A Survey of Recent Advances in Face Detection

Cha Zhang and Zhengyou Zhang

June 2010

Technical Report
MSR-TR-2010-66

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Abstract

Face detection has been one of the most studied topics in the computer vision literature. In this technical report, we survey the recent advances in face detection for the past decade. The seminal Viola-Jones face detector is first reviewed. We then survey the various techniques according to how they extract features and what learning algorithms are adopted. It is our hope that by reviewing the many existing algorithms, we will see even better algorithms developed to solve this fundamental computer vision problem. [1]

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

1. Introduction

With the rapid increase of computational powers and availability of modern sensing, analysis and rendering equipment and technologies, computers are becoming more and more intelligent. Many research projects and commercial products have demonstrated the capability for a computer to interact with human in a natural way by looking at people through cameras, listening to people through microphones, understanding these inputs, and reacting to people in a friendly manner.

One of the fundamental techniques that enables such natural human-computer interaction (HCI) is face detection. Face detection is the step stone to all facial analysis algorithms, including face alignment, face modeling, face relighting, face recognition, face verification/authentication, head pose tracking, facial expression tracking/recognition, gender/age recognition, and many many more. Only when computers can understand face well will they begin to truly understand people's thoughts and intentions.

Given an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face [112]. While this appears as a trivial task for human beings, it is a very challenging task for computers, and has been one of the top studied research topics in the past few decades. The difficulty associated with face detection can be attributed to many variations in scale, location, orientation (in-plane rotation), pose (out-of-plane rotation), facial expression, lighting conditions, occlusions, etc, as seen in Fig. 1.

There have been hundreds of reported approaches to face detection. Early Works (before year 2000) had been nicely surveyed in [112] and [30]. For instance, Yang et al. [112] 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 to determine a face based on human knowledge; feature invariant approaches aim to find face structure features that are robust to pose and lighting variations; template matching methods use pre-stored face templates to judge if an image is a face; appearance-based methods learn face models from a set of representative training face images to perform detection. In general, appearance-based methods had been showing superior performance to the others, thanks to the rapid growing computation power and data storage.

The field of face detection has made significant progress in the past decade. In particular, the seminal work by Viola and Jones [92] has made face detection practically feasible in real world applications such as digital cameras and photo organization software. In this report, we present a brief survey on the latest development in face detection techniques since the publication of [112]. More attention will be given to boosting-based face detection schemes, which have evolved as the de-facto standard of face detection in real-world applications since [92].

The rest of the paper is organized as follows. Section 2 gives an overview of the Viola-Jones face detector, which also motivates many of the recent advances in face detection. Solutions to two key issues for face detection: what features to extract, and which learning algorithm to apply, will be surveyed in Section 3 (feature extraction), Section 4 (boosting learning algorithms) and Section 5 (other learning algorithms). Conclusions and future work are given in Section 6.

2. The Viola-Jones Face Detector

If one were asked to name a single face detection algorithm that has the most impact in the 2000's, it will most likely be the seminal work by Viola and Jones [92]. The Viola-Jones face detector contains three main ideas that make it possible to build a successful face detector that can run *in real time*: the integral image, classifier learning with AdaBoost, and the attentional cascade structure.

[1] This technical report is extracted from an early draft of the book "Boosting-Based Face Detection and Adaptation" by Cha Zhang and Zhengyou Zhang, Morgan & Claypool Publishers, 2010.

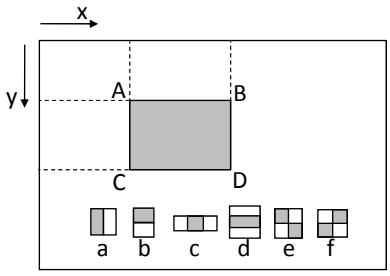

Figure 2. Illustration of the integral image and Haar-like rectangle features (a-f).

2.1. The Integral Image

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 by Crow [12] for use in mipmaps. Viola and Jones applied the integral image for rapid computation of Haar-like features, as detailed below.

The integral image is constructed as follows:

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

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 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 feature (a), the feature value is the difference in average pixel value 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) requires eight array references, and the four rectangle features (e and f) requires nine array references.

2.2. 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 [59] and [19].

The AdaBoost (Adaptive Boosting) algorithm is generally considered as the first step towards more practical boosting algorithms [17, 18]. In this section, following [80] and [19], we briefly present a generalized version of AdaBoost algorithm, usually referred as *RealBoost*. It has been advocated in various works [46, 6, 101, 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 $\text{sign}(\cdot)$ is the sign function. From the statistical view of boosting [19], 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 can be considered as 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, $h(x)$ is the Haar-like features computed with integral image, as was shown in Fig. 2 (a-f). A decision threshold H divide the output of $h(x)$ into two subregions, u_1 and u_2, $u_1 \cup u_2 = \mathbb{R}$. The base function $f(x)$ is thus:

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

which is often referred 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

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

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:

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

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.

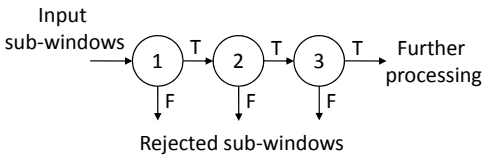

Figure 4. The attentional cascade.

Eq. (8) is referred as the Z score in [80]. 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.

2.3. The Attentional Cascade Structure

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 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 forms a degenerate decision tree, which was called a "cascade" in [92]. As shown in Fig. 4, the input sub-windows pass a series of nodes during detection. Each node will make a binary decision whether the window will be kept for the next round or rejected immediately. The number of weak classifiers in the nodes usually increases as the number of nodes a sub-window passes. For instance, in [92], 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, Viola and Jones [92] 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 4.

In [92], 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 specified manually. 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 hurt. On the other hand, if the decision thresholds were 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 early 2000's, it is no wonder that training a good face detector can take months of fine-tuning.

3. 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, Lienhart and Maydt[49] generalized the feature set of [92] by introducing 45 degree

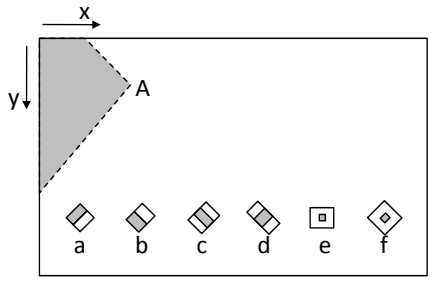

Figure 5. The rotated integral image/summed area table.

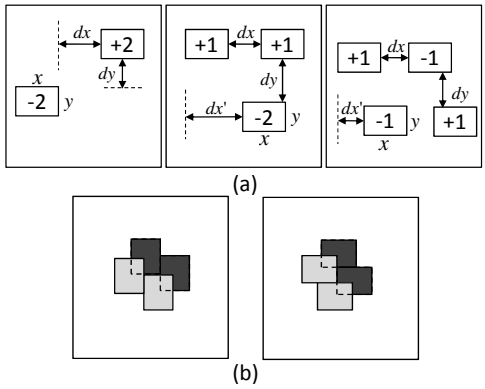

Figure 6. (a) Rectangular features with flexible sizes and distances introduced in [46]. (b) Diagonal filters in [38].

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:

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

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 [92] for multi-view face detection, and proposed to extend the feature set by allowing more flexible combination of rectangular regions. For instance, in [46], 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 they are at certain distances of (dx, dy) apart. The authors argued that these features can be non-symmetrical to cater to non-symmetrical characteristics of non-frontal faces. Jones and Viola [38] 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.

Jones et al. [39] further extended the Haar-like feature set 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, Jones et al. added single box rectangular sum features, and new features across two images. For instance:

$$f_i = r_i(\Delta) - r_i(S), \quad (10)$$

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

One must be careful 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 [39].

Mita et al. [62] proposed joint Haar-like features, which is based on co-occurrence of multiple Haar-like features. The authors 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 [8]. It was shown that CART tree based weak classifiers improved results across various boosting algorithms with a small loss

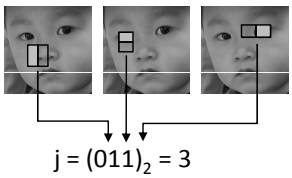

Figure 7. The joint Haar-like feature introduced in [62].

in speed. In another variation for improving the weak classifier, [101] proposed to use a single Haar-like feature, and equally bin the feature values into a histogram to be used in a RealBoost learning algorithm. 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. [101] proposed to use 64 bins. And in their later work [32], they specifically pointed out that too fine granularity of the histogram may cause overfitting, and suggested to use fine granularity in the first few layers of the cascade, and coarse granularity in latter layers. Another interesting recent work is [107], where the authors proposed a new weak classifier called Bayesian stump. 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 that the Haar features are usually normalized by the test windows' intensity covariance [92]. In [21] a modified census transform was adopted to generate illumination-insensitive features for face detection. On each pixel's 3×3 neighborhood, the authors 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) [65], which have been very effective for face recognition tasks [2, 117]. In [37, 119], LBP was applied for face detection tasks under a Bayesian and a boosting framework, respectively. More recently, inspired by LBP, Yan et al. [110] proposed locally assembled binary feature, which showed great performance on standard face detection data sets.

To explore possibilities to further improve performance, more and more complex features were proposed in the literature. For instance, Liu and Shum [52] studied generic linear features, which is defined by a 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 [60]. In [10], the linear features were constructed by prelearning them using local non-negative matrix factorization (LNMF), which is still sub-optimal. Instead, Liu and Shum [52] 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 [97], 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 [52] and [97] 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 are generally much higher than the traditional Haar-like features. Oppositely, Baluja et al. [4] proposed to use simple pixel pairs as features, and Abramson and Steux [1] proposed to use the relative values of a set of control points as features. Such pixel-based feature 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. Levi and Weiss [45] proposed local edge orientation histograms, which computes the histogram of edges 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, Dalal and Triggs [13] proposed a similar scheme called histogram of oriented gradients (HoG), which became a very popular feature for human/pedestrian detection [120, 25, 88, 43, 15]. In [99], the authors proposed spectral histogram features, which adopts a broader set of filters before collecting the histogram features, including gradient filters, Laplacian of Gaussian filters and Gabor filters. Compared with [45], the histogram features in [99] were based on the whole testing window rather than local regions, and support vector machines (SVMs) were used for classification. Zhang et al. [118] proposed another histogram-based feature called spatial histograms, which is based on local statistics of LBP. HoG and LBP were also combined in [98], which achieved excellent performance on human detection with partial occlusion handling. Region

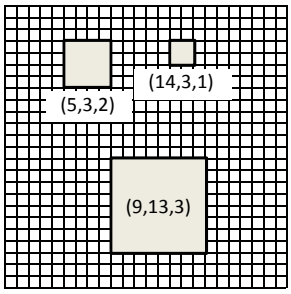

Figure 8. The sparse feature set in granular space introduced in [33].

covariance was another statistics based feature, proposed by Tuzel et al. [91] for generic object detection and texture classification tasks. Instead of using histograms, they compute the covariance matrices among the color channels and gradient images. Regional covariance features can also be efficiently computed using integral images.

Huang et al. [33] proposed a sparse feature set in order to strengthen the features' discrimination power without incurring too much additional computational cost. Each sparse feature can be represented as:

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

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 [33], 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 authors proposed 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 a small set of training examples to evaluate all features first and rejects those that are unlikely to be good. The performance of the multi-view face detector trained in [33] using sparse features was very good.

As new features are composed in seeking the best discrimination power, the feature pool becomes larger and larger, which creates new challenges in the feature selection process. A number of recent works have attempted to address this issue. For instance, [113] proposed to discover compositional features using the classic frequent item-set mining scheme in data mining. Instead of using the raw feature values, they assume a collection of induced binary features (e.g., decision stumps with known thresholds) 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 [26], 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 [53], 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 [25], 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, Opelt et al. [66] composed multiple boundary fragments to weak classifiers and formed a strong "boundary-fragment-model" detector using boosting. They ensure the feasibility of the feature selection process by limiting the number of boundary fragments to 2-3 for each weak classifier. Shotton et al. [86] learned their object detectors with a boosting algorithm and their feature set consisted of a randomly chosen dictionary of contour fragments. A very similar edgelet feature was proposed in [102], and was used to learn human body part detectors in order to handle multiple, partially occluded humans. In [79], 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

Table 1. Features for face/object detection.

Feature Type	Representative Works
Haar-like features and its variations	Haar-like features [92]
	Rotated Haar-like features [49]
	Rectangular features with structure [46, 38]
	Haar-like features on motion filtered image [39]
Pixel-based features	Pixel pairs [4]
	Control point set [1]
Binarized features	Modified census transform [21]
	LBP features [37, 119]
	Locally assembled binary feature [110]
Generic linear features	Anisotropic Gaussian filters [60]
	LNMF [10]
	Generic linear features with KL boosting [52]
	RNDA [97]
Statistics-based features	Edge orientation histograms [45, 13] etc.
	Spectral histogram [99]
	Spatial histogram (LBP-based) [118]
	HoG and LBP [98]
	Region covariance [91]
Composite features	Joint Haar-like features [62]
	Sparse feature set [33]
Shape features	Boundary/contour fragments [66, 86]
	Edgelet [102]
	Shapelet [79]

jointly, such as the work in [104] and [23].

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

4. 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 [92], the standard AdaBoost algorithm [17] was adopted. In a number of follow-up works [46, 6, 101, 62], researchers advocated the use of RealBoost, which was explained in detail in Section 2.2. Both Lienhart et al. [48] and Brubaker et al. [8] compared three boosting algorithms: AdaBoost, RealBoost and GentleBoost, though they reach different conclusions as the former recommended GentleBoost while the latter showed 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 [46], the authors proposed FloatBoost, 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 [73] into AdaBoost, which not only add features during training, but also backtrack and examine 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. Jang and Kim [36] proposed to used evolutionary algorithms 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 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 2.3, Viola and Jones [92] 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 instead of starting from scratch at each new node. For instance, in [108], the authors proposed to use a "chain" structure 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 [101], the authors proposed the so-called nesting-structured cascade. Instead of taking the existing partial classifier as a prefix, they took the confidence output of the partial classifier and used it as a feature to build the first weak classifier. Both paper 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, Viola and Jones [93] 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. They proposed a new scheme called asymmetric AdaBoost, which artificially increase the weights on positive examples in each round of AdaBoost such that the error criterion biases towards having low false negative rates. In [71], the authors extended the above work and sought to balance the skewness of labels presented to each weak classifiers, so that they are trained more equally. Masnadi-Shirazi and Vasconcelos [55] further proposed a more rigorous form of asymmetric boosting based on the statistical in-

terpretation of boosting [19] with an extension of the boosting loss. Namely, the exponential cost criterion in Eq. (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. Masnadi-Shirazi and Vasconcelos [55] minimized the above criterion following the AnyBoost framework in [57]. They were able to build a detector with very high detection rate [56], though the performance of the detector deteriorates very quickly when the required false positive rate is low.

Wu et al. [105] 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) [14], except that only the covariance matrix of the positive feature projections are considered in LAC. In practice, their performance are also similar. Applying LAC or FDA on a set of features pre-selected by AdaBoost is equivalent to readjust the confidence values of the AdaBoost learning (Eq. (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 diminished. Wu et al. attributed the phenomenon 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 tradeoff 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, Viola and Jones [92] attempted to reject zero positive examples until this become 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. Lienhart et al. [48] 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 [87], the authors proposed to use a a ratio test to determine the rejection thresholds. Specifically, the authors viewed the cascade detector as a sequential decision-making problem. A sequential decision-making theory had been developed by Wald [95], which proved that the solution to minimizing the expected evaluation time for a sequential decision-making problem is the sequential probability ratio test. Sochman and Matas [87] abandoned the notion of nodes, and set rejection threshold after each weak classifier. They 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 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. Brubaker et al. [8] proposed another fully automatic algorithm for setting the intermediate thresholds during training. 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 [107], a dynamic cascade was proposed, which assumes that the false negative rate of the nodes changes exponentially in each stage, following the idea in [7]. 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, Luo [54] 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. Bourdev and Brandt [7] instead proposed a heuris-

tic approach to use a parameterized exponential curve to set the intermediate nodes' detection targets, called a "rejection distribution vector". 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. Zhang and Viola [115] proposed a more principled data-driven scheme for setting intermediate thresholds named multiple instance pruning. 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 [7] 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 [7], the authors proposed a scheme 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 [107] and [115], the authors proposed to use importance sampling to help address the large data set issue. The training positive or negative data set are resampled every once a while to ensure feasible computation. Both work 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 process. For instance, McCane and Novins [58] proposed a discrete downhill search scheme to limit the number of features compared during feature selection. Such a greedy search strategy offered a 300–400 fold speed up in training, though the false positive rate of the resultant detector increased by almost a factor of 2. Brubaker et al. [8] studied various filter schemes to reduce the size of the feature pool, and showed that randomly selecting a subset of features at each iteration for feature selection appears to work reasonably well. Wu et al. [106] proposed a cascade learning algorithm based on forward feature selection [100], which is 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. Pham and Cham [70] presented another fast method to train and select Haar features. It 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 [106] and [70] 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 [101, 7].

Various efforts have also been made to improve the detector's test speed. For instance, in the sparse feature set in [33], the authors 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 [72] was often adopted for faster detection. Schneiderman [81] 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 [110] to speed up their locally assembled binary feature based detector. In [69], the authors proposed a scheme to improve the detection speed on quasi-repetitive inputs, such as the video input during videoconferencing. The idea is to cache a set of image exemplars, each induces its own discriminant subspace. Given a new video frame, the algorithm quickly searches through the exemplar database indexed with an online version of tree-structured vector quantization, S-tree [9]. 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 achieve through selective attention, such as those based on motion, skin color, background modeling 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.

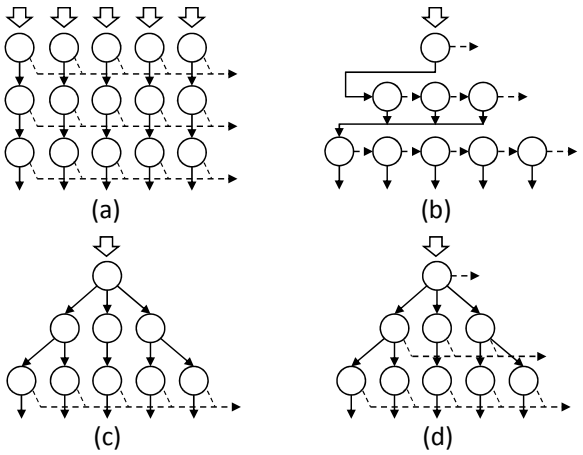

Figure 9. Various detector structures for multiview face detection. Each circle represents a strong classifier. The solid arrows are pass route, and the dashed arrows are reject route. (a) Parallel cascade [101]. (b) Detector-pyramid [46]. (c) Decision tree I [38]. (d) Decision tree II [20, 32, 50]. Note in (d) the early nodes are all able to perform rejection in order to speed up the detection. In addition, in [32, 50] the selection of the pass route for a branching node is non-exclusive.

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, by Wu et al. [101]. 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. Li et al. [46] used a pyramid structure to handle the task, as shown in Fig. 9(b). 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 to improve.

Fig. 9(c) and (d) showed two decision tree structures for multiview face detection. In [38], 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. Fröba and Ernst [20] had a similar tree structure (Fig. 9(d)) 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. Huang et al. [32] and Lin and Liu [50] independently proposed a very similar solution to this issue, which were named vector boosting and multiclass Bhattacharyya boost (MBHBoost), respectively. The idea is to have 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 are fused from the vector output. Such a soft branching scheme can greatly reduce the risk of misclassification during testing. Another interesting idea in [32, 50] was to have all the subcategory classifiers 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 trained jointly [89].

Vector boosting and MBHBoost solved the issue of misclassification in pose estimation during testing. During training, they still used faces manually labeled with pose information to learn the multiview detector. However, for certain object classes such as pedestrians or cars, an agreeable manual pose labeling scheme is often unavailable. Seemann et al. [84] extended the implicit shape model in [44] 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. Shan et al. [85] proposed an exemplar-based categorization scheme for multiview object detection. At each round of boosting learning, the algorithm not only selects a feature to construct a weak classifier, it also select for a set of exemplars to guide the learning to focus on different views of the object. Tu [90] proposed probabilistic boosting tree, which embedded clustering in the learning phase. 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 node. The performance of boosting tree on multiview object detection is uncertain due to the limited experimental results provided in the paper. In [103], 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 any more. For instance, Kim and Cipolla [41] proposed an algorithm called multiple classifier boosting, which is a straightforward extension of the multiple instance boosting approach in Viola et al. [94]. 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. Babenko et al. [3] independently developed a very similar scheme they called multi-pose learning, and further combined it 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 [89]. Zhang and Zhang [116] proposed an algorithm called winner-take-all multiple category boosting (WTA-McBoost), which is more suitable for learning multiview detectors with huge amount of training data. Instead of using AnyBoost [57], WTA-McBoost is derived from confidence rated AdaBoost [80], which is much more efficient to train, and easy to support feature sharing.

To summarize this Section, we make a list of the challenges and approaches to address them in Table 2.

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

Challenges	**Representative Works**
General boosting schemes	AdaBoost [92]
	RealBoost [46, 6, 101, 62]
	GentleBoost [48, 8]
	FloatBoost [46]
Reuse previous nodes' results	Boosting chain [108]
	Nested cascade [101]
Introduce asymmetry	Asymmetric Boosting [93, 71, 55]
	Linear asymmetric classifier [105]
Set intermediate thresholds during training	Fixed node performance [48]
	WaldBoost [87]
	Based on validation data [8]
	Exponential curve [107]
Set intermediate thresholds after training	Greedy search [54]
	Soft cascade [7]
	Multiple instance pruning [115]
Speed up training	Greedy search in feature space [58]
	Random feature subset [8]
	Forward feature selection [106]
	Use feature statistics [70]
Speed up testing	Reduce number of weak classifiers [46, 36]
	Feature centric evaluation [81, 110]
	Caching/selective attention [69] etc.
Multiview face detection	Parallel cascade [101]
	Pyramid structure [46]
	Decision tree [38, 20]
	Vector valued boosting [32, 50]
Learn without subcategory labels	Cluster and then train [84]
	Exemplar-based learning [85]
	Probabilistic boosting tree [90]
	Cluster with selected features [103]
	Multiple classifier/category boosting [41, 3, 116]

5. Other learning schemes

As reviewed in the previous section, the seminal work by Viola and Jones [92] has inspired a lot of research applying the boosting cascade for face detection. Nevertheless, there were still a few papers approaching the problem in different ways, some providing very competitive performances. Again, we will only focus on works not covered in [112].

Keren et al. [40] proposed Antifaces, a multi-template scheme for detecting arbitrary objects including faces in images. The core idea is very similar to the cascade structure in [92], which uses a set of sequential classifiers to detect faces and rejects non-faces fast. Each classifier, referred as a "detector" in [40], 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 modeled by a Boltzmann distribution and assumed to be smooth, thus none is needed during template construction.

Liu [51] presented a Bayesian discriminating features method for frontal face detection. The face class was modeled as a multivariate normal distribution. A subset of the nonfaces that lie closest to the face class was then selected based on the face class model and also modeled with a multivariate normal distribution. The final face/nonface decision was made by a Bayesian classifier. Since only the nonfaces closest to the face class were modeled, the majority of the nonfaces were ignored during the classification. This was inspired by the concept of support vector machines

(SVMs) [11], where only a subset of the training examples (the support vectors) were used to define the final classifier.

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 successful approach for face detection [68, 27]. However, the speed of SVM based face detectors was generally slow. Various schemes have since been proposed to speed up the process. For instance, Romdhani et al. [75] proposed to compute a set of reduced set vectors from the original support vectors. These reduced set vectors are then tested against the test example sequentially, making early rejections possible. Later, Rätsch et al. [74] further improved the speed by approximating the reduced set vectors with rectangle groups, which gained another 6-fold speedup. Heisele et al. [29] instead used a hierarchy of SVM classifiers with different resolutions in order to speed up the overall system. The early classifiers are 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. Li et al. [47] proposed a multiview face detector similar to the approach in [78, 38]. They first constructed a face pose estimator using support vector regression (SVR), then trained separate face detectors for each face pose. Yan et al. [109] instead executed multiple SVMs first, and then applied 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 causing misclassification. Wang and Ji [96] remarked that in the real world the face poses may vary greatly and many SVMs are needed. They proposed an approach to combine cascade and bagging for multiview face detection. 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. Hotta [31] used a single SVM for multiview face detection, and relied on the combination of local and global kernels for better performance. No experimental results were given in [96, 31] 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.

Neural networks were another popular approach to build a face detector. Early representative methods included the detectors by Rowley et al. [77] and Roth et al. [76]. Féraud et al. [16] proposed an approach based on a neural network model called the constrained generative model (CGM). CGM is an autoassociative, 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. Multiview face detection was achieved by measuring the reconstruction errors of multiple CGMs, combined via a conditional mixture and an MLP gate network. In [24], 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 analyze. Osadchy et al. [67] proposed another convolutional network based approach, which was able to perform multiview face detection and facial pose estimation simultaneously. The idea is to train a convolutional neural network to map face images to points on a low dimensional face manifold parameterized by facial pose, and non-face images to points far away from the manifold. The detector was fast and achieved impressive performance – on par with the boosting based detectors such as [38].

Schneiderman and Kanade [83] described an object detector based on detecting localized parts of the object. 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, Schneiderman [82] further examined the cases where the statistical dependency cannot be easily decomposed into separate parts. He proposed a method to learn the dependency structure of a Bayesian network based classifier. Although the problem is known to be NP complete, he presented a scheme that selects a structure by seeking to optimize a sequence of two cost functions: the local modeling error using the likelihood ratio test as before, and the global empirical classification error computed on a cross-validation set of images. The commercial PittPatt face detection software that combines the above approach with the feature-centric cascade detection scheme in [81] showed state-of-the-art performance on public evaluation tests [64].

Schneiderman and Kanade [83] used wavelet variables to represent parts of the faces, which do not necessarily corresponds to semantic components. In the literature, there had been many component-based object detectors that relied on semantically meaningful component detectors [112, 63, 5]. In the recent work by Heisele et al. [28], the authors used 100 textured 3D head models to train 14 component detectors. 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 com-

Table 3. Other schemes for face/object detection (since [112]).

General Approach	Representative Works
Template matching	Antiface [40]
Bayesian	Bayesian discriminating features [52]
SVM – speed up	Reduced set vectors and approximation [75, 74]
	Resolution based SVM cascade [29]
SVM – multi-view face detection	SVR based pose estimator [47]
	SVR fusion of multiple SVMs [109]
	Cascade and bagging [96]
	Local and global kernels [31]
Neural networks	Constrained generative model [16]
	Convolutional neural network [24, 67]
Part-based approaches	Wavelet localized parts [83, 82]
	SVM component detectors adaptively trained [28]
	Overlapping part detectors [61]

ponent 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, Mikolajczyk et al. [61] applied 7 detectors to find body parts including frontal and profile faces, frontal and profile heads, frontal and profile upper body, and legs. A joint likelihood body model is then adopted to build a body structure by starting with one part and adding the confidence provided by other body part detectors.

Once again we summarize the approaches in the section in Table 3.

6. Conclusions and Future Work

In this paper, we surveyed some of the recent advances in face detection. 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 softwares 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, as was pointed in a recent technical report by Jain and Learned-Miller [35], face detection in completely unconstrained settings remains a very challenging task, particularly due to the significant pose and lighting variations. In our in-house tests, the state-of-the-art face detectors can achieve about 50-70% detection rate, with about 0.5-3% of the detected faces being false positives. Consequently, we believe there are still a lot of works that can be done to further improve the performance.

The most straightforward future direction is to further improve the learning algorithm and features. The Haar features used in the work by Viola and Jones [92] are very simple and effective for frontal face detection, but they are less ideal for faces at arbitrary poses. Complex features may increase the computational complexity, though they can be used in the form of a post-filter and still be efficient, which may significantly improve the detector's performance. Regarding learning, the boosting learning scheme is great if all the features can be pre-specified. However, other learning algorithms such as SVM or convolutional neural networks can often perform equally well, with built-in mechanisms for new feature generation.

The modern face detectors are mostly appearance-based methods, which means that they need training data to learn the 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 be very ideal to reduce the amount of work needed for data collection.

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 [22] and visual tracking [111]. One scheme of using local context to improve face detection was also presented in [42], and we think that is a very promising direction to pursue.

In environments which have low variations, adaptation could bring very significant improvements to face detection. Unlike in 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 [34, 114], and we strongly believe that this is a great direction for future work.

References

[1] Y. Abramson and B. Steux. YEF* real-time object detection. In *International Workshop on Automatic Learning and Real-Time*, 2005. 5, 7

[2] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *Proc. of ECCV*, 2004. 5

[3] B. Babenko, P. Dollár, Z. Tu, and 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. 11

[4] S. Baluja, M. Sahami, and H. A. Rowley. Efficient face orientation discrimination. In *Proc. of ICIP*, 2004. 5, 7

[5] S. M. Bileschi and B. Heisele. Advances in component-based face detection. In *Pattern Recognition with Support Vector Machines Workshop*, 2002. 12

[6] C. Bishop and P. Viola. Learning and vision: Discriminative methods. In *ICCV Course on Learning and Vision*, 2003. 2, 7, 11

[7] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *Proc. of CVPR*, 2005. 8, 9, 11

[8] S. C. Brubaker, J. Wu, J. Sun, M. D. Mullin, and J. M. Rehg. On the design of cascades of boosted ensembles for face detection. Technical report, Georgia Institute of Technology, GIT-GVU-05-28, 2005. 4, 7, 8, 9, 11

[9] M. M. Campos and G. A. Carpenter. S-tree: Self-organizing trees for data clustering and online vector quantization. *Neural Networks*, 14(4–5):505–525, 2001. 9

[10] X. Chen, L. Gu, S. Z. Li, and H.-J. Zhang. Learning representative local features for face detection. In *Proc. of CVPR*, 2001. 5, 7

[11] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*. Cambridge University Press, 2000. 12

[12] F. Crow. Summed-area tables for texture mapping. In *Proc. of SIGGRAPH*, volume 18, pages 207–212, 1984. 2

[13] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. of CVPR*, 2005. 5, 7

[14] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons Inc., 2 edition, 2001. 8

[15] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Trans. on PAMI*, 31(12):2179–2195, 2009. 5

[16] R. Féraud, O. J. Bernier, J.-E. Viallet, and M. Collobert. A fast and accurate face detector based on neural networks. *IEEE Trans. on PAMI*, 23(1):42–53, 2001. 12, 13

[17] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conf. on Computational Learning Theory*, 1994. 2, 7

[18] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. 2

[19] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics, Stanford University, 1998. 2, 8

[20] B. Fröba and A. Ernst. Fast frontal-view face detection using a multi-path decision tree. In *Proc. of Audio- and Video-based Biometric Person Authentication*, 2003. 10, 11

[21] B. Fröba and A. Ernst. Face detection with the modified census transform. In *IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, 2004. 5, 7

[22] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding (CVIU)*, 114:712–722, 2010. 13

[23] W. Gao, H. Ai, and S. Lao. Adaptive contour features in oriented granular space for human detection and segmentation. In *Proc. of CVPR*, 2009. 7

[24] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Trans. on PAMI*, 26(11):1408–1423, 2004. 12, 13

[25] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. of CVPR*, 2006. 5, 6

[26] F. Han, Y. Shan, H. S. Sawhney, and R. Kumar. Discovering class specific composite features through discriminative sampling with Swendsen-Wang cut. In *Proc. of CVPR*, 2008. 6

[27] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. Technical report, Center for Biological and Computational Learning, MIT, A.I. Memo 1687, 2000. 12

[28] B. Heisele, T. Serre, and T. Poggio. A component-based framework for face detection and identification. *International Journal of Computer Vision*, 74(2):167–181, 2007. 12, 13

[29] B. Heisele, T. Serre, S. Prentice, and T. Poggio. Hierarchical classification and feature reduction for fast face detection with support vector machines. *Pattern Recognition*, 36:2007–2017, 2003. 12, 13

[30] E. Hjelmas and B. K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83:236–274, 2001. 1

[31] K. Hotta. View independent face detection based on combination of local and global kernels. In *International Conference on Computer Vision Systems*, 2007. 12, 13

[32] C. Huang, H. Ai, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. In *Proc. of ICCV*, 2005. 5, 10, 11

[33] C. Huang, H. Ai, Y. Li, and S. Lao. Learning sparse features in granular space for multi-view face detection. In *Intl. Conf. on Automatic Face and Gesture Recognition*, 2006. 6, 7, 9

[34] C. Huang, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Incremental learning of boosted face detector. In *Proc. of ICCV*, 2007. 13

[35] V. Jain and E. Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, University of Massachusetts, Amherst, 2010. 13

[36] J.-S. Jang and J.-H. Kim. Fast and robust face detection using evolutionary pruning. *IEEE Trans. on Evolutionary Computation*, 12(5):562–571, 2008. 7, 11

[37] H. Jin, Q. Liu, H. Lu, and X. Tong. Face detection using improved lbp under bayesian framework. In *Third Intl. Conf. on Image and Grahics (ICIG)*, 2004. 5, 7

[38] M. Jones and P. Viola. Fast multi-view face detection. Technical report, Mitsubishi Electric Research Laboratories, TR2003-96, 2003. 4, 7, 10, 11, 12

[39] M. Jones, P. Viola, and D. Snow. Detecting pedestrians using patterns of motion and appearance. Technical report, Mitsubishi Electric Research Laboratories, TR2003-90, 2003. 4, 7

[40] D. Keren, M. Osadchy, and C. Gotsman. Antifaces: A novel fast method for image detection. *IEEE Trans. on PAMI*, 23(7):747–761, 2001. 11, 13

[41] T.-K. Kim and R. Cipolla. MCBoost: Multiple classifier boosting for perceptual co-clustering of images and visual features. In *Proc. of NIPS*, 2008. 11

[42] H. Kruppa, M. C. Santana, and 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. 13

[43] I. Laptev. Improvements of object detection using boosted histograms. In *British Machine Vision Conference*, 2006. 5

[44] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. of CVPR*, 2005. 10

[45] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In *Proc. of CVPR*, 2004. 5, 7

[46] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. In *Proc. of ECCV*, 2002. 2, 4, 7, 10, 11

[47] Y. Li, S. Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *International Conference on Automatic Face and Gesture Recognition*, 2000. 12, 13

[48] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Technical report, Microprocessor Research Lab, Intel Labs, 2002. 7, 8, 11

[49] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *Proc. of ICIP*, 2002. 3, 7

[50] Y.-Y. Lin and T.-L. Liu. Robust face detection with multi-class boosting. In *Proc. of CVPR*, 2005. 10, 11

[51] C. Liu. A bayesian discriminating features method for face detection. *IEEE Trans. on PAMI*, 25(6):725–740, 2003. 11

[52] C. Liu and H.-Y. Shum. Kullback-Leibler boosting. In *Proc. Of CVPR*, 2003. 5, 7, 13

[53] X. Liu and T. Yu. Gradient feature selection for online boosting. In *Proc. of ICCV*, 2007. 6

[54] H. Luo. Optimization design of cascaded classifiers. In *Proc. of CVPR*, 2005. 8, 11

[55] H. Masnadi-Shirazi and N. Vasconcelos. Asymmetric boosting. In *Proc. of ICML*, 2007. 7, 8, 11

[56] H. Masnadi-Shirazi and N. Vasconcelos. High detection-rate cascades for real-time object detection. In *Proc. of ICCV*, 2007. 8

[57] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Proc. of NIPS*, 2000. 8, 11

[58] B. McCane and K. Novins. On training cascade face detectors. In *Image and Vision Computing*, 2003. 9, 11

[59] R. Meir and G. Rätsch. An introduction to boosting and leveraging. *S. Mendelson and A. J. Smola Ed., Advanced Lectures on Machine Learning, Springer-Verlag Berlin Heidelberg*, pages 118–183, 2003. 2

[60] J. Meynet, V. Popovici, and J.-P. Thiran. Face detection with boosted gaussian features. *Pattern Recognition*, 40(8):2283–2291, 2007. 5, 7

[61] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Proc. of ECCV*, 2004. 13

[62] T. Mita, T. Kaneko, and O. Hori. Joint Haar-like features for face detection. In *Proc. of ICCV*, 2005. 2, 4, 7, 11

[63] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. on PAMI*, 23(4):349–361, 2001. 12

[64] M. C. Nechyba, L. Brandy, and H. Schneiderman. Pittpatt face detection and tracking for the CLEAR 2007 evaluation. In *Classification of Events, Activities and Relations Evaluation and Workshop*, 2007. 12

[65] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on PAMI*, 24:971–987, 2002. 5

[66] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proc. of CVPR*, 2006. 6, 7

[67] M. Osadchy, M. L. Miller, and Y. L. Cun. Synergistic face detection and pose estimation with energy-based models. In *Proc. of NIPS*, 2004. 12, 13

[68] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proc. of CVPR*, 1997. 12

[69] M.-T. Pham and T.-J. Cham. Detection caching for faster object detection. In *Proc. of CVPR*, 2005. 9, 11

[70] M.-T. Pham and T.-J. Cham. Fast training and selection of haar features during statistics in boosting-based face detection. In *Proc. of ICCV*, 2007. 9, 11

[71] M.-T. Pham and T.-J. Cham. Online learning asymmetric boosted classifiers for object detection. In *Proc. of CVPR*, 2007. 7, 11

[72] F. Porikli. Integral histogram: A fastway to extract histograms in cartesian spaces. In *Proc. of CVPR*, 2005. 9

[73] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994. 7

[74] M. Rätsch, S. Romdhani, and T. Vetter. Efficient face detection by a cascaded support vector machine using haar-like features. In *Pattern Recognition Symposium*, 2004. 12, 13

[75] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake. Computationally efficient face detection. In *Proc. of ICCV*, 2001. 12, 13

[76] D. Roth, M.-H. Yang, and N. Ahuja. A SNoW-based face detector. In *Proc. of NIPS*, 2000. 12

[77] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Proc. of CVPR*, 1996. 12

[78] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. Technical report, School of Coomputer Science, Carnegie Mellow Univ., CMU-CS-97-201, 1997. 12

[79] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *Proc. of CVPR*, 2007. 6, 7

[80] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999. 2, 3, 11

[81] H. Schneiderman. Feature-centric evaluation for efficient cascaded object detection. In *Proc. of CVPR*, 2004. 9, 11, 12

[82] H. Schneiderman. Learning a restricted bayesian network for object detection. In *Proc. of CVPR*, 2004. 12, 13

[83] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2004. 12, 13

[84] E. Seemann, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *Proc. of CVPR*, 2006. 10, 11

[85] Y. Shan, F. Han, H. S. Sawhney, and R. Kumar. Learning exemplar-based categorization for the detection of multi-view multi-pose objects. In *Proc. of CVPR*, 2006. 10, 11

[86] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *Proc. of ICCV*, 2005. 6, 7

[87] J. Sochman and J. Matas. Waldboost - learning for time constrained sequential detection. In *Proc. of CVPR*, 2005. 8, 11

[88] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi. Pedestrian detection using infrared images and histograms of oriented gradients. In *IEEE Intelligent Vehicles Symposium*, 2006. 5

[89] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *Proc. of CVPR*, 2004. 10, 11

[90] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Proc. of ICCV*, 2005. 10, 11

[91] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proc. of ECCV*, 2006. 6, 7

[92] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of CVPR*, 2001. 1, 3, 4, 5, 7, 8, 11, 13

[93] P. Viola and M. Jones. Fast and robust classification using asymmetric AdaBoost and a detector cascade. In *Proc. of NIPS*, 2002. 7, 11

[94] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Proc. of NIPS*, volume 18, 2005. 11

[95] A. Wald. *Sequential Analysis*. Dover, 1947. 8

[96] P. Wang and Q. Ji. Multi-view face detection under complex scene based on combined svms. In *Proc. of ICPR*, 2004. 12, 13

[97] P. Wang and Q. Ji. Learning discriminant features for multi-view face and eye detection. In *Proc. of CVPR*, 2005. 5, 7

[98] X. Wang, T. X. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *Proc. of ICCV*, 2009. 5, 7

[99] C. A. Waring and X. Liu. Face detection using spectral histograms and SVMs. *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics*, 35(3):467–476, 2005. 5, 7

[100] A. R. Webb. *Statistical Pattern Recognition*. Oxford University Press, 1 edition, 1999. 9

[101] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *Proc. of IEEE Automatic Face and Gesture Recognition*, 2004. 2, 5, 7, 9, 10, 11

[102] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proc. of ICCV*, 2005. 6, 7

[103] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *Proc. of ICCV*, 2007. 11

[104] B. Wu and R. Nevatia. Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *Proc. of CVPR*, 2007. 7

[105] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg. Fast asymmetric learning for cascade face detection. Technical report, Georgia Institute of Technology, GIT-GVU-05-27, 2005. 8, 11

[106] J. Wu, J. M. Rehg, and M. D. Mullin. Learning a rare event detection cascade by direct feature selection. In *Proc. of NIPS*, volume 16, 2004. 9, 11

[107] R. Xiao, H. Zhu, H. Sun, and X. Tang. Dynamic cascades for face detection. In *Proc. of ICCV*, 2007. 5, 8, 9, 11

[108] R. Xiao, L. Zhu, and H. Zhang. Boosting chain learning for object detection. In *Proc. of ICCV*, 2003. 7, 11

[109] J. Yan, S. Li, S. Zhu, and H. Zhang. Ensemble svm regression based multi-view face detection system. Technical report, Microsoft Research, MSR-TR-2001-09, 2001. 12, 13

[110] S. Yan, S. Shan, X. Chen, and W. Gao. Locally assembled binary (LAB) feature with feature-centric cascade for fast and accurate face detection. In *Proc. of CVPR*, 2008. 5, 7, 9, 11

[111] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *IEEE Trans. on PAMI*, 31(7):1195–1209, 2009. 13

[112] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. on PAMI*, 24(1):34–58, 2002. 1, 11, 12, 13

[113] J. Yuan, J. Luo, and Y. Wu. Mining compositional features for boosting. In *Proc. of CVPR*, 2008. 6

[114] C. Zhang, R. Hamid, and Z. Zhang. Taylor expansion based classifier adaptation: Application to person detection. In *Proc. of CVPR*, 2008. 13

[115] C. Zhang and P. Viola. Multiple-instance pruning for learning efficient cascade detectors. In *Proc. of NIPS*, 2007. 9, 11

[116] C. Zhang and Z. Zhang. Winner-take-all multiple category boosting for multi-view face detection. Technical report, Microsoft Research MSR-TR-2009-190, 2009. 11

[117] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu. Boosting local binary pattern (LBP)-based face recognition. In *Proc. Advances in Biometric Person Authentication*, 2004. 5

[118] H. Zhang, W. Gao, X. Chen, and D. Zhao. Object detection using spatial histogram features. *Image and Vision Computing*, 24(4):327–341, 2006. 5, 7

[119] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li. Face detection based on multi-block LBP representation. 2007. 5, 7

[120] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *Proc. of CVPR*, 2006. 5