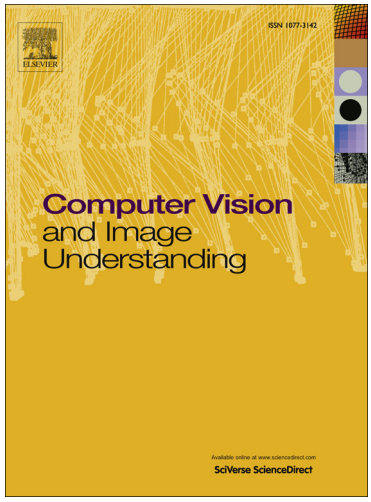
Accepted Manuscript



A Survey on Face Detection in the wild: past, present and future

Stefanos Zafeiriou, Cha Zhang, Zhengyou Zhang

PII: S1077-3142(15)00072-7
DOI: http://dx.doi.org/10.1016/j.cviu.2015.03.015
Reference: YCVIU 2241

To appear in: *Computer Vision and Image Understanding*

Received Date: 16 September 2014
Accepted Date: 27 March 2015

Please cite this article as: S. Zafeiriou, C. Zhang, Z. Zhang, A Survey on Face Detection in the wild: past, present and future, *Computer Vision and Image Understanding* (2015), doi: http://dx.doi.org/10.1016/j.cviu.2015.03.015

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

CVIU 00 (2015) 1–1

CVIU

A Survey on Face Detection in the wild: past, present and future

Stefanos Zafeiriou, Cha Zhang and Zhengyou Zhang

Cha Zhang and Zhengyou Zhang are with Microsoft Research
Microsoft Corporation, One Microsoft Way,
Redmond WA 98052-6399, USA
emails: {zhang@microsoft.com,chazhang@microsoft.com}
Corresponding author:
Stefanos Zafeiriou
Visual Information Processing
Department of Computing
Imperial College London
Room: 375, Floor: 3, Huxley Building
South Kensington Campus
London, UK,
Post Code: SW7 2AZ
tel: +44-207-594-8461
fax : +44-207-581-8024
email:{s.zafeiriou@imperial.ac.uk}

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Available online at www.sciencedirect.com

ScienceDirect

CVIU 00 (2015) 1–33

CVIU

A Survey on Face Detection in the wild: past, present and future

Stefanos Zafeiriou, Cha Zhang and Zhengyou Zhang

Abstract

Face detection is one of the most studied topics in computer vision literature, not only because of the challenging nature of face as an object, but also due to the countless applications that require the application of face detection as a first step. During the past 15 years, tremendous progress has been made due to the availability of data in unconstrained capture conditions (so-called 'in-the-wild') through the Internet, the effort made by the community to develop publicly available benchmarks, as well as the progress in the development of robust computer vision algorithms. In this paper, we survey the recent advances in real-world face detection techniques, beginning with the seminal Viola-Jones face detector methodology. These techniques are roughly categorized into two general schemes: rigid templates, learned mainly via boosting based methods or by the application of deep neural networks, and deformable models that describe the face by its parts. Representative methods will be described in detail, along with a few additional successful methods that we briefly go through at the end. Finally, we survey the main databases used for the evaluation of face detection algorithms and recent benchmarking efforts, and discuss the future of face detection.

© 2014 Published by Elsevier Ltd.

Keywords: face detection, feature extraction, boosting, deformable models, deep neural networks.

1. Introduction

Automatic face detection is the cornerstone of all applications revolving around automatic facial image analysis including, but not limited to, face recognition and verification [1], face tracking for surveillance [2], facial behaviour analysis [3], facial attribute recognition [4] (i.e., gender/age recognition [5] and assessment of beauty [6]), face relighting and morphing [7], facial shape reconstruction [8], image and video retrieval, as well as organization and presentation of digital photo-albums [9]. Face detection is also the initial step to all modern vision-based human-computer and human-robot interaction systems (e.g., the recent commercial robots such as Nao come with an embedded face detection module [10]). Furthermore, the majority of the commercial digital cameras have an embedded face detector that is used to help auto-focusing. Finally, many social networks, such as FaceBook, use face detection mechanisms for the purpose of image/person tagging.

Automatic face and facial feature detection was one of the first computer vision applications with early works dating back 45–50 years [11, 12, 13, 14]. During the middle of 1990's until the beginning of 2000's, the field witnessed an explosion [15, 16]. Unfortunately, the majority of these early works were not able to provide good performance in unconstrained conditions (so called "in-the-wild", please see Fig. 1 for some examples of "in-the-wild" faces), and thus were not appropriate to be directly applied in real-world settings. This was achieved by the seminal work of Viola and Jones [17] on boosting based face detection, which was the first algorithm that made face detection practically feasible in real-world applications and until today is widely applied in digital cameras and photo organization software. Since then, research in face detection has made significant progress in the direction of

ACCEPTED MANUSCRIPT

providing algorithms that are able to detect faces 'in-the-wild'. Except of the rapid growth in processing power and storage capacity of the modern computers, modern face detection algorithms have also benefited from:

- The introduction of robust feature extraction methodologies, such as Scale Invariant Feature Transform (SIFT) features [18], Histograms of oriented Gradients (HoGs) [19], Local Binary Patterns (LBPs) and their variations [20, 21, 22], their fast counterparts such as Speeded Up Robust Features (SURF) [23] and DAISY [24], as well as transformations that combine the above features with integral images, such as Integral Channel Features (ICF) [25, 26]. These features are densely or sparsely sampled and are used to describe face appearance.
- The efforts of the research community to develop databases and benchmarks for "in-the-wild" object detection, such as PASCAL [27, 28], LFW [29], FDDB [30] etc. The development of these large collections was facilitated by the abundance of visual data spread via the major breakthrough of the Internet (mostly through data sharing services and search engines).
- The development of many powerful, mainly discriminative methodologies such as Boosting [31], Support Vector machines [32] and their recent structural extension [33] and (deep) neural network architectures [34] which can harness the amount of information provided by the above mentioned modern large scale datasets and facilitate training of complex deformable object and facial models [35].
- The development of repositories of high quality publicly available code. Notable examples include the OpenCV series of releases [36], the series of deformable parts-based models releases [37], as well as the recent efforts to make a high quality repository of convolutional neural network architectures [38].

The last comprehensive survey of face detection algorithms in [15] grouped the various methods into four categories: knowledge-based methods, feature invariant approaches, template matching methods, and appearance-based methods. Knowledge-based methods use pre-defined rules based on human knowledge in order to detect a face, feature invariant approaches aim to find face structure features robust to pose and lighting variations, template matching methods use pre-stored face templates to determine where a human face is depicted in an image, while appearance-based methods learn face models from a set of representative training face images which are used for face detection. The above categorization, however, hardly applies on recent methodologies developed since [17]. Thus in this survey we attempt to organize algorithms in the following two major categories:

- The family of algorithms that are based on rigid-templates that include:
  - variations of boosting; the main representative of this family of algorithms include the Viola-Jones face detection algorithm and its variations [17, 39].
  - algorithms that are based on Convolutional Neural Networks (CNNs) and Deep CNNs (DCNNs) [34, 40]; recently, a DCNN showed exceptional performance in multi-object class detection [41], thus currently its learning architectures have been investigated for face detection [42].
  - methods such as [43, 44] which apply strategies inspired by image-retrieval and Generalized Hough Transform [45, 46].
- The family of algorithms that learn and apply a Deformable Parts-based Model (DPM) [35, 47, 14] to model a potential deformation between facial parts. These methods can also combine face detection and facial part localization [48]. This family of algorithms mainly revolves around extensions and variations of the general object detection methodology [35, 47]. Other notable parts-based methods include [49, 50, 51].

Hence, the aims of the paper are

- thoroughly describe variations and extensions of face detection algorithms based on rigid templates (i.e., cascades of weak classifiers using boosting, DCNNs etc.), as well as deformable templates
- describe the current benchmarks and metrics used for evaluation, as well as compare the performance of the state-of-the-art

ACCEPTED MANUSCRIPT

- perform a critical comparison between the two families of algorithms, i.e. based on rigid and non-rigid templates, as well as discuss future challenges and research directions in face detection.

The remaining of the manuscript is outlined as follows. Section 2 surveys the approaches based on learning rigid-templates, including (a) boosting-based algorithms, (b) DCNNs and (c) notable approaches that explot rigid-templates. In Section 3, we survey pictorial structures, DPMs and other part-based methods which evolved to be some of the state-of-the-art methodologies for face detection. In Section 4, we survey the current benchmarks for 'in-the-wild' face detection. Finally, Section 5 concludes the survey and discusses future challenges.

2. Face Detection Algorithms with Rigid Templates

In this Section we survey face detection algorithms that are based on learning a set of rigid-templates. The main line of research in this direction is based on learning rigid templates using boosted cascades of classifiers. Hence, we start by providing an overview of the Viola-Jones face detector, which also motivates many of the recent advances in face detection. Then, solutions to two key issues for boosting-based face detection: what features to extract, and which learning algorithm to apply, are surveyed in Section 2.2 and Section 2.3 learning algorithms, respectively. These Sections of the paper are largely based on the technical report [52].

Another line of research on rigid-templates, which currently gains momentum, is based on Deep Convolutional Neural Networks (DCNNs). DCNNs-based approaches are surveyed in Section 2.4. Finally, in Section 2.5, we survey other approaches that do not strictly fall to the above categories.

2.1. The Viola-Jones Face Detector

If one were asked to name the face detection algorithm that had the most impact in the 2000's, it will most likely be the seminal work by Viola and Jones [17]. The Viola-Jones face detector contains three main ideas that make it possible to build and run *in real time*: the integral image, classifier learning with AdaBoost, and the attentional cascade structure.

2.1.1. The Integral Image

The integral image, also known as a summed area table, is an algorithm for quickly and efficiently computing the sum of values in a rectangle subset of a grid. It was first introduced to the computer graphics field in [53] to be used in mipmaps. Viola-Jones face detector applied the integral image for rapid computation of Haar-like features, as detailed below.

An integral image is constructed as follows:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \tag{1}$$

where $ii(x, y)$ is the integral image at pixel location (x, y) and $i(x', y')$ is the original image. Using the integral image to compute the sum of any rectangular area is extremely efficient, as shown in Fig. 2. The sum of pixels in the rectangle region $ABCD$ can be calculated as:

$$\sum_{(x,y) \in ABCD} i(x, y) = ii(D) + ii(A) - ii(B) - ii(C), \tag{2}$$

which only requires four array references.

The integral image can be used to compute simple Haar-like rectangular features, as shown in Fig. 2 (a-f). The features are defined as the (weighted) intensity difference between two to four rectangles. For instance, in Fig. 2 (a), the feature value is the average difference of pixel values in the gray and white rectangles. Since the rectangles share corners, the computation of two rectangle features (a and b) requires six array references, the three rectangle features (c and d) require eight array references, and the four rectangle features (e and f) require nine array references.

ACCEPTED MANUSCRIPT

2.1.2. AdaBoost Learning

Boosting is a method of finding a highly accurate hypothesis by combining many "weak" hypotheses, each with moderate accuracy. For an introduction on boosting, we refer the readers to [54] and [55].

The AdaBoost (Adaptive Boosting) algorithm is generally considered as the first step towards more practical boosting algorithms [56, 57]. In this section, following [58] and [55], we briefly present a generalized version of AdaBoost algorithm, usually referred to as *RealBoost*. It has been advocated in various works [59, 60, 61, 62] that RealBoost yields better performance than the original AdaBoost algorithm.

Consider a set of training examples as $\mathcal{S} = \{(x_i, z_i), i = 1, \cdots, N\}$, where x_i belongs to a domain or instance space $\mathcal{X}$, and z_i belongs to a finite label space $\mathcal{Z}$. In binary classification problems, $\mathcal{Z} = \{1, -1\}$, where $z_i = 1$ for positive examples and $z_i = -1$ for negative examples. AdaBoost produces an additive model $F^T(x) = \sum_{t=1}^{T} f_t(x)$ to predict the label of an input example x, where $F^T(x)$ is a real valued function in the form $F^T : \mathcal{X} \to \mathbb{R}$. The predicted label is $\hat{z}_i = \text{sign}(F^T(x_i))$, where sign$(\cdot)$ is the sign function. From the statistical view of boosting [55], AdaBoost algorithm fits an additive logistic regression model by using adaptive Newton updates for minimizing the expected exponential criterion:

$$L^T = \sum_{i=1}^{N} \exp\{-z_i F^T(x_i)\}. \tag{3}$$

The AdaBoost learning algorithm tries to find the best additive base function $f_{t+1}(x)$ once $F^t(x)$ is given. For this purpose, we assume the base function pool $\{f(x)\}$ is in the form of confidence rated decision stumps. That is, a certain form of real feature value $h(x)$ is first extracted from x, $h : \mathcal{X} \to \mathbb{R}$. For instance, in the Viola-Jones face detector, $h(x)$ are the Haar-like features computed with integral image, as was shown in Fig. 2 (a-f). A decision threshold H divides the output of $h(x)$ into two subregions, v_1 and v_2, $v_1 \cup v_2 = \mathbb{R}$. The base function $f(x)$ is thus:

$$f(x) = c_j, \text{if } h(x) \in v_j, j = 1, 2, \tag{4}$$

which is often referred to as the stump classifier. c_j is called the confidence. The optimal values of the confidence values can be derived as follows. For $j = 1, 2$ and $k = 1, -1$, let

$$W_{kj} = \sum_{i: z_i = k, f(x_i) \in v_j} \exp\{-kF^t(x_i)\}. \tag{5}$$

The target criterion can thus be written as:

$$L^{t+1} = \sum_{j=1}^{2} \left[W_{+1j} e^{-c_j} + W_{-1j} e^{c_j} \right]. \tag{6}$$

Using standard calculus, we see L^{t+1} is minimized when

$$c_j = \frac{1}{2} \ln \left(\frac{W_{+1j}}{W_{-1j}} \right). \tag{7}$$

Plugging into (6), we have:

$$L^{t+1} = 2 \sum_{j=1}^{2} \sqrt{W_{+1j} W_{-1j}}. \tag{8}$$

(8) is referred to as the Z score in [58]. In practice, at iteration $t + 1$, for every Haar-like feature $h(x)$, we find the optimal threshold H and confidence score c_1 and c_2 in order to minimize the Z score L^{t+1}. A simple pseudo code of the AdaBoost algorithm is shown in Fig. 3.

2.1.3. The Attentional Cascade Structure

The attentional cascade is a critical component in the Viola-Jones detector. The key insight is that smaller, and thus more efficient, boosted classifiers can be built which reject most of the negative sub-windows while keeping

ACCEPTED MANUSCRIPT

almost all the positive examples. Consequently, majority of the sub-windows will be rejected in early stages of the detector, making the detection process extremely efficient.

The overall process of classifying a sub-window thus forming a degenerate decision tree, which is called a "cascade" was presented in [17]. As shown in Fig. 4, the input sub-windows pass a series of nodes during detection. Each node will make a binary decision according to which the window will be kept for the next round or rejected immediately. The number of weak classifiers usually increases as the number of nodes a sub-window passes. For instance, in [17], the first five nodes contain 1, 10, 25, 25, 50 weak classifiers, respectively. This is intuitive, since each node is trying to reject a certain amount of negative windows while keeping all the positive examples, and the task becomes harder at late stages. Having fewer weak classifiers at early stages also improves the speed of the detector.

The cascade structure also has an impact on the training process. Face detection is a rare event detection task. Consequently, there are usually billions of negative examples needed in order to train a high performance face detector. To handle the huge amount of negative training examples, the Viola-Jones face detector [17] used a bootstrap process. That is, at each node, a threshold was manually chosen, and the partial classifier was used to scan the negative example set to find more unrejected negative examples for the training of the next node. Furthermore, each node is trained independently, as if the previous nodes does not exist. One argument behind such a process is to force the addition of some nonlinearity in the training process, which could improve the overall performance. However, recent works showed that it is actually beneficial not to completely separate the training process of different nodes, as will be discussed in Section 2.3.

In [17], the attentional cascade is constructed manually. That is, the number of weak classifiers and the decision threshold for early rejection at each node are both manually specified. This is a non-trivial task. If the decision thresholds were set too aggressively, the final detector will be very fast, but the overall detection rate may be affected. On the other hand, if the decision thresholds are set very conservatively, most sub-windows will need to pass through many nodes, making the detector very slow. Combined with the limited computational resources available in the early 2000's, it is no wonder that training a good face detector can take months of fine-tuning.

2.2. Feature Extraction

As mentioned earlier, thanks to the rapid expansion in storage and computation resources, appearance based methods have dominated the recent advances in face detection. The general practice is to collect a large set of face and non-face examples, and adopt certain machine learning algorithms to learn a face model to perform classification. There are two key issues in this process: what features to extract, and which learning algorithm to apply. In this section, we first review the recent advances in feature extraction.

The Haar-like rectangular features as in Fig. 2 (a-f) are very efficient to compute due to the integral image technique, and provide good performance for building frontal face detectors. In a number of follow-up works, researchers extended the straightforward features with more variations in the ways rectangle features are combined.

For instance, as shown in Fig. 5, the feature set of [17] was generalized in [63] by introducing 45 degree rotated rectangular features (a-d), and center-surround features (e-f). In order to compute the 45 degree rotated rectangular features, a new rotated summed area table was introduced as:

$$rii(x, y) = \sum_{x' \le x, |y-y'| \le x-x'} i(x', y'). \tag{9}$$

As seen in Fig. 5, $rii(A)$ is essentially the sum of pixel intensities in the shaded area. The rotated summed area table can be calculated with two passes over all pixels.

A number of researchers noted the limitation of the original Haar-like feature set in [17] for multi-view face detection, and proposed to extend the feature set by allowing a more flexible combination of rectangular regions. For instance, in [59], three types of features were defined in the detection sub-window, as shown in Fig. 6 (a). The rectangles are of flexible sizes $x \times y$ and at certain distances of (dx, dy) apart. It was argued that these features can be non-symmetrical to cater to non-symmetrical characteristics of non-frontal faces. Viola and Jones [64] also proposed a similar feature called diagonal filters, as shown in Fig. 6 (b). These diagonal filters can be computed with 16 array references to the integral image.

ACCEPTED MANUSCRIPT

In [65] the Haar-like feature set was further extended to work on motion filtered images for video-based pedestrian detection. Let the previous and current video frames be i_{t-1} and i_t. Five motion filters are defined as:

$$\begin{aligned} \Delta &= |i_t - i_{t-1}| \\ U &= |i_t - i_{t-1} \uparrow| \\ L &= |i_t - i_{t-1} \leftarrow| \\ R &= |i_t - i_{t-1} \rightarrow| \\ D &= |i_t - i_{t-1} \downarrow| \end{aligned}$$

where $\{\uparrow, \leftarrow, \rightarrow, \downarrow\}$ are image shift operators. $i_t \uparrow$ is i_t shifted up by one pixel. In addition to the regular rectangular features (Fig. 2) on these additional motion filtered images, the method in [65] proposed to add single box rectangular sum features, and new features across two images. For instance:

$$f_i = r_i(\Delta) - r_i(S), \tag{10}$$

where $S \in \{U, L, R, D\}$ and $r_i(\cdot)$ is a single box rectangular sum within the detection window.

It should be noted here that the construction of the motion filtered images $\{U, L, R, D\}$ is not scale invariant. That is, when detecting pedestrians at different scales, these filtered images need to be recomputed. This can be done by first constructing a pyramid of images for i_t at different scales and computing the filtered images at each level of the pyramid, as was done in [65].

In [62] joint Haar-like features were proposed, which are based on the co-occurrence of multiple Haar-like features. It was claimed that feature co-occurrence can better capture the characteristics of human faces, making it possible to construct a more powerful classifier. As shown in Fig. 7, the joint Haar-like feature uses a similar feature computation and thresholding scheme, however, only the binary outputs of the Haar-like features are concatenated into an index for 2^F possible combinations, where F is the number of combined features. To find distinctive feature co-occurrences with limited computational complexity, the suboptimal sequential forward selection scheme was used in [62]. The number F was also heuristically limited to avoid statistical unreliability.

To some degree, the above joint Haar-like features resemble a CART tree, which was explored in [66]. It was shown that CART tree based weak classifiers improved results across various boosting algorithms with a small loss in speed. Another variation was presented in [61] for improving the weak classifier, where the use of a single Haar-like feature and equally binning of the feature values into a histogram to be used in a RealBoost learning algorithm was proposed. Similar to the number F in the joint Haar-like features, the number of bins for the histogram is vital to the performance of the final detector. [61] proposed to use 64 bins. In the subsequent work [67], it was specifically pointed out that too fine granularity of the histogram may cause over-fitting. To deal with that it was suggested to use fine granularity in the first few layers of the cascade, and coarse granularity in latter layers. Another interesting recent method was proposed in [68], where a new weak classifier called Bayesian stump was introduced. Bayesian stump is also a histogram based weak classifier, however, the split thresholds of the Bayesian stump are derived from iterative split and merge operations instead of being at equal distances and fixed. Experimental results showed that such a flexible multi-split thresholding scheme is effective in improving the detector's performance.

Another limitation of the original Haar-like feature set is its lack of robustness in handling faces under extreme lighting conditions, despite the fact that Haar features are usually normalized by the test windows' intensity covariance [17]. In [69] a modified census transform was adopted to generate illumination-insensitive features for face detection. On each pixel's 3×3 neighborhood, the method applied a modified census transform that compares the neighborhood pixels with their intensity mean. The results are concatenated into an index number representing the pixel's local structure. During boosting, the weak classifiers are constructed by examining the distributions of the index numbers for the pixels. Another well-known feature set robust to illumination variations is the local binary patterns (LBP) [20], which have been very effective for face recognition tasks [21, 70]. In [71, 72], LBP were applied for face detection tasks under a Bayesian and a boosting framework, respectively. More recently, inspired by LBP, in [73] locally assembled binary feature was proposed, which showed great performance on standard face detection data sets.

In order to explore possibilities for further improving the performance, more and more complex features were proposed in the literature. For instance, in [74] generic linear features were introduced, which are defined by a

ACCEPTED MANUSCRIPT

mapping function $\phi() : \mathbb{R}^d \rightarrow \mathbb{R}^1$, where d is the size of the test patch. For linear features, $\phi(x) = \phi^T x, \phi \in \mathbb{R}^d$. The classification function is in the following form:

$$F^T(x) = \text{sign}[\sum_t^T \lambda_t(\phi_t^T x)], \tag{11}$$

where $\lambda_t()$ are $\mathbb{R} \rightarrow \mathbb{R}$ discriminating functions, such as the conventional stump classifiers in AdaBoost. $F^T(x)$ shall be 1 for positive examples and -1 for negative examples. Note the Haar-like feature set is a subset of linear features. Another example is the anisotropic Gaussian filters in [75]. In [76], the linear features were constructed by pre-learning them using local non-negative matrix factorization (LNMF), which is still sub-optimal. Instead, Liu and Shum [74] proposed to search for the linear features by examining the Kullback-Leibler (KL) divergence of the positive and negative histograms projected on the feature during boosting (hence the name Kullback-Leibler boosting). In [77], the authors proposed to apply Fisher discriminant analysis and more generally recursive nonparametric discriminant analysis (RNDA) to find the linear projections ϕ_t. Linear projection features are very powerful features. The selected features shown in [74] and [77] were like face templates. They may significantly improve the convergence speed of the boosting classifier at early stages. However, caution must be taken to avoid overfitting if these features are to be used at the later stages of learning. In addition, the computational load of linear features is generally much higher than the traditional Haar-like features. On the contrary, in [78] the use of simple pixel pairs as features, and in [79] the use of the relative values of a set of control points as features, was proposed. Such pixel-based features can be computed even faster than the Haar-like features, however, their discrimination power is generally insufficient to build high performance detectors.

Another popular complex feature for face/object detection is based on regional statistics such as histograms. In [80] local edge orientation histograms was proposed, which compute the histogram of edge orientations in subregions of the test windows. These features are then selected by an AdaBoost algorithm to build the detector. The orientation histogram is largely invariant to global illumination changes, and it is capable of capturing geometric properties of faces that are difficult to capture with linear edge filters such as Haar-like features. However, similar to motion filters, edge based histogram features are not scale invariant, hence one must first scale the test images to form a pyramid to make the local edge orientation histograms features reliable. Later, in [19] a similar scheme called histogram of oriented gradients (HoG) was proposed, which became a very popular feature for human/pedestrian detection [81, 82, 83, 84, 85] (we will discuss about the use of HoG features in face detection in the next subsection). In [86], the authors proposed spectral histogram features, which adopts a broader set of filters before collecting the histogram features, including gradient, Laplacian of Gaussian and Gabor filters. Compared with [80], the histogram features in [86] were based on the whole testing window rather than local regions, and Support Vector Machines (SVMs) were used for classification. In [87] another histogram-based feature, called spatial histograms, was proposed. The spatial histograms are based on local statistics of LBP. HoG and LBP were also combined in [88], which achieved excellent performance in human detection with partial occlusion handling. Region covariance is another statistics based feature, proposed in [89] for generic object detection and texture classification tasks. To extract these features the covariance matrices among the color channels and gradient images are computed instead of the histograms. Regional covariance features can also be efficiently computed using integral images.

In [90] a sparse feature set was proposed in order to strengthen the features' discrimination power without incurring too much additional computational cost. Each sparse feature can be represented as:

$$f(x) = \sum_i \alpha_i p_i(x; u, v, s), \alpha_i \in \{-1, +1\} \tag{12}$$

where x is an image patch, and p_i is a granule of the sparse feature. A granule is specified by 3 parameters: horizontal offset u, vertical offset v and scale s. For instance, as shown in Fig. 8, $p_i(x; 5, 3, 2)$ is a granule with top-left corner (5,3), and scale $2^2 = 4$, and $p_i(x; 9, 13, 3)$ is a granule with top-left corner (9,13), and scale $2^3 = 8$. Granules can be computed efficiently using pre-constructed image pyramids, or through the integer image. In [90], the maximum number of granules in a single sparse feature is 8. Since the total number of granules is large, the search space is very large and exhaustive search is infeasible. The method employed a heuristic search scheme, where granules are added to a sparse feature one-by-one, with an expansion operator that removes, refines and adds granules to a partially selected sparse feature. To reduce the computation, the authors further conducted multi-scaled search, which uses

ACCEPTED MANUSCRIPT

a small set of training examples to first evaluate all features and then reject those that are unlikely to be good. The performance of the multi-view face detector trained in [90] using sparse features was very good.

As new features are composed in seeking the best discrimination power, the feature pool becomes larger and larger, thus creating new challenges in the feature selection process. A number of recent works have attempted to address this issue. For instance, [91] proposed to discover compositional features using the classic frequent item-set mining scheme in data mining. Instead of using the raw feature values, they assumed a collection of induced binary features (e.g., decision stumps with known thresholds) that are already available. By partitioning the feature space into sub-regions through these binary features, the training examples can be indexed by the sub-regions they are located. The algorithm then searches for a small subset of compositional features that are both frequent to have statistical significance and accurate to be useful for label prediction. The final classifier is then learned based on the selected subset of compositional features through AdaBoost. In [92], the authors first established an analogue between compositional feature selection and generative image segmentation, and applied the Swendsen-Wang Cut algorithm to generate n-partitions for the individual feature set, where each subset of the partition corresponds to a compositional feature. This algorithm re-runs for every weak classifier selected by the AdaBoost learning framework. On a person detection task tested, the composite features showed significant improvement, especially when the individual features were very weak (e.g., Haar-like features).

In some applications such as object tracking, even if the number of possible features is not extensive, an exhaustive feature selection is still impractical due to computational constraints. In [93], the authors proposed a gradient based feature selection scheme for online boosting with primary applications in person detection and tracking. Their work iteratively updates each feature using a gradient descent algorithm, by minimizing the weighted least square error between the estimated feature response and the true label. This is particularly attractive for tracking and updating schemes such as [82], where at any time instance, the object's appearance is already represented by a boosted classifier learned from previous frames. Assuming there is no dramatic change in the appearance, the gradient descent based algorithm can refine the features in a very efficient manner.

There have also been many features that attempted to model the shape of the objects. For instance, in [94] multiple boundary fragments to weak classifiers were composed and formed a strong "boundary-fragment-model" detector using boosting. They ensured the feasibility of the feature selection process by limiting the number of boundary fragments to 2-3 for each weak classifier. In [95] the object detectors were learned with a boosting algorithm and the feature set consisted of a randomly chosen dictionary of contour fragments. A very similar edgelet feature was proposed in [96], and was used to learn human body part detectors in order to handle multiple, partially occluded humans. In [97], shapelet features focusing on local regions of the image were built from low-level gradient information using AdaBoost for pedestrian detection. An interesting side benefit of having contour/edgelet features is that object detection and object segmentation can be performed jointly, such as the work in [98] and [99].

2.2.1. Robust Descriptors meet Boosting

The field of pedestrian detection has been dominated by the now classic HoG plus SVM approach proposed in [19]. This pivotal paper started an era where robust descriptors such as HoGs, SIFT and their fast counterparts such as SURF features, densely or sparsely measured all over the image have been concatenated and fed to a classifier. These very simple schemes achieve competitive pedestrian and face detection performance [100, 26, 101]. The application of these robust features with cascades of weak classifiers and boosting methodologies has recently started to receive attention.

One of the first such approaches was recently introduced combining a cascade of weak-classifiers with SURF features [102, 103]. In particular, in [103] the detection region was represented by patches and each patch was described by a multi-dimensional SURF descriptor. The number of SURF patches contained only few hundreds of features. Logistic regression was adopted as the weak classifier on each local SURF patch and Area Under ROC curve (AUC) was used as the criterion for convergence. In [103] it was shown that the SURF cascade is able to converge very fast (within even an hour on a standard desktop).

Variations of LBP and HoG features were proposed in [22] and applied for face detection. In particular, the Local Gradient Patterns (LGP) and Binary Histograms of Oriented Gradients (BHOG) were proposed. The LGP feature construction methodology compares the gradient values, which are computed as the absolute value of the intensity difference between the given pixel and its neighbouring pixels, in predefined neighbourhood structures. The LGP feature is then created by assigning 1 if the gradient value of a neighbouring pixel is greater than the threshold value,

ACCEPTED MANUSCRIPT

and 0 otherwise. When compared to LBPs the LGP provides robustness to local gradient variations caused by make-up, glasses and possible background variations. The other descriptor proposed in [22], the so-called BHOG, computes features as follows: The square of the gradient magnitude and the orientation of all pixels within a predefined block is computed, then an orientation histogram is computed in a similar manner as in HoGs, finally by thresholding the histogram bins (i.e., assigning 1 if the histogram bin has a value higher than the threshold otherwise assigning 0) the orientation histogram is encoded into an 8 bit vector. The threshold value was set to be the average value of the total histogram bins. The advantage of BHOG features is that they can be efficiently computed using the integral histogram methods [104] and could also be efficiently combined with boosting methods (such as Adaboost). The paper also proposed to fuse LBP, LGP and BHOG for face detection [22].

Very recently an approach that uses various robust features with boosting achieved state-of-the-art performance in face detection in the wild [26][1]. The method used the so-called Integral Channel Features (ICF) proposed in [25]. The IGF method applies the idea of integral images, which used only with the gray scale channel of pixel intensities in the Viola-Jones algorithm, on top of other features (channels) to efficiently pool statistics with different regions of support. In the original IGF paper [25] the channels used were gradient histograms, color (including grayscale, RGB, HSV and LUV), and gradient magnitude. The adaptation of the IGM method for face detection used HoG and LUV color channels combined with global feature normalization [26]. Furthermore, it learned 22 different face detection templates (11 for each of the two scales, 5 for near frontal faces and 6 for rotated faces). For each template the cascade was trained using the maximum amount of pooling features, i.e. all possible rectangles for the baseline and all possible squares for the largest templates. This detector, coined Headhunter, along with a DPM face detector, described in the next section, are among the current state-of-the-art in face detection in the wild.

Finally, a state-of-the-art approach similar to HeadHunter was independently proposed in [105]. In particular, in [105] it was proposed to use sub-sampled channel features, such gray-scale, RGB, HSV and LUV, gradient magnitude, and gradient histograms, to learn a cascade of weak classifiers. Features are extracted directly from the down-sampled channels without computing rectangular sums (i.e., using integral images). This could lead to a faster feature computation and smaller feature pool size for boosting learning. For weak classifier a depth-2 decision tree was used and to build the cascade the soft-cascade strategy [106] was adopted.

We summarize the features presented in this Section in Table 1.

2.3. Variations of the Boosting Learning Algorithm

In addition to exploring better features, another venue to improve the detector's performance is through improving the boosting learning algorithm, particularly under the cascade decision structure. In the original face detection paper by Viola and Jones [17], the standard AdaBoost algorithm [56] was adopted. In a number of follow-up works [59, 60, 61, 62], researchers advocated the use of RealBoost, which was explained in detail in Section 2.1.2. In [107] and [66] comparisons between three boosting algorithms: AdaBoost, RealBoost and GentleBoost were performed. The papers reached different conclusions. In particular, in [107] it is recommended to use GentleBoost while the [66] showed that RealBoost works slightly better when combined with CART-based weak classifiers. In the following, we describe a number of recent works on boosting learning for face/object detection, with emphasis on adapting to the cascade structure, the training speed, multi-view face detection, etc.

In [59], the FloatBoost algorithm was proposed, which attempted to overcome the monotonicity problem of the sequential AdaBoost Learning. Specifically, AdaBoost is a sequential forward search procedure using a greedy selection strategy, which may be suboptimal. FloatBoost incorporates the idea of floating search [108] into AdaBoost, which not only adds features during training, but also backtracks and examines the already selected features to remove those that are least significant. The authors claimed that FloatBoost usually needs fewer weak classifiers than AdaBoost to achieve a given objective. In [109] the use of evolutionary algorithms was proposed to minimize the number of classifiers without degrading the detection accuracy. They showed that such an algorithm can reduce the total number of weak classifiers by over 40%. Note that in practice only the first few nodes are critical to the detection speed, since most testing windows are rejected by the first few weak classifiers in a cascade architecture.

As mentioned in Section 2.1.3, Viola and Jones [17] trained each node independently. A number of follow-up works showed that there is indeed information in the results from the previous nodes, and it is best to reuse them

[1] The method was inspired by the previous work by the same authors for pedestrians detection [101]

ACCEPTED MANUSCRIPT

Table 1. Features for face/object detection.

Feature Type	**Representative Works**
Haar-like features and its variations	Haar-like features [17]
	Rotated Haar-like features [63]
	Rectangular features with structure [59, 64]
	Haar-like features on motion filtered image [65]
Pixel-based features	Pixel pairs [78]
	Control point set [79]
Binarized features	Modified census transform [69]
	LBP features [71, 72]
	Locally assembled binary feature [73]
Generic linear features	Anisotropic Gaussian filters [75]
	LNMF [76]
	Generic linear features with KL boosting [74]
	RNDA [77]
Statistics-based features	Edge orientation histograms [80, 19] etc.
	Spectral histogram [86]
	Spatial histogram (LBP-based) [87]
	HoG and LBP [88]
	Region covariance [89]
	SURF [102, 103]
Composite features	Joint Haar-like features [62]
	Sparse feature set [90]
	LGB, BHOG [22]
	Integral Channel Features on HoG and LUV (Headhunter) [26]
	HoG, HSV, RGB, LUV, Grayscale, Gradient Magnitude [105]
Shape features	Boundary/contour fragments [94, 95]
	Edgelet [96]
	Shapelet [97]

ACCEPTED MANUSCRIPT

instead of starting from scratch at each new node. For instance, in [110], the use of a "chain" structure was proposed to integrate historical knowledge into successive boosting learning. At each node, the existing partial classifier is used as a prefix classifier for further training. Boosting chain learning can thus be regarded as a variant of AdaBoost learning with similar generalization performance and error bound. In [61], the so-called nesting-structured cascade was proposed. Instead of taking the existing partial classifier as a prefix, the method took the confidence output of the partial classifier and used it as a feature to build the first weak classifier. Both papers demonstrated better detection performance than the original Viola-Jones face detector.

One critical challenge in training a cascade face detector is how to set the thresholds for the intermediate nodes. This issue has inspired a lot of works in the literature. First, in [111] it was observed that the goal of the early stages of the cascade is mostly to retain a very high detection rate, while accepting modest false positive rates if necessary. The method proposed a new scheme called asymmetric AdaBoost, which artificially increases the weights on positive examples in each round of AdaBoost such that the error criterion biases towards having low false negative rates. In [112], the above work was extended by proposing to balance the skewness of labels presented to each weak classifiers, so that they are trained more equally. In [113] a more rigorous form of asymmetric boosting based on the statistical interpretation of boosting [55] with an extension of the boosting loss, was proposed. Namely, the exponential cost criterion in (3) is rewritten as:

$$L^T = \sum_{i=1}^{N} \exp\{-c_i z_i F^T(x_i)\}, \tag{13}$$

where $c_i = C_1$ for positive examples and $c_i = C_0$ for negative examples. The method in [113] minimized the above criterion following the AnyBoost framework in [114]. The method wes able to build a detector with very high detection rate [115], though the performance of the detector deteriorates very quickly when the required false positive rate is low.

In [116] it was proposed to decouple the problems of feature selection and ensemble classifier design in order to introduce asymmetry. They first applied the forward feature selection algorithm to select a set of features, and then formed the ensemble classifier by voting among the selected features through a linear asymmetric classifier (LAC). The LAC is supposed to be the optimal linear classifier for the node learning goal under the assumption that the linear projection of the features for positive examples follows a Gaussian distribution, and that for negative examples is symmetric. Mathematically, LAC has a similar form as the well-known Fisher discriminant analysis (FDA) [117], except that only the covariance matrix of the positive feature projections are considered in LAC. In practice, their performances are also similar. Applying LAC or FDA on a set of features pre-selected by AdaBoost is equivalent to readjusting the confidence values of the AdaBoost learning (7). Since at each node of the cascade, the AdaBoost learning usually has not converged before moving to the next node, readjusting these confidence values could provide better performance for that node. However, when the full cascade classifier is considered, the performance improvement over AdaBoost is diminished. In [116] the phenomenon was attributed to the booststrapping step and the post processing step, which also have significant effects on the cascade's performance.

With or without asymmetric boosting/learning, at the end of each cascade node, a threshold still has to be set in order to allow the early rejection of negative examples. These node thresholds reflect a trade-off between detection quality and speed. If they are set too aggressively, the final detector will be fast, but the detection rate may drop. On the other hand, if the thresholds are set conservatively, many negative examples will pass the early nodes, making the detector slow. In early works, the rejection thresholds were often set in very ad hoc manners. For instance, the learning procedure in [17] attempted to reject zero positive examples until this became impossible and then reluctantly gave up on one positive example at a time. Huge amount of manual tuning is thus required to find a classifier with good balance between quality and speed, which is very inefficient. The method proposed in [107] instead built the cascade targeting each node to have 0.1% false negative rate and 50% rejection rate for the negative examples. Such a scheme is simple to implement, though no speed guarantee can be made about the final detector.

In [118], the use of a ratio test to determine the rejection thresholds, was proposed. Specifically, the authors viewed the cascade detector as a sequential decision-making problem. A sequential decision-making theory had been developed in [119], which proved that the solution to minimizing the expected evaluation time for a sequential decision-making problem is the sequential probability ratio test. In [118] the notion of nodes was abandoned, and set the rejection threshold after each weak classifier. The method then approximated the joint likelihood ratio of all the weak classifiers between negative and positive examples with the likelihood ratio of the partial scores, in which

ACCEPTED MANUSCRIPT

case the algorithm simplified to be rejecting a test example if the likelihood ratio at its partial score value is greater than $\frac{1}{\alpha}$, where α is the false negative rate of the entire cascade. In [66] another fully automatic algorithm for setting the intermediate thresholds during training, was proposed. Given the target detection and false positive rates, their algorithm used the empirical results on validation data to estimate the probability that the cascade will meet the goal criteria. Since a reasonable goal make not be known a priori, the algorithm adjusts its cost function depending on the attainability of the goal based on cost prediction. In [68], a dynamic cascade was proposed, which assumes that the false negative rate of the nodes changes exponentially in each stage, following the idea in [106]. The approach is simple and ad hoc, though it appears to work reasonably well.

Setting intermediate thresholds during training is a specific scheme to handle huge amount of negative examples during boosting training. Such a step is unnecessary in AdaBoost, at least according to its theoretical derivation. Recent development of boosting based face detector training have shifted toward approaches where these intermediate thresholds are not set during training, but rather done until the whole classifier has been learnt. For instance, in [120] it was assumed that a cascade of classifiers is already designed, and proposed an optimization algorithm to adjust the intermediate thresholds. It represents each individual node with a uniform abstraction model with parameters (e.g., the rejection threshold) controlling the tradeoff between detection rate and false alarm rate. It then uses a greedy search strategy to adjust the parameters such that the slope of the logarithm scale ROC curves of all the nodes are equal. One issue in such a scheme is that the ROC curves of the nodes are dependent to changes in thresholds of any earlier nodes, hence the greedy search scheme can at best be an approximation. In [106] a heuristic approach to use a parameterized exponential curve to set the intermediate nodes' detection targets, called a "rejection distribution vector", was proposed instead. By adjusting the parameters of the exponential curve, different tradeoffs can be made between speed and quality. Perhaps a particular family of curves is more palatable, but it is still arbitrary and non-optimal. In [121] a more principled data-driven scheme for setting intermediate thresholds named multiple instance pruning, was proposed. They explored the fact that nearby a ground truth face there are many rectangles that can be considered as good detection. Therefore, only one of them needs to be retained while setting the intermediate thresholds. Multiple instance pruning does not have the flexibility as [106] to be very aggressive in pruning, but it can guarantee identical detection rate as the raw classifier on the training data set.

The remaining issue is how to train a cascade detector with billions of examples without explicitly setting the intermediate thresholds. In [106], a scheme was proposed that starts with a small set of training examples, and adds to it new samples at each stage that the current classifier misclassifies. The number of new non-faces to be added at each training cycle affects the focus of AdaBoost during training. If the number is too large, AdaBoost may not be able to catch up and the false positive rate will be high. If the number is too small, the cascade may contain too many weak classifiers in order to reach a reasonable false positive rate. In addition, later stages of the training will be slow due to the increasing number of negative examples, since none of them will be removed during the process. In [68] and [121], the use of importance sampling in order to address the large data set issue, was proposed. The training positive or negative data set were periodically re-sampled to ensure feasible computation. Both works reported excellent results with such a scheme.

Training a face detector is a very time-consuming task. In early works, due to the limited computing resources, it could easily take months and lots of manual tuning to train a high quality face detector. The main bottleneck is at the feature selection stage, where hundreds of thousands of Haar features will need to be tested at each iteration. A number of papers has been published to speed up the feature selection process. For instance, in [122] a discrete downhill search scheme to limit the number of features compared during feature selection, was proposed. Such a greedy search strategy offered a 300–400 fold speed up in training time, though the false positive rate of the resultant detector increased by almost a factor of 2. In [66] various filter schemes to reduce the size of the feature pool were studied. It was shown that randomly selecting a subset of features at each iteration for feature selection appears to work reasonably well. In [123] a cascade learning algorithm based on forward feature selection [124] was proposed. The learning procedure was shown to be two orders of magnitude faster than the traditional approaches. The idea is to first train a set of weak classifiers that satisfy the maximum false positive rate requirement of the entire detector. During feature selection, these weak classifiers are added one by one, each making the largest improvement to the ensemble performance. Weighting of the weak classifiers can be conducted after the feature selection step. In [125] another fast method to train and select Haar features was presented. The method treated the training examples as high dimensional random vectors, and kept the first and second order statistics to build classifiers from features. The time complexity of the method is linear to the total number of examples and the total number of Haar features. Both [123]

ACCEPTED MANUSCRIPT

and [125] reported experimental results demonstrating better ROC curve performance than the traditional AdaBoost approach, though it appears unlikely that they can also outperform the state-of-the-art detectors such as [61, 106].

Various efforts have also been made to improve the detector's test speed. For instance, in the sparse feature set in [90], the method limited the granules to be in square shape, which is very efficient to compute in both software and hardware through building pyramids for the test image. For HoG and similar gradient histogram based features, the integral histogram approach [126] was often adopted for faster detection. Schneiderman [127] designed a feature-centric cascade to speed up the detection. The idea is to pre-compute a set of feature values over a regular grid in the image, so that all the test windows can use their corresponding feature values for the first stage of the detection cascade. Since many feature values are shared by multiple windows, significant gains in speed can be achieved. A similar approach was deployed in [73] to speed up their locally assembled binary feature based detector. In [128], the authors proposed a scheme to improve the detection speed on quasi-repetitive inputs, such as the video input during video-conferencing. The idea is to cache a set of image exemplars, each inducing its own discriminant subspace. Given a new video frame, the algorithm quickly searches through the exemplar database indexed with an on-line version of tree-structured vector quantization, S-tree [129]. If a similar exemplar is found, the face detector will be skipped and the previously detected object states will be reused. This results in about 5-fold improvement in detection speed. Similar amount of speed-up can also be achieved through selective attention, such as those based on motion, skin color, background modelling and subtraction, etc.

As shown in Fig. 1, in real-world images, faces have significant variations in orientation, pose, facial expression, lighting conditions, etc. A single cascade with Haar features has proven to work very well with frontal or near-frontal face detection tasks. However, extending the algorithm to multi-pose/multi-view face detection is not straightforward. If faces with all pose/orientation variations are trained in a single classifier, the results are usually sub-optimal. To this end, researchers have proposed numerous schemes to combat the issue, most of them following the "divide and conquer" strategy.

Fig. 9 showed a number of detector structures for multiview face detection. Among these structures, the most straightforward one is Fig. 9(a), the parallel cascade in [61]. An individual classifier is learned for each view. Given a test window, it is passed to all the classifiers. After a few nodes, one cascade with the highest score will finish the classification and make the decision. This simple structure could achieve rather good performance, though its running speed is generally slow, and the correlation between faces of different views could have been better exploited. In [59] a pyramid structure to handle the task, as shown in Fig. 9 (b), was used. The detector pyramid consists of 3 levels. The first level of the pyramid works on faces at all poses; the second level detects faces between -90° and -30° (left profile), between -30° and 30° (frontal), and between 30° and 90° (right profile), respectively; the third level detects faces at 7 finer angles. Once a test window passes one level of the detector, it will be passed to all the children nodes for further decision. This design is more efficient than the parallel cascade structure, but still has room for improvement.

Fig. 9(c) and (d) showed two decision tree structures for multiview face detection. In [64], the authors proposed to first use a pose estimator to predict the face pose of a test window. Given the predicted pose, a cascade for that pose will be invoked to make the final decision. A decision tree was adopted for pose estimation, which resulted in the detector structure in Fig. 9(c). With this structure, a test window will only run through a single cascade once its pose has been estimated, thus the detector is very efficient. In [130] a similar tree structure (Fig. 9(d)) was used for frontal face detection at different orientations, except that their early nodes were able to perform rejection to further improve speed. However, pose/orientation estimation is a non-trivial task, and can have many errors. If a profile face is misclassified as frontal, it may never be detected by the frontal face cascade. The works [67] and [131] independently proposed very similar solutions to this issue, which were named vector boosting and multiclass Bhattacharyya boost (MBHBoost), respectively. The idea is to have a vector valued output for each weak classifier, which allows an example to be passed into multiple subcategory classifiers during testing (Fig. 9(d)), and the final results to be fused from the vector output. Such a soft branching scheme can greatly reduce the risk of misclassification during testing. Another interesting idea in [67, 131] was to force all the subcategory classifiers to share the same features. Namely, at each iteration, only one feature is chosen to construct a weak classifier with vector output, effectively sharing the feature among all the subcategories. Sharing features among multiple classifiers had been shown as a successful idea to reduce the computational and sample complexity when multiple classifiers are jointly trained [132].

Vector boosting and MBHBoost solved the issue of misclassification in pose estimation during testing. During training, they still used faces manually labelled with pose information to learn the multiview detector. However, for

ACCEPTED MANUSCRIPT

certain object classes such as pedestrians or cars, an agreeable manual pose labelling scheme is often unavailable. In [133] the implicit shape model in [134] was extended to explicitly handle and estimate viewpoints and articulations of an object category. The training examples were first clustered, with each cluster representing one articulation and viewpoint. Separate models were then trained for each cluster for classification. In [135] an exemplar-based categorization scheme for multi-view object detection was proposed. At each round of boosting learning, the algorithm not only selects a feature to construct a weak classifier, but also selects a set of exemplars to guide the learning to focus on different views of the object. In [136] a probabilistic boosting tree, which embedded clustering in the learning phase was proposed. At each tree node, a strong AdaBoost based classifier was built. The output of the AdaBoost classifier was used to compute the posterior probabilities of the examples, which were used to split the data into two clusters. In some sense, the traditional boosting cascade can be viewed as a special case of the boosting tree, where all the positive examples are pushed into one of the child nodes. The performance of boosting tree on multi-view object detection is uncertain due to the limited experimental results provided in the paper. In [137], a similar boosted tree algorithm was proposed. Instead of performing clustering before boosting learning or using posterior probabilities, they showed that by using the previously selected features for clustering, the learning algorithm converges faster and achieves better results.

Some recent works went one step further and did not maintain a fixed subcategory label for the training examples. For instance, in [138] an algorithm called multiple classifier boosting was proposed. The algorithm is a straightforward extension of the multiple instance boosting approach in [139]. In this approach, the training examples no longer have a fixed subcategory label. A set of likelihood values were maintained for each example, which describe the probability of it belonging to the subcategories during training. These likelihood values are combined to compute the probability of the example being a positive example. The learning algorithm then maximizes the overall probability of all examples in the training data set. In [140] a very similar scheme, so-called multi-pose learning, was independently developed. The method was further combined with multiple instance learning in a unified framework. One limitation of the above approaches is that the formulation requires a line search at each weak classifier to find the optimal weights, which makes it slow to train and hard to deploy feature sharing [132]. In [141] an algorithm called winner-take-all multiple category boosting (WTA-McBoost) was proposed. The algorithm is more suitable for learning multiview detectors with huge amount of training data. Instead of using AnyBoost [114], WTA-McBoost is derived from confidence rated AdaBoost [58], which is much more efficient to train, and easy to support feature sharing.

Table 2 summarizes this Section listing all the challenges in boosting learning and the approaches to address them.

2.4. Rigid-Template Face Detection using Neural Networks

Neural networks (NN) have always been a popular approach for face detection in the literature. Early representative methods included the detectors in [142, 143]. In [144] an approach based on a neural network model, the so-called constrained generative model (CGM), was proposed. CGM is an auto-associative, fully connected multilayer perceptron (MLP) with three large layers of weights, trained to perform nonlinear dimensionality reduction in order to build a generative model for faces. Multi-view face detection was achieved by measuring the reconstruction errors of multiple CGMs, combined via a conditional mixture and an MLP gate network. In [145], the authors proposed a face detection scheme based on a convolutional neural architecture. Compared with traditional feature-based approaches, convolutional neural network derives problem-specific feature extractors from the training examples automatically, without making any assumptions about the features to extract or the areas of the face patterns to analyse.

In [146] another convolutional neural network (CNN) based approach, which is pictorially described in Fig. 10 and similar to LeNet5 [40], was proposed. The input is image patches at 32×32 pixels resolution, and the network contains four convolution layers and one fully connected layer. The network performs two tasks: (a) rejecting the non-face hypotheses and (b) estimating the facial pose of the correct face hypothesis (which is performed by regressing to a set of facial pose parameters such as yaw, pitch angles etc.). Assuming that $\mathbf{w}$ is the set of parameters of the DCCN architecture, the cost function to be minimized in order to estimate $\mathbf{w}$ is

$$E(\mathbf{w}) = \frac{1}{N_1} \sum_{i \in N_1} \|G(\mathbf{q}_i, \mathbf{w}) - f(\mathbf{z})\|^2 + \frac{1}{N_2} \sum_{j \in N_1} k \exp(-\|G(\mathbf{q}_j, \mathbf{w}) - f(\tilde{\mathbf{z}}_j)\|) \tag{14}$$

where $G(\mathbf{q}_i, \mathbf{w})$ is the network output given input $\mathbf{q}_i$, N_1 and N_2 are the numbers of positive and negative examples for training, $f(\mathbf{z})$ is a face manifold parameterized by the facial pose, $\tilde{\mathbf{z}}_j = \arg\min_{\mathbf{z}_j} \|G(\mathbf{q}_j, \mathbf{w}) - f(\tilde{\mathbf{z}}_j)\|$ is the closest

ACCEPTED MANUSCRIPT

Table 2. Face/object detection schemes to address challenges in boosting learning.

Challenges	**Representative Works**
General boosting schemes	AdaBoost [17]
	RealBoost [59, 60, 61, 62]
	GentleBoost [107, 66]
	FloatBoost [59]
Reuse previous nodes' results	Boosting chain [110]
	Nested cascade [61]
Introduce asymmetry	Asymmetric Boosting [111, 112, 113]
	Linear asymmetric classifier [116]
Set intermediate thresholds during training	Fixed node performance [107]
	WaldBoost [118]
	Based on validation data [66]
	Exponential curve [68]
Set intermediate thresholds after training	Greedy search [120]
	Soft cascade [106]
	Multiple instance pruning [121]
Speed up training	Greedy search in feature space [122]
	Random feature subset [66]
	Forward feature selection [123]
	Use feature statistics [125]
Speed up testing	Reduce number of weak classifiers [59, 109]
	Feature centric evaluation [127, 73]
	Caching/selective attention [128] etc.
Multiview face detection	Parallel cascade [61]
	Pyramid structure [59]
	Decision tree [64, 130]
	Vector valued boosting [67, 131]
Learn without subcategory labels	Cluster and then train [133]
	Exemplar-based learning [135]
	Probabilistic boosting tree [136]
	Cluster with selected features [137]
	Multiple classifier/category boosting [138, 140, 141]

ACCEPTED MANUSCRIPT

point on the manifold to the negative example, and k is a positive constant. A stochastic version of the Levenberg-Marquardt algorithm with diagonal approximation of the Hessian was used to find the optimal set of parameters [40]. For detection the proposed CNN is applied to all 32×32 sub-windows of the image, stepped every 4 pixels horizontally and vertically. Also a multiscale approach is used for finding faces in many scales (using factors of $\sqrt{2}$).

In [147], it was shown that a simpler network can also achieve good performance in face detection. The method applies a neural network on features obtained from Haar transformations followed by two layers that sequentially apply image row summation, convolution with 1-D filters and 2×2 sub-sampling. Finally, a fully connected step makes a decision whether the input pattern is a face or not. The network has an exceptional small number of parameters (total of 457 parameters in contrast to the 63,493 parameters in [146]). These parameters are learned through a gradient-based learning algorithm. In order to expedite the procedure a small set of cascade rules plays the rule of a coarse face detector, while the output of the NN is the fine face detector.

Recently, deep convolutional neural networks (DCNN) showed remarkable performance for object categorization [34] and then a similar network was applied for multi-object detection [41]. Compared with previous neural networks that are often considered shallow, DCNN features many convolution/fully-connected layers that were once considered impossible to train. The performance break-through in object categorization and detection can be attributed to many reasons: (a) the availability of huge amount of labeled data, (b) the use of GPUs for fast parallel computation, and (c) the new tricks and regularization techniques such as rectified linear unit (ReLU) and dropout [34], which appear to be the key to avoid overfitting.

In [42], the authors proposed to train a multi-task DCNN for multiview face detection. It adopts the key techniques such as ReLU and dropout from [34], and achieved state-of-the-art results on the publicly available FDDB data set [148]. Inspired by the recent work in DPMs, they trained the DCNN with multiple tasks including face pose estimation and landmark localization. The idea is to allow the DCNN to obtain better lower level features by adding the other tasks. Slight improvement in face detection performance is achieved, although the experiments were not conclusive.

DCNN and boosting based approaches using HoG/SIFT type features (such as the state-of-the-art method in [26]) have common characteristics. In particular, it has been empirically verified that the first layer of DCNN resembles SIFT type features [149]. Currently, there is little theoretical or strong empirical evidence what type of features all the remaining layers of DCNNs are learning, except some preliminary scheme for feature visualization [41]. Intuitively, following the rational of invariant scattering networks [149], we can postulate that all the layers contribute to the development of features invariant to face deformations. Similarly, ICF on HoG features contribute to the development of another layer of invariant features, and the boosting algorithms choose the best features for face detection. In comparison, DPM-based methods have orthogonal strategies when compared to DCNN and boosting. Instead of finding deformation invariants, they try to explicitly model and learn the deformations. As has been empirically shown in [26], DPM-based methods often require less data to train from.

2.5. Other learning schemes for Rigid-Face Detection

Except the seminal work by Viola and Jones [17], there are also a few other research attempts that approach the problem of face detection using a series of rigid-templates providing very competitive performance. Again, we will only focus on works not covered in [15].

In [150] the so-called Antifaces methods proposed. Antifaces is a multi-template scheme for detecting arbitrary objects including faces in images. The core idea is very similar to the cascade structure in [17], which uses a set of sequential classifiers to detect faces and rejects non-faces fast. Each classifier, referred as a "detector" in [150], is a template image obtained through constrained optimization, where the inner product of the template with the example images are minimized, and the later templates are independent to the previous ones. Interestingly, in this approach, negative images were modelled by a Boltzmann distribution and assumed to be smooth, thus none is needed during template construction.

In [151] a Bayesian discriminating features method for frontal face detection was proposed. The face class was modelled as a multivariate normal distribution. A subset of the non-faces that lie closest to the face class was then selected based on the face class model and also modelled with a multivariate normal distribution. The final face/non-face decision was made by a Bayesian classifier. Since only the non-faces closest to the face class were modelled, the majority of the non-faces were ignored during the classification. This was inspired by the concept of support vector

ACCEPTED MANUSCRIPT

machines (SVMs) [152], where only a subset of the training examples (the support vectors) are used to define the classification boundary.

SVMs are known as maximum margin classifiers, as they simultaneously minimize the empirical classification error and maximize the geometric margin. Due to their superior performance in general machine learning problems, they have also become a very popular approach for face detection [153, 154]. However, the detection speed of SVM based face detectors was generally slow. Thus, various schemes have been proposed to speed up the process. For instance, in [155] a method that computes a reduced set of vectors from the original support vectors was proposed. These reduced set vectors are then tested against the test example sequentially, making early rejections possible. In [156], detection speed was further improved by approximating the reduced set vectors with rectangle groups, gaining another 6-fold speedup. A hierarchy of SVM classifiers with different resolutions in order to speed up the overall system was applied [157]. The early classifiers were at low resolution, say, 3×3 and 5×5 pixels, which can be computed very efficiently to prune negative examples.

Multiview face detection has also been explored with SVM based classifiers. In [158] a multiview face detector similar to the approach in [159, 64] was proposed. In this work a face pose estimator using support vector regression (SVR) is first constructed and subsequently separate SVM face detectors one for each face pose are trained. In contrast, the method in [160] applies multiple SVMs first, and then an SVR to fuse the results and generate the face pose. This method is slower, but it has lower risk of assigning a face to the wrong pose SVM and cause misclassification. In [161] it was argued that in real world settings the face poses may vary greatly thus many different SVMs are required. In the same work an approach to combine cascade and bagging for multiview face detection was proposed. Namely, a cascade of SVMs were first trained through bootstrapping. The remaining positive and negative examples were then randomly partitioned to train a set of SVMs, whose outputs were then combined through majority voting. In [162] a single SVM for multiview face detection was used, and relied on the combination of local and global kernels for better performance. No experimental results were given in [161, 162] to compare the proposed methods with existing schemes on standard data sets, hence it is unclear whether these latest SVM based face detectors can outperform those learned through boosting.

Recently, interesting methodologies have been proposed inspired from principles applied in image retrieval [44, 43]. In particular, in the first such work an exemplar-based approach was applied for face detection. In a straightforward exemplar-based approach a test sample is directly matched against a collection of facial images. A face can be then detected as long as enough similar exemplars are included in the exemplar set. Hence, the exemplar set should be large enough to cover the large appearance variations in faces. As a result, direct matching against a large data collection is highly inefficient, especially if combined with sliding window approaches. The method proposed in [44] capitalizes on bag-of-word image retrieval methods to extract features from each exemplar. That is, the bag-of-words representation produces a voting map over the test image (similar to the generalized Hough transform used in [45]). The voting map of each exemplar can be considered as a weak classifier. Face detection in an image region is performed by combining the voting maps of the exemplars. The methodology was theoretically justified by showing that if each exemplar is considered independent, then the voting scheme is operating as a Naive Bayes classifier. This methodology was further developed in [43] where a boosting-based strategy was proposed for selecting the exemplars to use as weak classifiers.

Table 3 summarizes the approaches presented in this section.

3. Deformable Parts-Model for Face Detection

DPMs or pictorial structures modelling is one of the de-facto choices for developing generic object detectors. DPM methods have been recently applied for face detection achieving state-of-the-art results. In this Section, we start by describing the simple generative pictorial structure model developed in [47]. Then, we show how it can be reformulated using a discriminative framework capitalizing on weak (i.e., facial bounding boxes) [35] or strong annotations consisting of locations of facial landmarks [48]. Subsequently, we describe combinations of facial DPMs with human body models to exploit the co-occurrence of face and body [163, 164].

One of the drawbacks of DPM models is that they have relative high computational complexity. We will describe several approaches for reducing the complexity without significant performance degradation [165, 166, 167, 168, 169]. Finally, we discuss about a DPM model that produces state-of-the-art results in recent benchmarks.

ACCEPTED MANUSCRIPT

Table 3. Other schemes for face/object detection (since [15]).

General Approach	**Representative Works**
Template matching	Antiface [150]
Bayesian	Bayesian discriminating features [74]
SVM – speed up	Reduced set vectors and approximation [155, 156]
	Resolution based SVM cascade [157]
SVM – multi-view face detection	SVR based pose estimator [158]
	SVR fusion of multiple SVMs [160]
	Cascade and bagging [161]
	Local and global kernels [162]
Retrieval-based Methods	Exemplar-based method [44]
	Boosted Exemplar-based method [43]

3.1. Facial Pictorial structures

Pictorial Structures (PS) have been first introduced in [14] and became quite popular after the influential study in [47]. In this study it was shown that, for tree-based pictorial structures, very efficient dynamic programming algorithms can be applied for finding the global optimum of the cost function using Generalized Distance Transforms (GDT).

The pictorial structure architecture is a general framework for describing the appearance of parts as well as the connections between them. The framework is general for many object-types, but in this work we confine ourselves to faces. One way to describe the pictorial structure is via an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a set of vertices that correspond to n-facial parts, and $\mathcal{E}$ is a set of edges (v_i, v_j) with $(v_i, v_j) \in \mathcal{E}$ if and only if there is a connection between facial part v_i and v_j. An instance of a face is given by a configuration $L = (\mathbf{l}_1, \ldots, \mathbf{l}_n)$ of parts, where $\mathbf{l}_i$ corresponds to the location of part v_i. Some examples of facial graphs used for facial pictorial structures can be found in Fig. 11 (a) and (b). The red points correspond to vertices (i.e., parts) and the blue lines to edges (i.e., part connections). Note that in this case parts correspond to semantically meaningful facial landmarks (i.e., mouth, nose etc.). However, such correspondence is not necessary, since the parts can be automatically learned through training examples.

Let $\theta = (\theta_v, \mathcal{E}, \theta_e)$ be the general set of parameters of a pictorial structure with $\theta_v = \{\theta_{v_1}, \ldots, \theta_{v_n}\}$ being the set of parameters that correspond to the appearance of the parts, and $\theta_e = \{\theta_e^{ij} | (v_i, v_j) \in \mathcal{E}\}$ being the parameters that correspond to the part connections. Furthermore, we define local generative models $p(I|\mathbf{l}_i, \theta_{v_i})$ and $p(\mathbf{l}_i, \mathbf{l}_j|\theta_e^{ij})$, which describe the appearance of parts and deformation, respectively, where I is the observed image. In [47], the local appearance term $p(I|\mathbf{l}_i, \theta_{v_i})$ was modeled as a normal distribution $\mathcal{N}(\mathbf{f}(\mathbf{l}_i)|\mathbf{m}_i, \mathbf{\Sigma}_i)$, where $\mathbf{f}(\mathbf{l}_i)$ are the extracted features at location $\mathbf{l}_i$ and $\theta_{v_i} = (\mathbf{m}_i, \mathbf{\Sigma}_i)$ is the set of local part parameters comprising of the mean and the covariance matrix for the features of part v_i. Similarly, the distribution of the relative location of part v_i with respect to the location of part v_j was selected as $p(\mathbf{l}_i, \mathbf{l}_j|\theta_e^{ij}) = \mathcal{N}(\mathbf{l}_i - \mathbf{l}_j|\mathbf{s}_{ij}, \mathbf{\Sigma}_{ij})$. In order to formulate the problem using GDT, part locations were linearly transformed, such that the covariance $\mathbf{\Sigma}_{ij}$ is diagonal.

Given an image and the set of parameters θ, we measure the probability of a certain configuration of parts L, $p(L|I, \theta)$ as:

$$p(L|I, \theta) \propto \prod_{i=1}^{n} p(I|\mathbf{l}_i, \theta_{v_i}) \prod_{(v_i, v_j) \in \mathcal{E}} p(\mathbf{l}_i, \mathbf{l}_j|\theta_e^{ij}) \tag{15}$$

or equivalently, by taking the negative logarithm, we get the form

$$C(L|I, \theta) = \sum_{i=1}^{n} m_i(\mathbf{l}_i) + \sum_{(v_i, v_j) \in \mathcal{E}} d_{ij}(\mathbf{l}_i, \mathbf{l}_j) \tag{16}$$

ACCEPTED MANUSCRIPT

where $m_i(\mathbf{l}_i) = -\log p(I|\mathbf{l}_i, \theta_{v_i})$ and $d_{ij}(\mathbf{l}_i, \mathbf{l}_j) = -\log p(\mathbf{l}_i, \mathbf{l}_j|\theta_e^{ij})$ are both quadratic functions, since the local distributions are Gaussians. Minimizing $C(L|I, \theta)$ for a number of training instances with regards to parameters θ gives as a maximum likelihood (ML) solution for parameter θ. Similarly, the ML solution for finding the optimum tree-based structure $\mathcal{E}$ is to perform in a similar way, leading to the so-called Chow-Liu algorithm [170].

The problem of face detection is to find the best match(es) of a pictorial facial model L to an image I by minimizing $C(L|I, \theta)$ as

$$L = \arg\min_{L} \sum_{i=1}^{n} m_i(\mathbf{l}_i) + \sum_{(v_i,v_j)\in\mathcal{E}} d_{ij}(\mathbf{l}_i, \mathbf{l}_j). \tag{17}$$

The above optimization problem has a functional form that is quite general and appears in various computer vision problems (e.g., estimation in Markov Random Fields in image restoration and stereo, optimization of active contour models etc.). It was shown in [47] that for tree-based graph structures and for $d_{ij}(\mathbf{l}_i, \mathbf{l}_j)$ that have a Mahalonobis distance form, the optimization problem (17) can be solved using the GDT in linear time with regards to the number of images n and image resolution. For more details regarding this algorithm, the interested reader may refer to [171, 47].

The simple pictorial structure described above has numerous drawbacks: (1) it does not make use of the object-class annotations (i.e., face vs non-face class annotations) in a discriminative manner, (2) it uses a single model for each object, while faces could have drastic changes between views, and (3) it uses only one scale. In order to remedy these drawbacks and train a part-based model using only weak object-class annotations (i.e., a bounding box per object), a multi-scale, discriminatively trained mixture of DPMs was proposed in [35] (publicly available code can be found in [37]).

The pictorial structure defined in [35] used a coarse root representing the whole face/object, and a number of smaller parts of the object at higher resolution. The parts are connected to the root using a star tree model. In a face detection task, the root can capture coarse resolution characteristics such as face boundaries, while parts capture facial details such as eyes, nose and mouth. Fig. 12 describes the above procedure pictorially. It is important to note that in a weakly-supervised setting it is not necessary the parts to correspond to semantically meaningful facial landmarks.

Let us assume that we have a mixture of N different tree models. In order to facilitate a discriminative training procedure such as SVM, the cost function (16) for the m-th component of the mixture is reformulated as

$$\begin{aligned} C(L|I, m) &= A(L|I, m) + S(L|I, m) + a^m \\ A(L|I, m) &= \textstyle\sum_{i=1}^{n_m} \mathbf{w}_i^{mT} \mathbf{f}(\mathbf{l}_i|I) \\ S(L|I, m) &= \textstyle\sum_{ij\in\mathcal{E}_m} a_{ij}^m dx^2 + b_{ij}^m dx + c_{ij}^m dy^2 + d_{ij}^m dy \end{aligned} \tag{18}$$

where

- α^m is a scalar bias (which can be also considered as a prior) associated with the component m of the mixture and $A(L|I, m)$ is the appearance cost for placing template $\mathbf{w}_i^m$ (also called filter) of part v_i at location $\mathbf{l}_i$ and $\mathbf{f}(\mathbf{l}_i|I)$ is the image feature vector (usually a HoG descriptor) extracted from pixel location that corresponds to pixel location $\mathbf{l}_i$ of image I.
- $S(L|I, m)$ is the cost of the spatial arrangement of parts L where $dx = x_i - x_j$ and $dy = y_i - y_j$ are the displacements of the i-th part relative to the j-th. The interpretation of the deformation model depends on the setting. That is, in a strongly supervised setting where the facial parts correspond to semantically meaningful landmarks, such as eyebrows, eyes, lips nose, etc. [48], the deformation cost models the deformations in the mouth, eyes etc. A DPM that corresponds to a frontal face component can be found in Fig. 11 (c).

  In a weakly supervised setting, where the facial parts have been automatically selected as in [35], there may be no explicit interpretations of the deformation cost. Nevertheless, both models have been successfully used for face detection and in both cases the shape can be interpreted as a spring model where $(a_{ij}^m, b_{ij}^m, c_{ij}^m, d_{ij}^m)$ are considered the parameters associated to the rest location and spring rigidity.

  In order to have a better understanding of the deformation model we will study the fully supervised setting. In this case it is convenient to write the shape cost function as:

$$S(L|I, m) = -(\mathbf{l} - \mathbf{m}_m)^T \mathbf{\Lambda}_m (\mathbf{l} - \mathbf{m}_m) + \text{const} \tag{19}$$

ACCEPTED MANUSCRIPT

where $\mathbf{l}$ is a vector containing the concatenation of all part locations $\mathbf{l}_i$. In this case $\mathbf{\Lambda}_m$ is a block sparse precision matrix which contains non-zero entries only for pairs of connected parts. This is in contrast to other facial shape models used in the literature, e.g., Constrained Local Models (CLMs) [172] and Active Appearance Models (AAMs)[173], where a full precision matrix is used for modelling the facial shape. The above shape parametrization allows the development of efficient dynamic programming algorithms which are able to find globally optimal solutions. On the other hand, tree-based models unavoidably allow the facial shape to deform more freely and thus produce some unnatural facial deformations.

In order to fit the model in various scales, the root or part filters are placed in different scales, and the cost functions have to be appropriately modified (the interested reader may refer to [35]) [2].

Formulation (18) leads to a convenient dot product form of the cost function as:

$$C(L|I, m) = \mathbf{w}_m^T \mathbf{y} \tag{20}$$

where the vector $\mathbf{w}_m$ concatenates the parameter values for appearance and deformation as:

$$\mathbf{w}_m = [\mathbf{w}_m^1, \ldots, \mathbf{w}_m^n, \ldots, a_{ij}^m, b_{ij}^m, c_{ij}^m, d_{ij}^m, \ldots, \alpha^m].$$

Finally, face detection is performed by minimizing $C(L|I, m)$ over L and m as

$$C^*(I) = \min_{m,L} C(L|I, m). \tag{21}$$

The minimum is computed by enumerating all components of the mixture model, and for each component we find the best configuration over parts L. Since we assume tree facial structures, the application of GDTs renders the complexity of problem (21) linear with regards to the number of components, parts and image resolution. In order to detect multiple faces and eliminate repeated detections, non-maximum suppression methodologies are applied [35]. Recently, it was shown that the choice of the non-maximum suppression threshold plays a crucial role in performance [26].

3.1.1. Weakly Supervised Setting

The two main annotation settings for estimating the parameters of DPMs are weakly and strongly supervised. In the weakly supervised setting, only the bounding boxes of the positive examples (i.e., faces) and a set of negative examples are available. In this case the mixtures and the part locations in the training set are considered as hidden (i.e., latent) information revealed during training [35]. Of course the number of mixtures and the number of parts have to be selected a priori.

Let us introduce a latent variable vector $\mathbf{z} = [m, \mathbf{l}_1, \ldots, \mathbf{l}_{n_m}]$. Our goal is to learn a general vector of parameters $\mathbf{w} = [\mathbf{w}_1 \ldots \mathbf{w}_n]$. Since only one of the mixture components (tree models) can be activated, we define a general sparse feature vector $\mathbf{y}(\mathbf{z}) = [\mathbf{0}, \ldots, \mathbf{y}(\tilde{\mathbf{z}}), \ldots, \mathbf{0}]$, which is the score for the hypothesis $\tilde{\mathbf{z}} = [\mathbf{l}_1, \ldots, \mathbf{l}_{n_m}]$. The classifier that scores an example $\mathbf{q}$ using the space of possible latent values for this example $\mathcal{Z}(\mathbf{q})$ has the form:

$$f_{\mathbf{w}}(\mathbf{q}) = \max_{\mathbf{z} \in Z(\mathbf{q})} \mathbf{w}^T \mathbf{y}(\mathbf{q}, \mathbf{z}). \tag{22}$$

In order to find the parameters $\mathbf{w}$ given a set of training examples $\mathcal{D} = ((\mathbf{q}_1, y_1), \ldots, (\mathbf{q}_N, y_N))$ and $y_i \in \{-1, 1\}$, which are images with a bounding box annotation, we minimize the SVM objective function using the standard hinge loss:

$$C_{\mathcal{D}}(\mathbf{b}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{j=1}^{N} \max(0, 1 - y_j f_{\mathbf{w}}(\mathbf{q})), \tag{23}$$

which can be reformulated as:

$$\begin{array}{ll} \mathbf{w}^* & = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{j=1}^{N} \max(0, 1 - y_j \mathbf{w}^T \mathbf{y}(\mathbf{q}, \mathbf{z}^*)) \\ \text{s.t} & \mathbf{z}^* = \max_{\mathbf{z} \in Z(\mathbf{q})} \mathbf{w}^T \mathbf{y}(\mathbf{q}, \mathbf{z}). \end{array} \tag{24}$$

[2] The modifications are straightforward and we do not apply it in order not to clatter the text with notations.

ACCEPTED MANUSCRIPT

The minimization of the above cost function with regards to the latent variables and the model parameters $\mathbf{w}$ is highly non-convex, but it becomes convex once the latent information is specified for the positive training examples (i.e., facial samples). In [35] an alternating optimization procedure was proposed: by fixing $\mathbf{w}$, the highest scoring latent value for each positive example is determined; and then by fixing the latent values for the positive set of examples, $\mathbf{w}$ is updated by minimizing the SVM cost in (23). For the latter task a stochastic gradient descent was applied. The above methodology, called latent-SVM, is similar to the multi-instance SVMs proposed in [174]. Furthermore, it can be viewed as a discriminative clustering approach which alternates between assigning cluster (mixture) labels for each positive example, estimating cluster means (root filters) and parts.

3.1.2. Strongly Supervised Setting

In a strongly supervised setting, we assume that (a) the mixture components have been labelled in the training database and (b) the training database contains annotated facial images with facial landmarks. In this case there are no hidden latent variables $\mathbf{z}_j$ to be estimated for each training sample $\mathbf{q}_j$ during training. Hence only the model parameters $\mathbf{w}$ should be learned from the optimization problem

$$\begin{array}{rl} \mathbf{w} & = \arg\min_{\mathbf{w},\xi_j} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{j=1}^{N}\xi_j \\ \text{s.t.} & \forall\ \mathbf{q}_j \in C^+,\ \mathbf{w}^T\mathbf{y}(\mathbf{q}_j,\mathbf{z}_j) \geq 1-\xi_j \\ & \forall\ \mathbf{q}_j \in C^-,\ \mathbf{w}^T\mathbf{y}(\mathbf{q}_j,\mathbf{z}) \leq -1+\xi_j \\ & \forall k \in \mathcal{K},\ w_k \leq 0 \end{array} \tag{25}$$

where $\mathcal{K}$ is the set of indices of $\mathbf{w}$ that correspond to the quadratic terms of the shape cost a_{ij}^m and c_{ij}^m. The above constraints state that the score for the positive examples should be larger than 1 (minus the small slack variable value), while the score of the negative examples, for all configurations of part positions and components should be less than -1 (adding the small slack variable value). The last set of constraints guarantee that the shape cost is a proper metric.

The above optimization problem is a quadratic program, and can be efficiently solved in theory. However, in practice, since there are exponentially many constraints, direct solving of (25) is infeasible. In the SVM literature, the family of optimization problems such as (25) is known as structural SVM. Many efficient methods have been proposed to reduce the number of active constraints including cutting plane algorithms (e.g., SVMStruct [33, 175]). Other efficient methods are based on dual coordinate-descent solvers [176]. In [48] for face detection, a modified coordinate descent method was applied, which allows the incorporation of the last set of negativity constraints. In simple terms the coordinate descent solver of [48] iterates between finding $\mathbf{w}$ in the dual and mining violated constraints according to current estimate of $\mathbf{w}$ and adding them in the constraint pool, until convergence is met.

An important aspect of the complexity of the solution is how many components/views will be chosen and whether there will be part sharing between components/views. For example, there is an order of magnitude difference in complexity between a fully shared model and an independent model which does not share any part templates across any mixtures. In [48], it was shown that for face detection the performance drop when using a shared-parts model is insignificant when compared with an independent model.

3.2. Weakly Supervised vs Strongly Supervised Model DPMs for Face Detection

The weakly supervised model requires much less annotation effort, since it requires only facial bounding box annotations (thousands of annotations can be produced within hours). On the other hand, annotation of semantically meaningful landmarks is a tedious and labour intensive procedure which requires hundreds of hours of annotation for a few thousand facial images. Naturally two questions come into mind: (a) Do we need a weakly or a strongly supervised DPMs for face detection? and (b) Should face and facial landmark be considered as different problems that require the development of different methodologies? Interestingly, recent works seem to suggest contradictory conclusions regarding these two questions.

To answer the first question, we will discuss the empirical findings of the recent papers [48] and [168, 26]. The study [48] introduced the strongly supervised DPMs and showed that they largely outperform facial DPMs that are trained with weak annotations in face detection. On the contrary, the recent study in [26] showed that a weakly supervised DPM having only the components fixed a-priori (a total of 6 components covering frontal faces and side views) can outperform all current state-of-the-art face detection methodologies including the strongly supervised

ACCEPTED MANUSCRIPT

DPM in [48] [3]. The main performance improvement was due to the change of the threshold in the non-maximum suppression step from 0.5 to 0.3. It is unclear whether this change in threshold would also increase the performance of the strongly supervised DPM in [48].

Facial landmark localization is very important for many applications such as face and facial expression recognition, and has recently attracted considerable attention. It is advantageous to develop a methodology that could perform both face and facial landmark detection at once. However, empirical evidence shows that even though the strongly supervised DPMs are able to produce state-of-the-art face detection results, they are not as accurate in facial landmark localization. The main problem is the flexibility of the texture and shape models. For example, current state-of-the-art landmark localization algorithms are usually based on variations of Active Appearance Models that use robust features such as IGOs, HoGs etc. [177, 178] or cascade regression schemes based on SIFT or HoG features [179]. In all cases holistic texture models are used by assembling feature vectors that concatenate all facial texture features (i.e., they do not treat each part separately). Furthermore, in the AAM variations [177, 178], the shape is modelled using a fully connected precision matrix. In contrast, DPMs use part-based texture models and a flexible tree-based model, which may lead to unnatural face deformations. As discussed above, the problem with fully connected shape models is that they are not easy to solve for the global optima. Certainly, the detection results produced by a tree-based DPM can be used as good initializations for Gauss-Newton [177, 178] or regression methods [179].

Variations of DPMs have been proposed recently for improving face detection. For example, the recent work in [164] proposed some modifications of the strongly supervised facial DPM in [48]. In particular, a global root template was introduced for the description of the face at low-resolution. The parts were placed at higher resolution (twice the size of the lower level). Furthermore, in addition to the connections between facial landmarks at the higher level, spatial constraints were introduced between the locations of the root and the parts. Finally, a part subtype was introduced to account for appearance variations of the facial parts. A total of $K = 4$ subtypes were used (found through cross-validation), and the cluster annotations were produced by K-means clustering. In the same paper, a human body DPM was trained to assist face detection, and a learning methodology that combines face and body activations scores was also proposed [164].

3.3. Making DPMs faster

One of the main drawbacks for the mixtures of DPMs model is computational cost. For example, the fully independent model in [48], which contains 1050 different part filters and does not share any templates across any mixtures, needs around 40 seconds per image to detect faces at the minimum scale of 80×80 pixels. Even the fully shared model needs many seconds per image, making it difficult to apply DPMs for real-time applications. Recently, efforts have been made to develop DPMs with decreased computational complexity achieving near real-time performance [165, 166, 167, 168, 169]. The main steps of DPMs that allow optimization include (a) feature extraction using HoGs, (b) filter design, (c) computation of filter responses (correlations) and (d) score computation (mainly using cascade methods).

An approach for accelerating DPM fitting was proposed in [167]. It designs a framework that exploits the Fourier transform to compute the filter correlations. The framework also carefully controls the memory usage required to store the transforms of the filters by building patch-works combining multiple image scales. In [166], a different scheme for accelerating star-shaped DPMs was proposed. It was designed based on a Branch-and-Bound framework, adopting a Dual Trees data structure [180].

Another general approach for accelerating DPMs is to build a cascade of classifiers from DPMs [165]. By using a sequence of learned thresholds, the methodology prunes the computation based on partial hypothesis. Even though cascade methods are not exact, they are empirically verified to cause little performance degradation, and at least an order of magnitude increase in speed. Recently, in [168], a fast neighbourhood-aware cascade strategy was proposed by applying a scheme inspired by the crosstalk method [181] used in boosting classifiers. In particular, motivated by the fact that if a hypothesis has a low score, then it is quite possible that its neighbours also have low scores, the method aggressively prunes hypothesis by applying a first order approximation of scores from its neighbourhoods, instead of explicit computation.

[3] To the best of our knowledge, the face detector in [48] was trained using only the MultiPIE database, which has been recorded in controlled conditions

ACCEPTED MANUSCRIPT

Another computation intensive module in the DPM pipeline is the computation of HoG features at various scales [35]. Even well-implemented HoG pyramids may require up to 0.5 seconds to compute the features on CPU [37]. In a recent paper [168], a Look-Up-Table (LUT) approach was proposed for more efficient feature extraction, and it achieves 6 times faster implementation than the standard one used in DPMs [37].

Finally, another step that requires optimization is feature filter design. In the major DPM approaches, the root scores are computed densely by applying 2D correlations between the learned root filter and the HoG features of the test image. In addition to using FFTs to reduce the computational cost, one can further decompose the root filter into a linear combination of rank-1 filters and apply efficient one dimensional correlations. In [168], instead of decomposing the learned filter, the authors proposed to learn from the beginning a low-rank discriminant filter by incorporating the nuclear norm of the filter in the cost function of the SVM learning problem. The nuclear-norm regularized optimization problem can be solved by applying a proximal gradient algorithm. Overall, by using a LUT for computing HoGs, a reduced rank filter for the root, and a cascade for score computation, the work in [168] reported almost 40 times increase in speed over the original DPM in [35] and 5 times speed increase over the cascade DPM in [165].

3.4. Other Part-Based Methods for Face Detection

Here we survey some part-based methods, such as [49] and [50], that do not fall in the above described DPM architecture and have attracted a lot of attention.

In [49] an object detector based on detecting localized parts of the object was proposed. Each part is a group of pixels or transform variables that are statistically dependent, and between parts it is assumed to be statistically independent. AdaBoost was used to compute each part's likelihood of belonging to the detected object. The final decision was made by multiplying the likelihood ratios of all the parts together and testing the result against a predefined threshold. In a later work [182] the cases where the statistical dependency cannot be easily decomposed into separate parts was further examined. The method learns the dependency structure of a Bayesian network based classifier. Although the problem is known to be NP complete, [182] presented a scheme that selects a structure by seeking to optimize a sequence of two cost functions: the local modelling error using the likelihood ratio test as before, and the global empirical classification error computed on a cross-validation set of images. The very successful commercial PittPatt face detection software, now acquired by Google, that combines the above approach with the feature-centric cascade detection scheme in [127] showed state-of-the-art performance on public evaluation tests [183].

In [49] wavelet variables to represent parts of the faces, which do not necessarily correspond to semantic components were proposed. In the literature, there had been many component-based object detectors that relied on semantically meaningful component detectors [15, 184, 185].

In [186], 100 textured 3D head models to train 14 component detectors were used. These components were initialized by a set of reference points manually annotated for the head models, and their rectangles were adaptively expanded during training to ensure good performance. The final decision was made by a linear SVM that combines all the output from the component detectors.

Another closely related approach is to detect faces/humans by integrating a set of individual detectors that may have overlaps with each other. For instance, in [50] 7 detectors to find body parts including frontal and profile faces, frontal and profile heads, frontal and profile upper body, and legs, were applied. A joint likelihood body model was then adopted to build a body structure by starting with one part and adding the confidence provided by other body part detectors.

A summary of the approaches discussed in this section can be found in Table 4.

4. Databases and Benchmarks

For the past twenty years the face analysis community had made great efforts in collecting databases [187, 188, 148, 189, 190, 191]. One of the most well-known databases is FERET face database. It was collected mainly for testing face recognition algorithms, and has been occasionally used for assessing the performance of early face detection algorithms. For a recent survey on face recognition databases, the interested reader may refer to [192].

It was evident that the databases collected in controlled conditions such as FERET [188], XM2VTS [190], PIE [191] and FRGC [187] did not include the facial appearance variations suitable either to train or to test face detection

ACCEPTED MANUSCRIPT

Table 4. Deformable and Part-based Face Detection Algorithms.

General Approach	**Representative Works**
Deformable parts-based methods (DPMs)	Strongly Supervised [48, 164]
	Weakly Supervised [26, 168]
Other Part-based Methods approaches	Wavelet localized parts [49, 182]
	SVM component detectors adaptively trained [186]
	Overlapping part detectors [50]

algorithms. Some of the first efforts to collect facial samples in arbitrary recording conditions were made in [193, 194, 195, 196]. For example the authors of [193] scanned images from newspapers to collect face samples. The main testbeds for 'in-the-wild' face detection were the databases developed in [193, 194, 195, 196], but contained a small number of faces in total.

During the past 10 years, with the huge development of the Internet, many efforts were made to collect and annotate considerable amount of images in unconstrained conditions. Arguably the pivotal efforts were the PASCAL Visual Object Classes (VOC) benchmarks and challenges [27, 28], and the collection of faces 'in-the-wild' for face verification and detection purposes [148, 189]. Recently, another benchmark was developed for assessing the performance of face detection and facial landmark localization algorithms [48], called Annotated Faces in-the-Wild (AFW) testset.

PASCAL VOC challenges included multi-object detection 'in-the-wild' images, and one of the objects used was the human face. Furthermore, the current PASCAL VOC challenges includes a person layout detection challenge [27, 28]. The set of these images (851 images with bounding boxes), called PASCAL faces has been recently used for training and testing face detection algorithms [164, 26]. An effort to develop a comprehensive benchmark for face detection was made in [148], introducing the FDDB database and benchmark. The FDDB database contains 2845 images with a total of 5171 faces 'in-the-wild'. The images have been rigorously annotated using ellipses (instead of the standard rectangular bounding boxes).

The AFW databases contains 205 images of 468 faces annotated with regards to a bounding box. Furthermore, the faces have been annotated with regards to 6 facial landmarks (the center of eyes, tip of nose, the two corners and center of mouth) and labelled discretized viewpoints (-90^o to 90^o every 15^o) along pitch and yaw directions and (left, center, right) viewpoints along the roll direction. This database was recently expanded to 68 landmarks [197]. Another database that was used for training 'in-the-wild' face detection algorithms is the AFLW database [198] which has been annotated with 21 landmarks. These landmarks were used to drive the annotation process of the AFLW database with regards to facial bounding boxes [26]. Finally, other databases that can be used for training face detection algorithms are the LFPW [199], HELEN [200] and iBUG databases [197], since facial landmark annotations are provided by the database creators.

The standard measure to assess the quality of face detection is to compute the match using the ratio of intersected areas to joined areas. Assume d is the detected face and a is the annotation, then:

$$S(d,a) = \frac{A(d) \cap A(a)}{A(d) \cup A(a)} \tag{26}$$

where $A()$ is the area operator. As in other object detection challenges, precision vs. recall curves are plotted by requiring the above degree of match to be greater that 0.5, i.e. $S(d,a) > 0.5$ (in other words 50% overlap between the annotated and the detected regions) [48]. The FDDB database comes with its own evaluation protocol [148]. Furthermore, it calls for sharing the ROC curves but not the actual detected bounding boxes. More precisely, the FDDB evaluation protocol recommends providing two ROC curves that both plot the false positives over the true

ACCEPTED MANUSCRIPT

Table 5. Performance comparison of the state-of-the-art in face detection. For each method the true positive rate is shown for (a) almost no false positives (around 10), (b) around 100 false positives and (c) 1000 false positives.

Method	♯ of False Positives ≈ 10	♯ of False Positives ≈ 100	♯ of False Positives ≈ 1000
K. Mikolajczyk et. al. [50]	10%	33%	54.8%
P. Viola and Jones, OpenCV version [39]	10%	33%	59.7%
V. Jain and E. Learned-Miller [202]	15.7%	51%	67.7%
X. Zhu and D. Ramanan [48]	63.8%	73.3%	76.6%
X. Shen et. al. [44]	8%	67.5%	78.6%
J. Li and Y. Zhang [103]	69.4%	80.6%	83.7%
H. Li et. al. [203]	10%	73.3%	80.9%
H. Li et. al. [43]	69.2%	80.8%	84.8%
J. Yan et. al. [168]	75.9%	81.3%	85.2%
D. Chen et. al. [51]	78.8%	83.9%	86.2%
M. Mathias et. al. [26]	72.5%	83.4%	87%
B. Yang [105]	75.4%	81.6%	85.2%
J. Yan et. al. [164]	-	~80%	84.6%
B. Jun et. al. [22]	~ 67%	~ 77%	~ 80.6%

positives by using a discrete or continuous version of the match degree. For the discrete case the threshold is similarly set to 0.5, while in the continuous setting, the overlapping ratio is used as a weight for every detection window.

Figures 13 (a) and (b) and 14 (a) and (b) plot the Receiver Operating Characteristic (ROC) curves for both continoous and discrete degree of match of state-of-the-art techniques, published in the recent top venues of the field [4], as well as some ten years old popular representative techniques, such as the OpenCV version of Viola-Jones [36, 39] and the method in [50]. Finally, a summary of the results is presented in Table 5. In this Table, we show particular operating points of the ROC curve for the discrete degree of match that correspond to approximate no false positives (i.e., false positives around 10), false positives around 100 and false positives around 1000.

By inspecting the above results we can conclude that

- in the last 10 years there has been a dramatic increase in performance (i.e., true positive rate). In particular, when limiting the number of false positives to be around zero (i..e, not more than 10 false positives in all tested images) has increased more than 65% in absolute terms;
- the performance increase is attributed, mainly, to combining ideas from Viola-Jones boosting and robust features [26] ;
- there is still a gap in performance of around 15-20%. That is, even when allowing a relative large number of false positives (around 1,000), there are still around 15-20% of faces that are not detected. It is speculated that this gap is due to out-of-focus faces (i.e., blurred faces) [26], nevertheless a detailed analysis of errors is missing from the literature;
- in this benchmark the performance of the best boosting-based technqiue and the best DPM technique is approximately equal [26] . Also, the best performing technique is a hybrid one that combines ideas from boosting, as well as deformable parts-based models [51] .

5. Discussion, Future Work and Conclusions

In the early years the families of algorithms that were applied for face detection were quite diverse [15, 16]. Now that face detection approaches an age of maturity the main lines of research revolve around three large families: (a) boosting-based methods, (b) application of Deep Convolutional Neural Networks (DCNNs) and (c) Deformable Parts-based Models (DPM) methods.

In previous surveys face detection algorithms were separated to various categories based on whether they were robust to illumination changes, facial expressions, facial pose etc. Such, categorization hardly applies any more,

[4]The ROC curves have been found from the publicly available repository of results in the FDDB database [201], as well as contacting some of authors of the paper.

ACCEPTED MANUSCRIPT

since the inability of some methods to explicitelity account for a number of variations is alleviated by the presence of large ammount of publicly available facial data. For example, algorithms that learn rigid templates using boosting, as well as detectors based on DCNNs, have difficulties to handle unseen views, something that is alleviated by (a) using the large amount of available data under different views, (b) enriching the data by creating novel views (e.g., by performing various image transformations) and using many rigid templates for both frontal and rotated faces (e.g., in [26] in each scale 5 templates for frontal and 6 templates for rotated faces were used). On the other hand, face detection methodologies based on DPMs, show better generalization performance to novel views, as they explicitely model deformations. A direct concequence is that DPM methodologies can be effectively trained using a smaller ammount of data.

Another example is that all state-of-the-art methods based on boosting or DPMs account for illumination variations using both robust features, such as HoGs etc., as well as using many different faces under different illuminations. Furthermore, DCNNs find features which are robust to illumination changes by exploiting their non-linear multi-layer architecture, as well as the large ammount of available training data.

In Table 6 we try to relatively quantify some of the operational aspects of its family of algorithms using four different characteristics:

- Annotation Effort: This measures how expensive the annotation that needs to provided for each family of algorithms is. Rigid template-based algorithms such as boosting and DCNN based require a consistent good quality bounding box annotation of the facial region of interest, which is a task of medium effort. Similarly, weakly supervised DPM methods [35] can be trained using only bounding box -style annotations. On the contrary, strongly supervised DPMs [48] require the presence of annotations with regards to facial parts or landmarks, hence for these kind of methods the annotation effort is large.

- Training Data: This measures the amount of training data required to achieve state-of-the-art performance. Rigid template-based algorithms, since they do not explicitly model facial deformations and facial pose, require a large number of data in order to be robustly trained. Both, boosting-based and DCNNs-based algorithms take deformations and facial pose implicitly into account by selecting (boosting) or learning (DCNN) a number of features invariant to deformation and/or pose. Since, DPMs explicitly model deformations and facial pose (by using mixtures), they require significantly less data to train.

- Training Time: This measures how computationally intensive is the training procedure for each of the families. Applying the Viola-Jones methodology to the hundredths of thousands of channel of features may require weeks to train [25]. This was one of the biggest drawback of the first boosting-based methods. Recently, with the application of 'soft-cascade' strategies it is feasible to train a rigid template in hours [25]. Hence, for boosting-based methods the amount of training time largely varies a lot depending both (a) the boosting strategy and (b) how the cascade of weak-classifiers is constructed. Even though significant progress has been made towards parallel and distributed training of deep neural architectures [34],[204], training of DCNNs with many layers is still very computationally demanding. Finally, training of DPMs depends on the training strategy used. In particular, generative DPMs, as in [47], can be trained very fast (even real time). The state-of-the-art discriminatively trained DPMs require to solve (a) an optimization problem like (24) which alternates between the model parameters and part locations (weakly supervised) or (b) an optimization problem like (25) where only the optimal model parameters need to be recovered, since the part locations are provided (strongly supervised). The computational complexity of these optimization problems depends on the number of training samples but generally it is feasible to train a model with thousands of training samples in few hours.

- Testing time: The huge advantage of boosting-based methods is that they offer real time face detection in test images (or near real-time when many channel features are used such as HoG etc.). DCNNs-based methodologies for detection can offer near real-time performance when combined with methodologies that return region-proposals [205][5]. Finally, computational complexity of DPM-based methods varies on the number of mixtures and parts used. Nevertheless, it is possible to build state-of-the-art DPMs with near real-time performance [168].

[5]Methods that provide proposals usually scan the image and return a number of regions that could potentially contain any kind of object by measuring a kind of generic objectness measure [206].

ACCEPTED MANUSCRIPT

Table 6. Comparison of the three main families of face detection algorithms with regards to (a) the effort required for annotation, (b) the number of training data required, (c) the training time required and (d) the test time.

Family of Methods		Annotation Effort	Training data	Training Time	Test Time
Boosting-based	Simple Haar-like features	Medium	Large	Medium-Large	Real-Time
	Channel Features	Medium	Large	Medium-Large	(near) Real-Time
DPMs	Weakly Supervised	Medium	Medium	Medium	(near) Real-Time
	Strongly Supervised	Large	Medium	Medium	(near) Real-Time
DCNNs		Medium	Large	Large	(near) Real-Time

Finally, we would like to highlight that the best performing algorithm for both families achieved very similar performance in the FDDB benchmark [26]. Finally, the best performing approach [51] combines ideas from both families.

Even though it was recently demonstrated that machines have started to become more efficient than humans in face recognition in unconstrained conditions [207, 208], humans still largely surpass machines in face detection [209]. Nevertheless, it is exciting to see face detection techniques be increasingly used in real-world applications and products. For instance, most digital cameras today have built-in face detectors, which can help the camera to do better auto-focusing and auto-exposure. Digital photo management pieces of software such as Apple's iPhoto, Google's Picasa and Microsoft's Windows Live Photo Gallery all have excellent face detectors to help tagging and organizing people's photo collections. On the other hand, face detection in completely unconstrained settings remains a very challenging task, particularly due to the significant variations in appearance [26], as well as out-of-focus images and blurring. The current state-of-the-art result in the standard benchmark, i.e., FDDB, is around 80% recall with almost no false positives [51, 26]. An interesting topic for further research is to perform a meticulous diagnostic analysis of the errors of the state-of-the-art detectors [210].

Technologically, the current success is indisputably attributed to the development of robust features such as HoGs [19], as well as the combination of these features with (a) ideas from the original Viola-Jones paper (i.e., ICF and boosting [26]) and (b) discriminatively trained parts-based models [35]. Furthermore, with the current development of the field of deep network architectures [211] it is expected to see state-of-the-art face detectors adopting ideas from this field [42]. Another interesting line of further research in object detection, in general, and face detection in particular, is how to combine the part-based methods with boosting based methods. An interesting method in this line is the recent method in [51] which combines cascade face alignment with cascade face detection, based on the observation that aligned facial shapes may provide better features for face detection. An effective approach is proposed by jointly learning face alignment and detection in the same cascade framework. Another interesting recent method along these lines is the combination of features produced by off-the-self pre-trained DCNN architectures with DPMs [212, 213].

Another interesting idea to improve face detection performance is to consider the contextual information. Human faces are most likely linked with other body parts, and these other body parts can provide a strong cue of faces. There has been some recent work on context based object categorization [214] and visual tracking [215]. One scheme that uses local context to improve face detection was also presented in [216], and we think that this is also a very promising research direction to pursue.

The modern face detectors are mostly appearance-based methods, which means that they need data to train classifiers. Collecting a large amount of ground truth data remains a very expensive task, which certainly demands more research. Schemes such as multiple instance learning boosting and multiple category boosting are helpful in reducing the accuracy needed for the labeled data, though ideally one would like to leverage unlabeled data to facilitate learning. Unsupervised or semi-supervised learning schemes would also be ideal to reduce the amount of work needed

ACCEPTED MANUSCRIPT

for data collection. In this line of research, another question is how to transfer knowledge from the current face detection methods trained with images captured from standard cameras to face detectors for special cameras, such as omnidirectional cameras [217].

Another interesting line of research is two create simulated data for training using available 3D facial databases. Recently it was demonstrated that it is possible to train a state-of-the-art face detection system using an adaptation of Viola-Jones algorithm using only artificially generated data from a facial 3D Morphable Model (3DMM) [218]. In particular, adaptive training samples are generated from a 3DMM trying to simulate the variability of actual 'in-the-wild' facial images including variations based on age and body weight. Moreover, it was shown how the procedure can automatically adapt to environmental constraints, such as illumination or viewing angle of recorded video footage from surveillance cameras.

It remains an open question whether a face detector can detect faces in arbitrary collections. For example, how do current face detectors trained on standard databases perform in digitized images of the 19th century or in a collection of out-of-focus images from surveillance cameras for a particular environment? Instead of retraining the detectors from scratch when a new collection is available, an interesting topic of further research is to develop techniques which can adapt to a new image dataset without having access to the original training data [202]. That way camera and environment specific face detectors with very high performance could be routinely developed. Unlike other domains such as speech recognition and handwriting recognition, where adaptation has been indispensable, adaptation for visual object detection has received relatively little attention. Some early work has been conducted in this area [219, 220, 202, 203] and we strongly believe that this is a great direction for future work.

6. Acknowledgement

Stefanos Zafeiriou acknowledges support from the EPSRC projects Analysis of Facial Behaviour for Security in 4D (4D-FAB) (EP/J017787/1) and Adaptive Facial Deformable Models for Tracking (ADAManT) (EP/L026813/1).

References

[1] W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld, Face recognition: A literature survey, Acm Computing Surveys (CSUR) 35 (4) (2003) 399–458.
[2] Z. Kalal, K. Mikolajczyk, J. Matas, Face-tld: Tracking-learning-detection applied to faces, in: Image Processing (ICIP), 2010 17th IEEE International Conference on, IEEE, 2010, pp. 3789–3792.
[3] M. Pantic, L. J. M. Rothkrantz, Automatic analysis of facial expressions: The state of the art, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (12) (2000) 1424–1445.
[4] N. Kumar, A. C. Berg, P. N. Belhumeur, S. K. Nayar, Attribute and simile classifiers for face verification, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 365–372.
[5] Y. Fu, G. Guo, T. S. Huang, Age synthesis and estimation via faces: A survey, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (11) (2010) 1955–1976.
[6] A. Laurentini, A. Bottino, Computer analysis of face beauty: A survey, Computer Vision and Image Understanding.
[7] Y. Wang, L. Zhang, Z. Liu, G. Hua, Z. Wen, Z. Zhang, D. Samaras, Face relighting from a single image under arbitrary unknown lighting conditions, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (11) (2009) 1968–1984.
[8] V. Blanz, T. Vetter, A morphable model for the synthesis of 3d faces, in: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.
[9] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, S. M. Seitz, Exploring photobios, in: ACM Transactions on Graphics (TOG), Vol. 30, ACM, 2011, p. 61.
[10] A. Robotics, Nao robot, http://www.aldebaran.com/en (cited August 2014).
[11] W. W. Bledsoe, H. Chan, A man-machine facial recognition system?some preliminary results, Panoramic Research, Inc, Palo Alto, California., Technical Report PRI A 19 (1965) 1965.
[12] W. Bledsoe, Man-machine facial recognition, Rep. PRi 22.
[13] T. Sakai, M. Nagao, T. Kanade, Computer analysis and classification of photographs of human faces, Kyoto University, 1972.
[14] M. A. Fischler, R. A. Elschlager, The representation and matching of pictorial structures, IEEE Transactions on Computers 22 (1) (1973) 67–92.
[15] M.-H. Yang, D. J. Kriegman, N. Ahuja, Detecting faces in images: A survey, IEEE Trans. on PAMI 24 (1) (2002) 34–58.
[16] E. Hjelmas, B. K. Low, Face detection: A survey, Computer Vision and Image Understanding 83 (2001) 236–274.
[17] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proc. of CVPR, 2001.
[18] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.
[19] N. Dalal, B. Triggs, Histogram of oriented gradients for human detection, in: Proc. of CVPR, 2005.
[20] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. on PAMI 24 (2002) 971–987.

ACCEPTED MANUSCRIPT

[21] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local binary patterns, in: Proc. of ECCV, 2004.

[22] B. Jun, I. Choi, D. Kim, Local transform features and hybridization for accurate face and human detection, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (6) (2013) 1423–1436.

[23] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (surf), Computer vision and image understanding 110 (3) (2008) 346–359.

[24] E. Tola, V. Lepetit, P. Fua, Daisy: An efficient dense descriptor applied to wide-baseline stereo, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (5) (2010) 815–830.

[25] P. Dollar, Z. Tu, P. Perona, S. Belongie, Integral channel features., in: BMVC, Vol. 2, 2009, p. 5.

[26] M. Mathias, R. Benenson, M. Pedersoli, L. V. Gool, Face detection without bells and whistles, in: ECCV, 2014.

[27] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge–a retrospective, International Journal on Computer Vision.

[28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2) (2010) 303–338.

[29] G. B. Huang, R. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments., Tech. rep., University of Massachusetts, Amherst, Technical Report 07-49 (2007).

[30] V. Jain, E. Learned-Miller, FDDB: A benchmark for face detection in unconstrained settings, Tech. rep., University of Massachusetts, Amherst (2010).

[31] Y. Freund, R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: Computational learning theory, Springer, 1995, pp. 23–37.

[32] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297.

[33] I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, Support vector machine learning for interdependent and structured output spaces, in: Proceedings of the twenty-first international conference on Machine learning, ACM, 2004, p. 104.

[34] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[35] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (9) (2010) 1627–1645.

[36] G. Bradski, A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, " O'Reilly Media, Inc.", 2008.

[37] R. B. Girshick, P. F. Felzenszwalb, D. McAllester, Discriminatively trained deformable part models, release 5, http://people.cs.uchicago.edu/ rbg/latent-release5/.

[38] Y. Jia, Caffe: An open source convolutional architecture for fast feature embedding, http://caffe.berkeleyvision.org/ (2013).

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

[40] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[41] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, arXiv preprint arXiv:1311.2524.

[42] C. Zhang, Z. Zhang, Improving multiview face detection with multi-task deep convolutional neural networks, in: Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on, IEEE, 2014, pp. 1036–1041.

[43] H. Li, Z. Lin, J. Brandt, X. Shen, G. Hua, Efficient boosted exemplar-based face detection, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2014.

[44] X. Shen, Z. Lin, J. Brandt, Y. Wu, Detecting and aligning faces by image retrieval, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 3460–3467.

[45] B. Leibe, A. Leonardis, B. Schiele, Robust object detection with interleaved categorization and segmentation, International journal of computer vision 77 (1-3) (2008) 259–289.

[46] D. H. Ballard, Generalizing the hough transform to detect arbitrary shapes, Pattern recognition 13 (2) (1981) 111–122.

[47] P. F. Felzenszwalb, D. P. Huttenlocher, Pictorial structures for object recognition, International Journal of Computer Vision 61 (1) (2005) 55–79.

[48] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2879–2886.

[49] H. Schneiderman, T. Kanade, Object detection using the statistics of parts, International Journal of Computer Vision 56 (3) (2004) 151–177.

[50] K. Mikolajczyk, C. Schmid, A. Zisserman, Human detection based on a probabilistic assembly of robust part detectors, in: Proc. of ECCV, 2004.

[51] D. Chen, S. Ren, Y. Wei, X. Cao, J. Sun, Joint cascade face detection and alignment, in: European Conference on Computer Vision (ECCV) 2014, 2014.

[52] C. Zhang, Z. Zhang, A survey of recent advances in face detection, Tech. rep., Tech. rep., Microsoft Research (2010).

[53] F. Crow, Summed-area tables for texture mapping, in: Proc. of SIGGRAPH, Vol. 18, 1984, pp. 207–212.

[54] R. Meir, G. Rätsch, An introduction to boosting and leveraging, S. Mendelson and A. J. Smola Ed., Advanced Lectures on Machine Learning, Springer-Verlag Berlin Heidelberg (2003) 118–183.

[55] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Tech. rep., Dept. of Statistics, Stanford University (1998).

[56] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: European Conf. on Computational Learning Theory, 1994.

[57] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.

[58] R. E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (1999) 297–336.

[59] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, in: Proc. of ECCV, 2002.

ACCEPTED MANUSCRIPT

[60] C. Bishop, P. Viola, Learning and vision: Discriminative methods, in: ICCV Course on Learning and Vision, 2003.
[61] B. Wu, H. Ai, C. Huang, S. Lao, Fast rotation invariant multi-view face detection based on real adaboost, in: Proc. of IEEE Automatic Face and Gesture Recognition, 2004.
[62] T. Mita, T. Kaneko, O. Hori, Joint Haar-like features for face detection, in: Proc. of ICCV, 2005.
[63] R. Lienhart, J. Maydt, An extended set of Haar-like features for rapid object detection, in: Proc. of ICIP, 2002.
[64] M. Jones, P. Viola, Fast multi-view face detection, Tech. rep., Mitsubishi Electric Research Laboratories, TR2003-96 (2003).
[65] M. Jones, P. Viola, D. Snow, Detecting pedestrians using patterns of motion and appearance, Tech. rep., Mitsubishi Electric Research Laboratories, TR2003-90 (2003).
[66] S. C. Brubaker, J. Wu, J. Sun, M. D. Mullin, J. M. Rehg, On the design of cascades of boosted ensembles for face detection, Tech. rep., Georgia Institute of Technology, GIT-GVU-05-28 (2005).
[67] C. Huang, H. Ai, Y. Li, S. Lao, Vector boosting for rotation invariant multi-view face detection, in: Proc. of ICCV, 2005.
[68] R. Xiao, H. Zhu, H. Sun, X. Tang, Dynamic cascades for face detection, in: Proc. of ICCV, 2007.
[69] B. Fröba, A. Ernst, Face detection with the modified census transform, in: IEEE Intl. Conf. on Automatic Face and Gesture Recognition, 2004.
[70] G. Zhang, X. Huang, S. Z. Li, Y. Wang, X. Wu, Boosting local binary pattern (LBP)-based face recognition, in: Proc. Advances in Biometric Person Authentication, 2004.
[71] H. Jin, Q. Liu, H. Lu, X. Tong, Face detection using improved lbp under bayesian framework, in: Third Intl. Conf. on Image and Grahics (ICIG), 2004.
[72] L. Zhang, R. Chu, S. Xiang, S. Liao, S. Z. Li, Face detection based on multi-block LBP representation, 2007.
[73] S. Yan, S. Shan, X. Chen, W. Gao, Locally assembled binary (LAB) feature with feature-centric cascade for fast and accurate face detection, in: Proc. of CVPR, 2008.
[74] C. Liu, H.-Y. Shum, Kullback-Leibler boosting, in: Proc. Of CVPR, 2003.
[75] J. Meynet, V. Popovici, J.-P. Thiran, Face detection with boosted gaussian features, Pattern Recognition 40 (8) (2007) 2283–2291.
[76] X. Chen, L. Gu, S. Z. Li, H.-J. Zhang, Learning representative local features for face detection, in: Proc. of CVPR, 2001.
[77] P. Wang, Q. Ji, Learning discriminant features for multi-view face and eye detection, in: Proc. of CVPR, 2005.
[78] S. Baluja, M. Sahami, H. A. Rowley, Efficient face orientation discrimination, in: Proc. of ICIP, 2004.
[79] Y. Abramson, B. Steux, YEF* real-time object detection, in: International Workshop on Automatic Learning and Real-Time, 2005.
[80] K. Levi, Y. Weiss, Learning object detection from a small number of examples: The importance of good features, in: Proc. of CVPR, 2004.
[81] Q. Zhu, S. Avidan, M.-C. Yeh, K.-T. Cheng, Fast human detection using a cascade of histograms of oriented gradients, in: Proc. of CVPR, 2006.
[82] H. Grabner, H. Bischof, On-line boosting and vision, in: Proc. of CVPR, 2006.
[83] F. Suard, A. Rakotomamonjy, A. Bensrhair, A. Broggi, Pedestrian detection using infrared images and histograms of oriented gradients, in: IEEE Intelligent Vehicles Symposium, 2006.
[84] I. Laptev, Improvements of object detection using boosted histograms, in: British Machine Vision Conference, 2006.
[85] M. Enzweiler, D. M. Gavrila, Monocular pedestrian detection: Survey and experiments, IEEE Trans. on PAMI 31 (12) (2009) 2179–2195.
[86] C. A. Waring, X. Liu, Face detection using spectral histograms and SVMs, IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics 35 (3) (2005) 467–476.
[87] H. Zhang, W. Gao, X. Chen, D. Zhao, Object detection using spatial histogram features, Image and Vision Computing 24 (4) (2006) 327–341.
[88] X. Wang, T. X. Han, S. Yan, An HOG-LBP human detector with partial occlusion handling, in: Proc. of ICCV, 2009.
[89] O. Tuzel, F. Porikli, P. Meer, Region covariance: A fast descriptor for detection and classification, in: Proc. of ECCV, 2006.
[90] C. Huang, H. Ai, Y. Li, S. Lao, Learning sparse features in granular space for multi-view face detection, in: Intl. Conf. on Automatic Face and Gesture Recognition, 2006.
[91] J. Yuan, J. Luo, Y. Wu, Mining compositional features for boosting, in: Proc. of CVPR, 2008.
[92] F. Han, Y. Shan, H. S. Sawhney, R. Kumar, Discovering class specific composite features through discriminative sampling with Swendsen-Wang cut, in: Proc. of CVPR, 2008.
[93] X. Liu, T. Yu, Gradient feature selection for online boosting, in: Proc. of ICCV, 2007.
[94] A. Opelt, A. Pinz, A. Zisserman, A boundary-fragment-model for object detection, in: Proc. of CVPR, 2006.
[95] J. Shotton, A. Blake, R. Cipolla, Contour-based learning for object detection, in: Proc. of ICCV, 2005.
[96] B. Wu, R. Nevatia, Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors, in: Proc. of ICCV, 2005.
[97] P. Sabzmeydani, G. Mori, Detecting pedestrians by learning shapelet features, in: Proc. of CVPR, 2007.
[98] B. Wu, R. Nevatia, Simultaneous object detection and segmentation by boosting local shape feature based classifier, in: Proc. of CVPR, 2007.
[99] W. Gao, H. Ai, S. Lao, Adaptive contour features in oriented granular space for human detection and segmentation, in: Proc. of CVPR, 2009.
[100] M. Köstinger, P. Wohlhart, P. M. Roth, H. Bischof, Robust face detection by simple means.
[101] R. Benenson, M. Mathias, T. Tuytelaars, L. Van Gool, Seeking the strongest rigid detector, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 3666–3673.
[102] J. Li, T. Wang, Y. Zhang, Face detection using surf cascade, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 2183–2190.
[103] J. Li, Y. Zhang, Learning surf cascade for fast and accurate object detection, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2013, pp. 3468–3475.
[104] F. Porikli, Integral histogram: A fast way to extract histograms in cartesian spaces, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 829–836.

ACCEPTED MANUSCRIPT

[105] B. Yang, J. Yan, Z. Lei, S. Z. Li, Aggregate channel features for multi-view face detection, in: Biometrics (IJCB), 2014 IEEE International Joint Conference on, IEEE, 2014, pp. 1–8.
[106] L. Bourdev, J. Brandt, Robust object detection via soft cascade, in: Proc. of CVPR, 2005.
[107] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, Tech. rep., Microprocessor Research Lab, Intel Labs (2002).
[108] P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (11) (1994) 1119–1125.
[109] J.-S. Jang, J.-H. Kim, Fast and robust face detection using evolutionary pruning, IEEE Trans. on Evolutionary Computation 12 (5) (2008) 562–571.
[110] R. Xiao, L. Zhu, H. Zhang, Boosting chain learning for object detection, in: Proc. of ICCV, 2003.
[111] P. Viola, M. Jones, Fast and robust classification using asymmetric AdaBoost and a detector cascade, in: Proc. of NIPS, 2002.
[112] M.-T. Pham, T.-J. Cham, Online learning asymmetric boosted classifiers for object detection, in: Proc. of CVPR, 2007.
[113] H. Masnadi-Shirazi, N. Vasconcelos, Asymmetric boosting, in: Proc. of ICML, 2007.
[114] L. Mason, J. Baxter, P. Bartlett, M. Frean, Boosting algorithms as gradient descent, in: Proc. of NIPS, 2000.
[115] H. Masnadi-Shirazi, N. Vasconcelos, High detection-rate cascades for real-time object detection, in: Proc. of ICCV, 2007.
[116] J. Wu, S. C. Brubaker, M. D. Mullin, J. M. Rehg, Fast asymmetric learning for cascade face detection, Tech. rep., Georgia Institute of Technology, GIT-GVU-05-27 (2005).
[117] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, 2nd Edition, John Wiley & Sons Inc., 2001.
[118] J. Sochman, J. Matas, Waldboost - learning for time constrained sequential detection, in: Proc. of CVPR, 2005.
[119] A. Wald, Sequential Analysis, Dover, 1947.
[120] H. Luo, Optimization design of cascaded classifiers, in: Proc. of CVPR, 2005.
[121] C. Zhang, P. Viola, Multiple-instance pruning for learning efficient cascade detectors, in: Proc. of NIPS, 2007.
[122] B. McCane, K. Novins, On training cascade face detectors, in: Image and Vision Computing, 2003.
[123] J. Wu, J. M. Rehg, M. D. Mullin, Learning a rare event detection cascade by direct feature selection, in: Proc. of NIPS, Vol. 16, 2004.
[124] A. R. Webb, Statistical Pattern Recognition, 1st Edition, Oxford University Press, 1999.
[125] M.-T. Pham, T.-J. Cham, Fast training and selection of haar features during statistics in boosting-based face detection, in: Proc. of ICCV, 2007.
[126] F. Porikli, Integral histogram: A fastway to extract histograms in cartesian spaces, in: Proc. of CVPR, 2005.
[127] H. Schneiderman, Feature-centric evaluation for efficient cascaded object detection, in: Proc. of CVPR, 2004.
[128] M.-T. Pham, T.-J. Cham, Detection caching for faster object detection, in: Proc. of CVPR, 2005.
[129] M. M. Campos, G. A. Carpenter, S-tree: Self-organizing trees for data clustering and online vector quantization, Neural Networks 14 (4–5) (2001) 505–525.
[130] B. Fröba, A. Ernst, Fast frontal-view face detection using a multi-path decision tree, in: Proc. of Audio- and Video-based Biometric Person Authentication, 2003.
[131] Y.-Y. Lin, T.-L. Liu, Robust face detection with multi-class boosting, in: Proc. of CVPR, 2005.
[132] A. Torralba, K. P. Murphy, W. T. Freeman, Sharing features: Efficient boosting procedures for multiclass object detection, in: Proc. of CVPR, 2004.
[133] E. Seemann, B. Leibe, B. Schiele, Multi-aspect detection of articulated objects, in: Proc. of CVPR, 2006.
[134] B. Leibe, E. Seemann, B. Schiele, Pedestrian detection in crowded scenes, in: Proc. of CVPR, 2005.
[135] Y. Shan, F. Han, H. S. Sawhney, R. Kumar, Learning exemplar-based categorization for the detection of multi-view multi-pose objects, in: Proc. of CVPR, 2006.
[136] Z. Tu, Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering, in: Proc. of ICCV, 2005.
[137] B. Wu, R. Nevatia, Cluster boosted tree classifier for multi-view, multi-pose object detection, in: Proc. of ICCV, 2007.
[138] T.-K. Kim, R. Cipolla, MCBoost: Multiple classifier boosting for perceptual co-clustering of images and visual features, in: Proc. of NIPS, 2008.
[139] P. Viola, J. C. Platt, C. Zhang, Multiple instance boosting for object detection, in: Proc. of NIPS, Vol. 18, 2005.
[140] B. Babenko, P. Dollár, Z. Tu, S. Belongie, Simultaneous learning and alignment: Multi-instance and multi-pose learning, in: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, 2008.
[141] C. Zhang, Z. Zhang, Winner-take-all multiple category boosting for multi-view face detection, Tech. rep., Microsoft Research MSR-TR-2009-190 (2009).
[142] H. A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, in: Proc. of CVPR, 1996.
[143] D. Roth, M.-H. Yang, N. Ahuja, A SNoW-based face detector, in: Proc. of NIPS, 2000.
[144] R. Féraud, O. J. Bernier, J.-E. Viallet, M. Collobert, A fast and accurate face detector based on neural networks, IEEE Trans. on PAMI 23 (1) (2001) 42–53.
[145] C. Garcia, M. Delakis, Convolutional face finder: A neural architecture for fast and robust face detection, IEEE Trans. on PAMI 26 (11) (2004) 1408–1423.
[146] M. Osadchy, Y. L. Cun, M. L. Miller, Synergistic face detection and pose estimation with energy-based models, The Journal of Machine Learning Research 8 (2007) 1197–1215.
[147] Y.-N. Chen, C.-C. Han, C.-T. Wang, B.-S. Jeng, K.-C. Fan, A cnn-based face detector with a simple feature map and a coarse-to-fine classifier, Pattern Analysis and Machine Intelligence, IEEE Transactions on (99) (2009) 1–1.
[148] V. Jain, E. G. Learned-Miller, Fddb: A benchmark for face detection in unconstrained settings, UMass Amherst Technical Report.
[149] J. Bruna, S. Mallat, Invariant scattering convolution networks, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (8) (2013) 1872–1886.
[150] D. Keren, M. Osadchy, C. Gotsman, Antifaces: A novel fast method for image detection, IEEE Trans. on PAMI 23 (7) (2001) 747–761.
[151] C. Liu, A bayesian discriminating features method for face detection, IEEE Trans. on PAMI 25 (6) (2003) 725–740.
[152] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-Based Learning Methods, Cambridge Uni-

ACCEPTED MANUSCRIPT

versity Press, 2000.
[153] E. Osuna, R. Freund, F. Girosi, Training support vector machines: An application to face detection, in: Proc. of CVPR, 1997.
[154] B. Heisele, T. Poggio, M. Pontil, Face detection in still gray images, Tech. rep., Center for Biological and Computational Learning, MIT, A.I. Memo 1687 (2000).
[155] S. Romdhani, P. Torr, B. Schölkopf, A. Blake, Computationally efficient face detection, in: Proc. of ICCV, 2001.
[156] M. Rätsch, S. Romdhani, T. Vetter, Efficient face detection by a cascaded support vector machine using haar-like features, in: Pattern Recognition Symposium, 2004.
[157] B. Heisele, T. Serre, S. Prentice, T. Poggio, Hierarchical classification and feature reduction for fast face detection with support vector machines, Pattern Recognition 36 (2003) 2007–2017.
[158] Y. Li, S. Gong, H. Liddell, Support vector regression and classification based multi-view face detection and recognition, in: International Conference on Automatic Face and Gesture Recognition, 2000.
[159] H. A. Rowley, S. Baluja, T. Kanade, Rotation invariant neural network-based face detection, Tech. rep., School of Coomputer Science, Carnegie Mellow Univ., CMU-CS-97-201 (1997).
[160] J. Yan, S. Li, S. Zhu, H. Zhang, Ensemble svm regression based multi-view face detection system, Tech. rep., Microsoft Research, MSR-TR-2001-09 (2001).
[161] P. Wang, Q. Ji, Multi-view face detection under complex scene based on combined svms, in: Proc. of ICPR, 2004.
[162] K. Hotta, View independent face detection based on combination of local and global kernels, in: International Conference on Computer Vision Systems, 2007.
[163] J. Yan, X. Zhang, Z. Lei, D. Yi, S. Z. Li, Structural models for face detection, in: Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, IEEE, 2013, pp. 1–6.
[164] J. Yan, X. Zhang, Z. Lei, S. Z. Li, Structural models for face detection, Image and Vision Computing, accepted for publication.
[165] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, Cascade object detection with deformable part models, in: Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, IEEE, 2010, pp. 2241–2248.
[166] I. Kokkinos, Rapid deformable object detection using dual-tree branch-and-bound, in: Advances in Neural Information Processing Systems, 2011, pp. 2681–2689.
[167] C. Dubout, F. Fleuret, Exact acceleration of linear object detectors, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 301–311.
[168] J. Yan, Z. Lei, L. Wen, S. Z. Li, The fastest deformable part model for object detection, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014.
[169] J. Yan, X. Zhang, Z. Lei, S. Z. Li, Real-time high performance deformable model for face detection in the wild, in: Biometrics (ICB), 2013 International Conference on, IEEE, 2013, pp. 1–6.
[170] C. Chow, C. Liu, Approximating discrete probability distributions with dependence trees, Information Theory, IEEE Transactions on 14 (3) (1968) 462–467.
[171] P. F. Felzenszwalb, D. P. Huttenlocher, Distance transforms of sampled functions., Theory of computing 8 (1) (2012) 415–428.
[172] J. M. Saragih, S. Lucey, J. F. Cohn, Deformable model fitting by regularized landmark mean-shift, International Journal of Computer Vision 91 (2) (2011) 200–215.
[173] T. F. Cootes, G. J. Edwards, C. J. Taylor, Active appearance models, IEEE Transactions on pattern analysis and machine intelligence 23 (6) (2001) 681–685.
[174] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: Advances in neural information processing systems, 2002, pp. 561–568.
[175] T. Joachims, T. Finley, C.-N. J. Yu, Cutting-plane training of structural svms, Machine Learning 77 (1) (2009) 27–59.
[176] Y. Yang, D. Ramanan, Articulated pose estimation with flexible mixtures-of-parts, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1385–1392.
[177] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, S. Zafeiriou, Hog active appearance models, in: ICIP, 2014.
[178] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, M. Pantic, Generic active appearance models revisited, in: ACCV 2012, Springer, 2012, pp. 650–663.
[179] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 532–539.
[180] A. G. Gray, A. W. Moore, Nonparametric density estimation: Toward computational tractability., in: SDM, SIAM, 2003, pp. 203–211.
[181] P. Dollár, R. Appel, W. Kienzle, Crosstalk cascades for frame-rate pedestrian detection, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 645–659.
[182] H. Schneiderman, Learning a restricted bayesian network for object detection, in: Proc. of CVPR, 2004.
[183] M. C. Nechyba, L. Brandy, H. Schneiderman, Pittpatt face detection and tracking for the CLEAR 2007 evaluation, in: Classification of Events, Activities and Relations Evaluation and Workshop, 2007.
[184] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE Trans. on PAMI 23 (4) (2001) 349–361.
[185] S. M. Bileschi, B. Heisele, Advances in component-based face detection, in: Pattern Recognition with Support Vector Machines Workshop, 2002.
[186] B. Heisele, T. Serre, T. Poggio, A component-based framework for face detection and identification, International Journal of Computer Vision 74 (2) (2007) 167–181.
[187] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on, Vol. 1, IEEE, 2005, pp. 947–954.
[188] P. J. Phillips, H. Moon, S. A. Rizvi, P. J. Rauss, The feret evaluation methodology for face-recognition algorithms, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (10) (2000) 1090–1104.
[189] G. B. Huang, M. Mattar, T. Berg, E. Learned-Miller, et al., Labeled faces in the wild: A database for studying face recognition in uncon-

ACCEPTED MANUSCRIPT

strained environments, in: Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition, 2008.
[190] K. Messer, J. Matas, J. Kittler, J. Luettin, G. Maitre, Xm2vtsdb: The extended m2vts database, in: Second international conference on audio and video-based biometric person authentication, Vol. 964, Citeseer, 1999, pp. 965–966.
[191] T. Sim, S. Baker, M. Bsat, The cmu pose, illumination, and expression database, Pattern Analysis and Machine Intelligence, IEEE Transactions on 25 (12) (2003) 1615–1618.
[192] R. Gross, Face databases, in: Handbook of Face Recognition, Springer, 2005, pp. 301–327.
[193] K. Sung, T. Poggio, Example-based learning for view-based face detection, IEEE Trans. on PAMI 20 (1998) 39–51.
[194] A. C. Loui, C. N. Judice, S. Liu, An image database for benchmarking of automatic face detection and recognition algorithms, in: Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, Vol. 1, IEEE, 1998, pp. 146–150.
[195] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. on PAMI 20 (1998) 23–38.
[196] H. Schneiderman, T. Kanade, A statistical model for 3d object detection applied to faces and cars, in: Proc. of CVPR, 2000.
[197] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, 300 faces in-the-wild challenge: The first facial landmark localization challenge, in: Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, IEEE, 2013, pp. 397–403.
[198] M. Kostinger, P. Wohlhart, P. M. Roth, H. Bischof, Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 2144–2151.
[199] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, N. Kumar, Localizing parts of faces using a consensus of exemplars, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 545–552.
[200] V. Le, J. Brandt, Z. Lin, L. Bourdev, T. S. Huang, Interactive facial feature localization, in: ECCV 2012, Springer, 2012, pp. 679–692.
[201] Fddb: Face detection data set and benchmark, http://vis-www.cs.umass.edu/fddb/.
[202] V. Jain, E. Learned-Miller, Online domain adaptation of a pre-trained cascade of classifiers, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 577–584.
[203] H. Li, G. Hua, Z. Lin, J. Brandt, J. Yang, Probabilistic elastic part model for unsupervised face detector adaptation, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 793–800.
[204] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al., Large scale distributed deep networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1223–1231.
[205] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 580–587.
[206] B. Alexe, T. Deselaers, V. Ferrari, Measuring the objectness of image windows, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (11) (2012) 2189–2202.
[207] A. J. O'Toole, P. J. Phillips, F. Jiang, J. Ayyad, N. Pénard, H. Abdi, Face recognition algorithms surpass humans matching faces over changes in illumination, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (9) (2007) 1642–1646.
[208] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.
[209] W. Scheirer, S. Anthony, K. Nakayama, D. Cox, Perceptual annotation: Measuring human vision to improve computer vision, Pattern Analysis and Machine Intelligence, IEEE Transactions on, accepted for publication.
[210] D. Hoiem, Y. Chodpathumwan, Q. Dai, Diagnosing error in object detectors, in: ECCV 2012, Springer, 2012, pp. 340–353.
[211] S. Bengio, L. Deng, H. Larochelle, H. Lee, R. Salakhutdinov, Guest editors' introduction: Special section on learning deep architectures, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (8) (2013) 1795–1797.
[212] P.-A. Savalle, S. Tsogkas, G. Papandreou, I. Kokkinos, Deformable part models with cnn features, in: 3rd Parts and Attributes Workshop, ECCV, Vol. 8.
[213] R. Girshick, F. Iandola, T. Darrell, J. Malik, Deformable part models are convolutional neural networks, arXiv preprint arXiv:1409.5403.
[214] C. Galleguillos, S. Belongie, Context based object categorization: A critical survey, Computer Vision and Image Understanding (CVIU) 114 (2010) 712–722.
[215] M. Yang, Y. Wu, G. Hua, Context-aware visual tracking, IEEE Trans. on PAMI 31 (7) (2009) 1195–1209.
[216] H. Kruppa, M. C. Santana, B. Schiele, Fast and robust face finding via local context, in: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), 2003.
[217] Y. Dupuis, X. Savatier, J.-Y. Ertaud, P. Vasseur, Robust radial face detection for omnidirectional vision, Image Processing, IEEE Transactions on 22 (5) (2013) 1808–1821.
[218] K. Scherbaum, J. Petterson, R. S. Feris, V. Blanz, H.-P. Seidel, Fast face detector training using tailored views, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 2848–2855.
[219] C. Huang, H. Ai, T. Yamashita, S. Lao, M. Kawade, Incremental learning of boosted face detector, in: Proc. of ICCV, 2007.
[220] C. Zhang, R. Hamid, Z. Zhang, Taylor expansion based classifier adaptation: Application to person detection, in: Proc. of CVPR, 2008.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Figure 1. Examples of face images. Note the huge variations in pose, facial expression, lighting conditions, etc.

ACCEPTED MANUSCRIPT

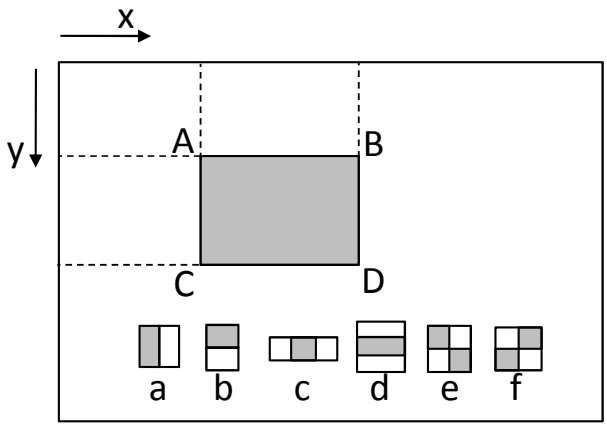


Figure 2. Illustration of the integral image and Haar-like rectangle features (a-f).

ACCEPTED MANUSCRIPT

Input

- Training examples $\mathcal{S} = \{(x_i, z_i), i = 1, \cdots, N\}$.
- T is the total number of weak classifiers to be trained.

Initialize

- Initialize example score $F^0(x_i) = \frac{1}{2} \ln\left(\frac{N_+}{N_-}\right)$, where N_+ and N_- are the number of positive and negative examples in the training data set.

Adaboost Learning

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

1. For each Haar-like feature $h(x)$ in the pool, find the optimal threshold H and confidence score c_1 and c_2 to minimize the Z score L^t (8).
2. Select the best feature with the minimum L^t.
3. Update $F^t(x_i) = F^{t-1}(x_i) + f_t(x_i), i = 1, \cdots, N$,
4. Update $W_{+1j}, W_{-1j}, j = 1, 2$.

Output Final classifier $F^T(x)$.

Figure 3. Adaboost learning pseudo code.

ACCEPTED MANUSCRIPT

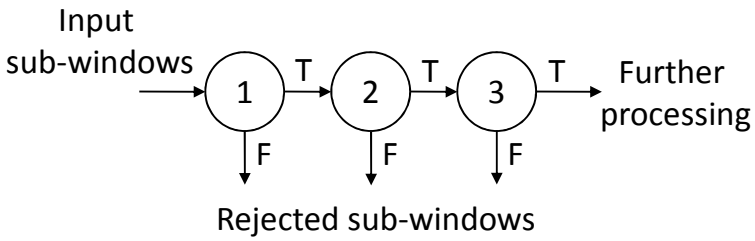


Figure 4. The attentional cascade.

ACCEPTED MANUSCRIPT

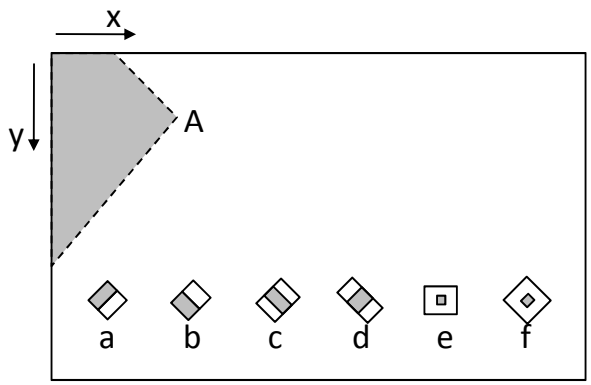


Figure 5. The rotated integral image/summed area table.

ACCEPTED MANUSCRIPT

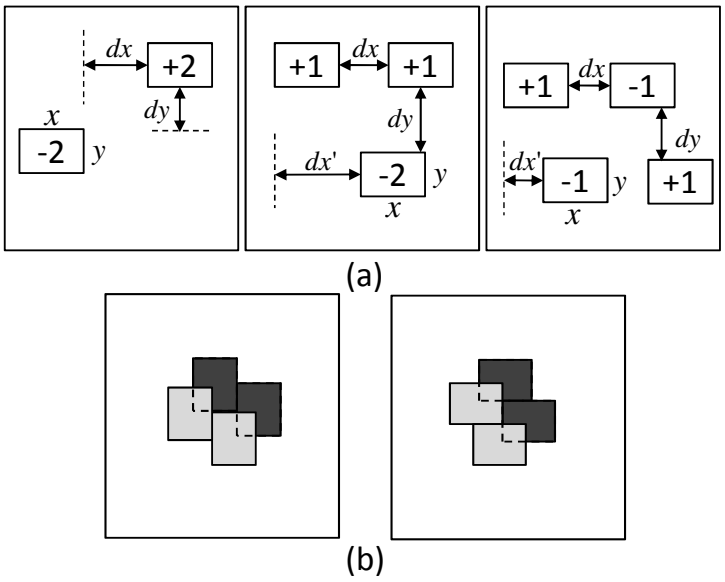


Figure 6. (a) Rectangular features with flexible sizes and distances introduced in [59]. (b) Diagonal filters in [64].

ACCEPTED MANUSCRIPT

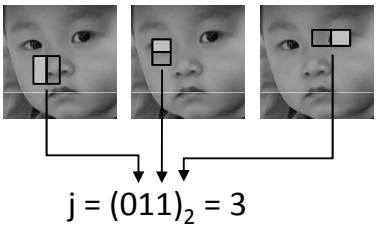


Figure 7. The joint Haar-like feature introduced in [62].

ACCEPTED MANUSCRIPT

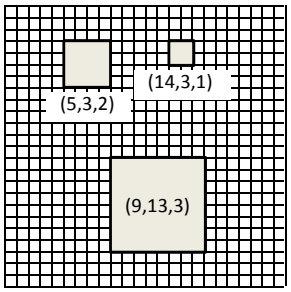


Figure 8. The sparse feature set in granular space introduced in [90].

ACCEPTED MANUSCRIPT

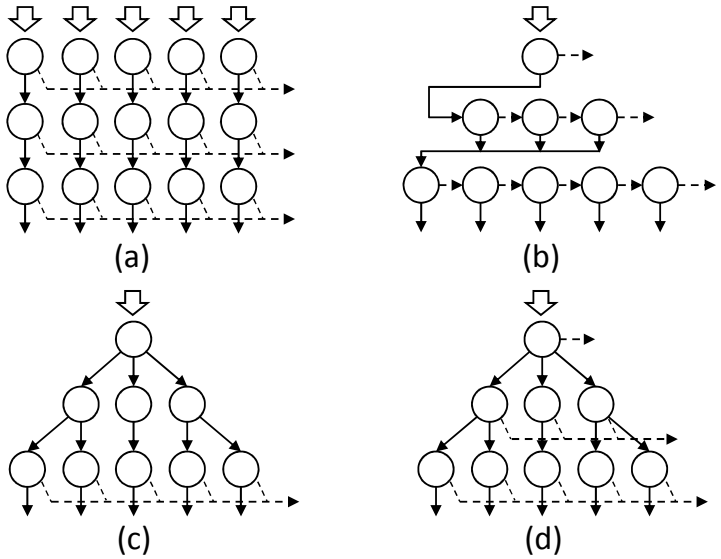


Figure 9. Various detector structures for multiview face detection. Each circle represents a strong classifier. The solid arrows are pass route, and the dashed arrows are reject route. (a) Parallel cascade [61]. (b) Detector-pyramid [59]. (c) Decision tree I [64]. (d) Decision tree II [130, 67, 131]. Note in (d) the early nodes are all able to perform rejection in order to speed up the detection. In addition, in [67, 131] the selection of the pass route for a branching node is non-exclusive.

ACCEPTED MANUSCRIPT

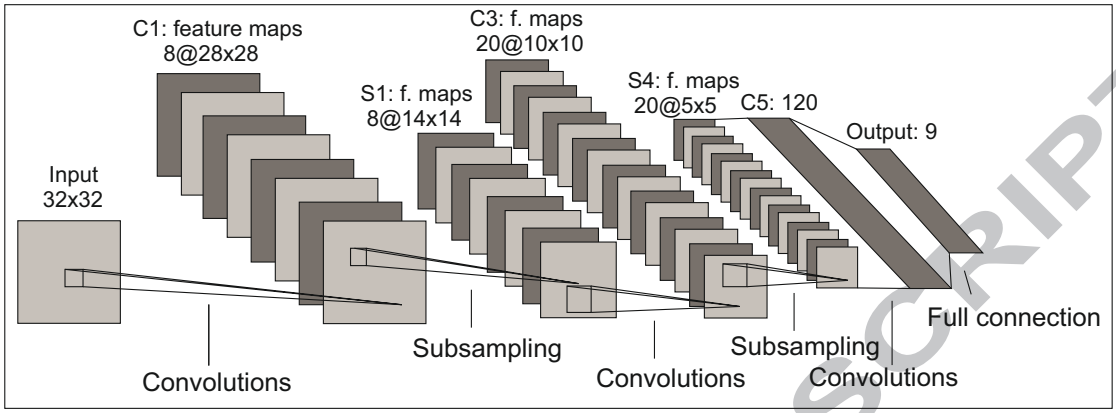
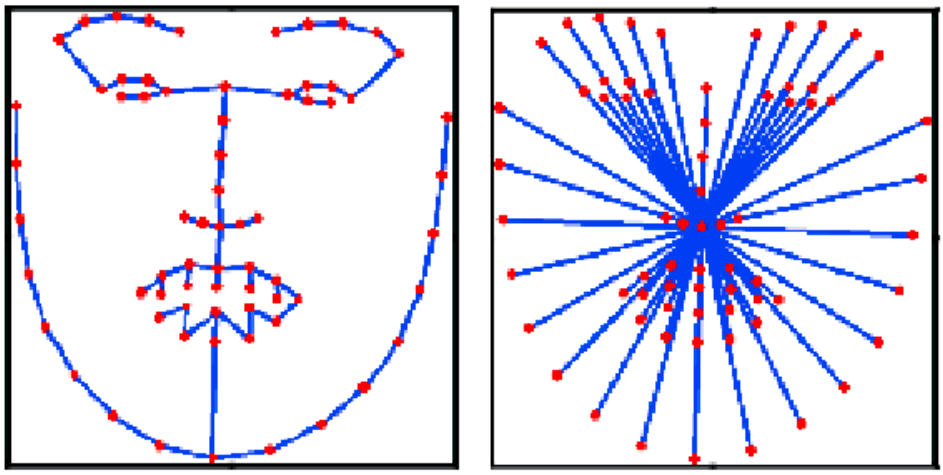
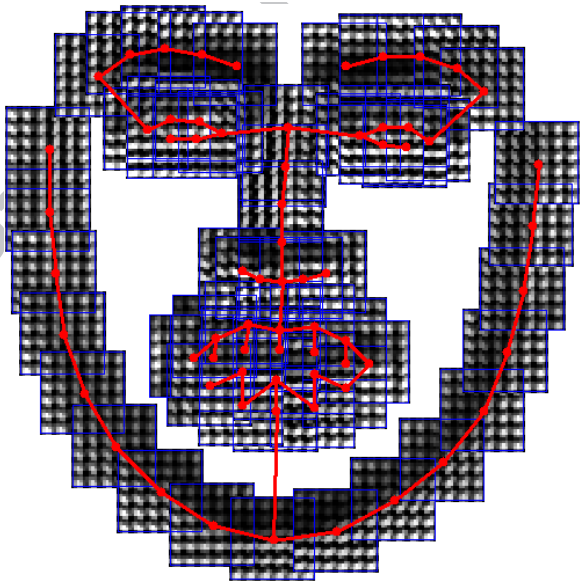


Figure 10. The deep CNN network for joint face detection and pose estimation.

ACCEPTED MANUSCRIPT



(a) (b)



(c)

Figure 11. Pictorial structures for faces. (a) Pictorial structure using a Minimum Spanning Tree (MST) approach, (b) Star-based Pictorial Structure and (c) filters per part for an MST tree component of a mixture.

ACCEPTED MANUSCRIPT

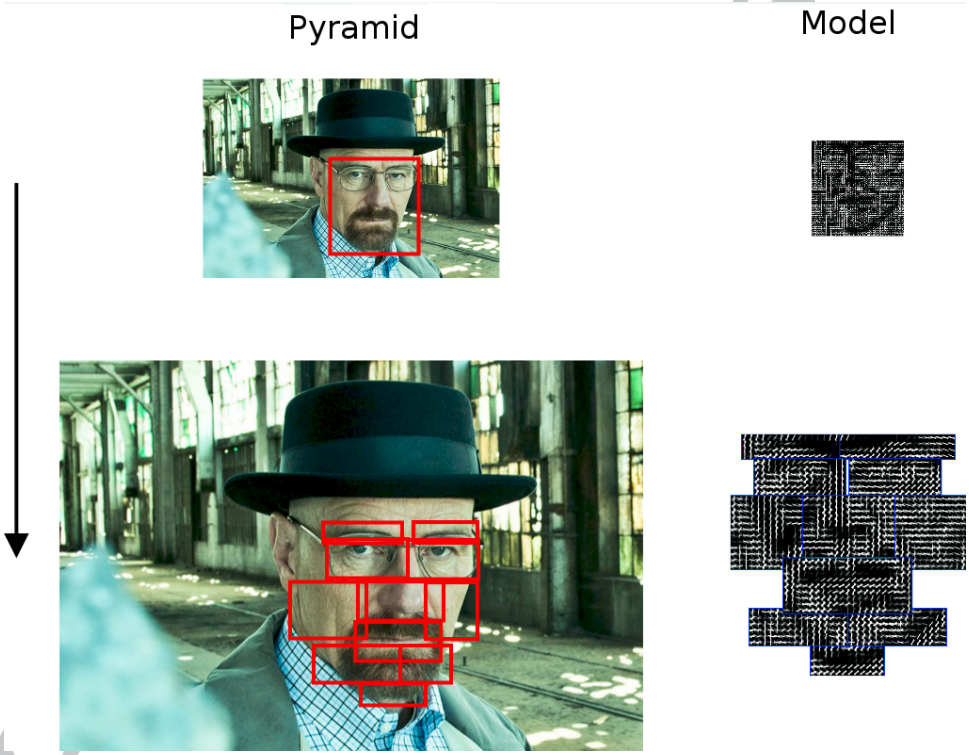
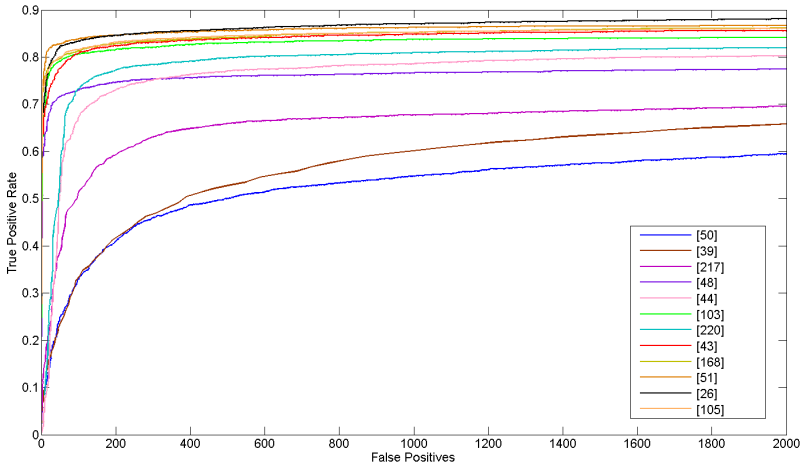
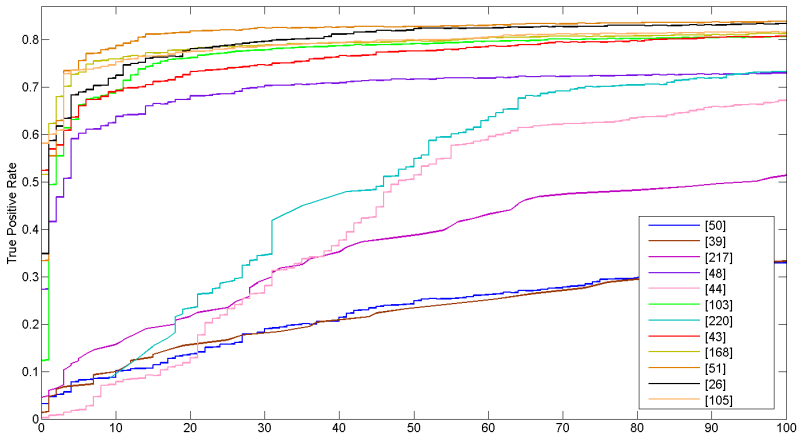


Figure 12. In the upper level of the pyramid we learn a rigid filter representing the whole face. In the second level filters are learned per part.

ACCEPTED MANUSCRIPT



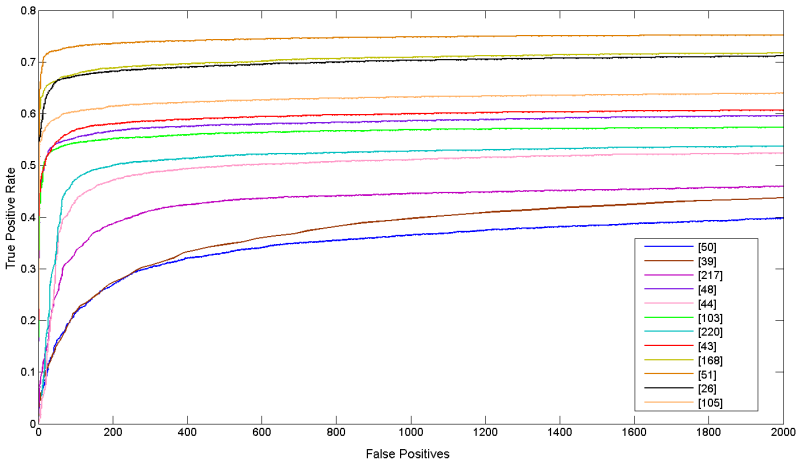
(a)



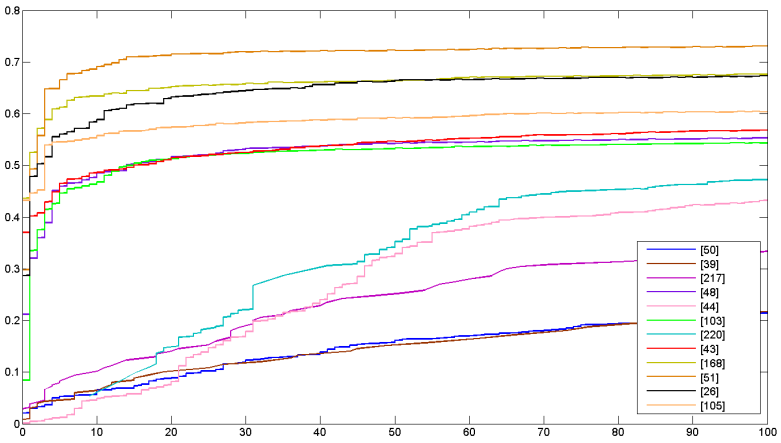
(b)

Figure 13. ROC curve, (true positive rate vs number of false positives) using the discrete degree of match for (a) allowing up to 2000 false positives and (b) close up in the region of small number of false positives.

ACCEPTED MANUSCRIPT



(a)



(b)

Figure 14. ROC curve, (true positives rate vs number of false positives) using the continuous degree of match for (a) allowing up to 2000 false positives and (b) close up in the region of small number of false positives.

ACCEPTED MANUSCRIPT

Highlights

- We present a comprehensive and survey for face detection ‘in-the-wild.
- We critically describe the advances in the three main families of algorithms.
- We comment on the performance of the state-of-the-art in the current benchmarks.
- We outline future research avenues on the topic and beyond.

ACCEPTED MANUSCRIPT