[28] A. Battocchi, F. Pianesi, D. Goren-Bar, Dafex: database of facial expressions, Intell. Technol. Interact. Entertain. (2005) 303–306.
[29] F. Wallhoff, Fgnet-Facial Expression and Emotion Database, Technische Universität München, 2004.
[30] M. Pantic, M. Valstar, R. Rademaker, L. Maat, Web-based database for facial expression analysis, ICME, IEEE 2005, pp. 317–321.
[31] W. Junek, Mind reading: the interactive guide to emotions, J. Can. Acad. Child Adolesc. Psychiatry 16 (2007) 182.
[32] J. Orozco, O. Rudovic, J. Gonzàlez, M. Pantic, Hierarchical on-line appearance-based tracking for 3d head pose, eyebrows, lips, eyelids and irises, Image Vis. Comput. 31 (2013) 322–340.
[33] G. Bradski, et al., The opencv library, Dr. Dobb's J. Softw. Tools 25 (11) (2000) 120–126.
[34] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, Pattern Recogn. (2003) 297–304.
[35] V. Jain, FDDB Results, http://vis-www.cs.umass.edu/fddb/results.html 2012.
[36] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, CVPR, IEEE 2015, pp. 5325–5334.
[37] M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool, Face detection without bells and whistles, ECCV, Springer 2014, pp. 720–735.
[38] S.S. Farfade, M. Saberian, L. Li, Multi-view face detection using deep convolutional neural networks, ICMR, Shanghai, China, 2015.
[39] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, 300 faces in-the-wild challenge: the first facial landmark localization challenge, ICCV-W, IEEE 2013, pp. 397–403.
[40] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, A semi-automatic methodology for facial landmark annotation, CVPR-W, IEEE 2013, pp. 896–903.
[41] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, CVPR, IEEE, vol. 1 2001, pp. 511–518.