Detecting and Aligning Faces by Image Retrieval

Xiaohui Shen[1] Zhe Lin[2] Jonathan Brandt[2] Ying Wu[1]

[1]Northwestern University
2145 Sheridan Road, Evanston, IL 60208
{xsh835, yingwu}@eecs.northwestern.edu

[2]Adobe Research
345 Park Ave, San Jose, CA 95110
{zlin, jbrandt}@adobe.com

Abstract

Detecting faces in uncontrolled environments continues to be a challenge to traditional face detection methods[24] due to the large variation in facial appearances, as well as occlusion and clutter. In order to overcome these challenges, we present a novel and robust exemplar-based face detector that integrates image retrieval and discriminative learning. A large database of faces with bounding rectangles and facial landmark locations is collected, and simple discriminative classifiers are learned from each of them. A voting-based method is then proposed to let these classifiers cast votes on the test image through an efficient image retrieval technique. As a result, faces can be very efficiently detected by selecting the modes from the voting maps, without resorting to exhaustive sliding window-style scanning. Moreover, due to the exemplar-based framework, our approach can detect faces under challenging conditions without explicitly modeling their variations. Evaluation on two public benchmark datasets shows that our new face detection approach is accurate and efficient, and achieves the state-of-the-art performance. We further propose to use image retrieval for face validation (in order to remove false positives) and for face alignment/landmark localization. The same methodology can also be easily generalized to other face-related tasks, such as attribute recognition, as well as general object detection.

1. Introduction

Although boosting-based object detection methods[24] and their variations[28] have achieved great success in frontal-view face detection, so-called face detection in the wild (i.e. in unconstrained environments) continues to be a challenge, due to large variation in pose, lighting and expressions, as well as occlusion and clutter. The performance of state-of-the-art methods under such challenging conditions still has considerable room to improve.

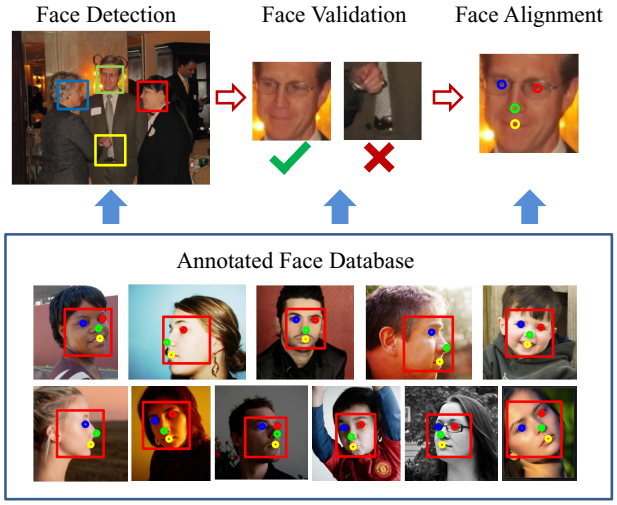


Figure 1. Overview of our retrieval-based face detection system.

Some approaches attempted to learn multiple models to detect faces in different viewpoints[8, 26], while part-based models have also been proposed to address the variations[7, 29]. Nevertheless, it is difficult, if not impossible, to explicitly model all possible variations in facial appearance.

The exemplar-based approach is an intuitive and straightforward alternative, in which a test sample can be directly matched against a collection of face images to determine its label. Without explicit modeling, a face can be detected as long as enough similar exemplars are included in the collection. However, there are two challenges confronting this approach: (1) To achieve good performance, lots of exemplar faces are needed to span the large appearance variation. As a result, simple direct matching methods (e.g. nearest neighbor search) against such a large data collection would be too inefficient.(2) With traditional sliding window scanning, all possible candidate regions at every location and scale for a given test image need to be examined. This also incurs considerable computational costs.

This paper addresses these two challenges by proposing an integration of state-of-the-art image retrieval[20] with

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.444

discriminative learning. Modern bag-of-words-based image retrieval methods allow us to retrieve similar images from millions of database images with near real-time performance. Our new face detector is essentially an image retrieval system that uses a database of face images annotated with bounding rectangles and landmark locations. To achieve robustness, a discriminative classifier is learned from each exemplar face. A voting-based approach is then proposed to let the classifiers project their predictions on the test image during search. The face regions in the test image, even with challenging poses or expressions, shall receive high prediction scores from similar exemplar faces. Face detection is then performed by simply selecting the voting peaks with high scores. Therefore, the detection can be very fast without exhaustive sliding window scanning.

The overview of our approach appears in Fig.1. In addition to the voting-based face detection, we also propose a new face validation step to further boost the detection performance by reducing false positives. Each candidate face rectangle is used to perform search and localization against a face database. True face samples shall retrieve similar faces and accurately localize those faces, while false positives tend to retrieve and localize on non-face image regions, and are consequently removed. We evaluate our method on two public face detection datasets and show that our approach outperforms state-of-the-art methods.

Although we mainly focus on face detection in this paper, since we retrieved similar faces to the test image during validation, robust face alignment can also be achieved as a by-product by transferring landmark locations from the exemplar face images, which is an additional benefit of our method. It can also be potentially extended to other face-related tasks such as attribute recognition as well as general object detection. Moreover, our approach is well suited for online training, as more exemplars can be incrementally added to improve the performance.

The contributions of this paper are three-fold:

1. We propose a novel exemplar-based face detection approach by combining image retrieval with discriminative learning, and designing a voting-based method to efficiently detect faces without exhaustive scanning.

2. We introduce an efficient image retrieval-based framework to simultaneously perform face validation and facial landmark localization.

3. We achieve the stat-of-the-art performance on two challenging face detection benchmarks.

2. Related Work

Face detection is a well studied vision problem, and various features and models have been proposed. Please refer to [28] for a full review. Most work in recent years have followed the paradigm proposed by Viola and Jones[24]. In their original work, a cascade of boosted classifiers is trained using Haar wavelets as features. Sliding window scanning is then performed for face detection. Variants include different features (e.g., HOG-LBP[25], SIFT/SURF[15]) and different boosting algorithms[2, 4, 3]. Multi-view models have been proposed to detect faces under viewpoint changes[8, 26]. Part-based models[7, 18, 5], especially deformable part-based models[6, 29] have also shown their efficacy in detecting faces with variations and occlusions.

Recently, the incorporation of object localization into image retrieval has been studied. Some image search methods not only retrieve similar images, but also localize similar objects in the retrieved images, either by sub-image search[13, 16], or by generalized Hough voting[14, 20]. In [27], face images with the same identities were retrieved. [21] localizes and segments a product in the query image with the help of the top-retrieved images. However, in all of those methods, the query image is given, and the task is to find the identical object or visually similar objects from the database, which is a different task from face detection, as the category of face has much larger appearance variations than a single object. In [1], parts of faces are localized by combining local detector outputs with a consensus of non-parametric global models computed from exemplars. However, they still need pre-trained classifiers (SVM) with sliding-window scanning to detect local facial parts. To the best of our knowledge, there is no previous work on face detection leveraging large-scale image retrieval.

3. Face Detection by Image Retrieval

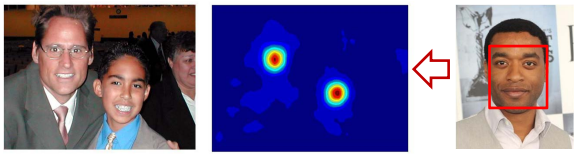
3.1. Exemplar Database

To detect faces using image retrieval, we build a database with 18486 exemplar face images under different viewpoints, poses, expressions and lighting conditions. The face region in the image is around the image center and manually marked with four main facial landmark locations: the center of two eyes, mouth center and nose tip. A rectangle bounding the face is then generated according to the landmark positions[1]. See the database images in Fig.1 for some examples. Some of them are from the Annotated Facial Landmarks in the Wild (AFLW) dataset[12], while others are annotated by ourselves. No images in the testing datasets are included in the database.

3.2. Algorithm

In order to detect faces in a test image by searching the database images, we need to define a similarity measure between any detection window(represented by a

[1]For profile faces, if one eye is invisible due to occlusion, its landmark annotation would be absent.



(a) (b) (c)

Figure 2. The voting-map based method to calculate similarity scores. (a) test image, (c) The face rectangle in an exemplar image, (b) generated voting map when using (c) to vote on (a).

sub-rectangle)[2] in the test image and the face rectangle in a database image. We employ the retrieval approach based on local features, visual vocabulary and inverted files, and choose the spatially-constrained similarity measure proposed in [20, 21], which is a variant of the traditional bag-of-words in image search, but with much better spatial matching consistency:

$$S(x, c_i) = \sum_{k=1}^{N} \sum_{\substack{(f,g) \\ f \in x, g \in c_i \\ w(f)=w(g)=k \\ ||\mathbf{T}(L(f))-L(g)||<\varepsilon}} \frac{\mathrm{idf}^2(k)}{\mathrm{tf}_x(k) \cdot \mathrm{tf}_{c_i}(k)} \qquad (1)$$

where x is a detection window in the test image, and c_i is the face rectangle in the i-th database image. f and g are the local features extracted from x and c_i, respectively. k denotes the k-th visual word in a learned vocabulary. $w(f) = w(g) = k$ means that f and g are both assigned to visual word k. $\mathrm{idf}(k)$ is the inverse document frequency of k, $\mathrm{tf}_x(k)$ and $\mathrm{tf}_{c_i}(k)$ are the term frequencies (i.e., number of occurrence) of k in x and c_i, respectively. $L(f)$ is the 2D image location of f. $\mathbf{T}$ is the spatial transformation that maps rectangle x in the test image to c_i in the exemplar image. We assume $\mathbf{T}$ only consists of scale change and translation. The spatial constraint $||\mathbf{T}(L(f)) - L(g)|| < \varepsilon$ means that the locations of two matched features should be sufficiently close under transformation $\mathbf{T}$.

In [20] and [21], such a similarity measure is efficiently calculated by multi-scale generalized Hough-voting[14]. We use a similar voting-based method to calculate the similarities[3]. Consider that if a feature g inside the face rectangle of an exemplar image is matched with a feature f in a possible positive detection window of a test image, the relative locations of g and f to their respective rectangle centers should be consistent under a certain scale change. Therefore we can calculate the the relative location of g to the face center in the exemplar image, and use that to predict the location of the face center in the test image accordingly. A vote will be cast at that location with score $\frac{\mathrm{idf}^2(k)}{\mathrm{tf}_x(k) \cdot \mathrm{tf}_{c_i}(k)}$ as introduced in Eqn.1. To achieve better detection performance, we differ from [20] in that each vote is further weighted by the distance from the feature g to the face center in the exemplar image. Features closer to the face center will cast votes with higher weights, as they contain more feature information on the faces. Consider Fig.2, for example, if we use all the features in the exemplar face (Fig.2(c)) to vote on the test image (Fig.2(a)) at a certain scale, we can get a voting map as in Fig.2(b), in which the value at each location corresponds to a similarity score between a sub-rectangle (with that location as its center) in Fig.2(a) and the face rectangle (Fig.2(c)). Therefore the similarities between any sub-rectangle of the test image and the exemplars can be obtained from the voting maps, without resorting to sliding window search.

[2]In this paper, we fix the aspect ratio of a detection window to 1.

[3]For the details of the voting-based method, please refer to [20].

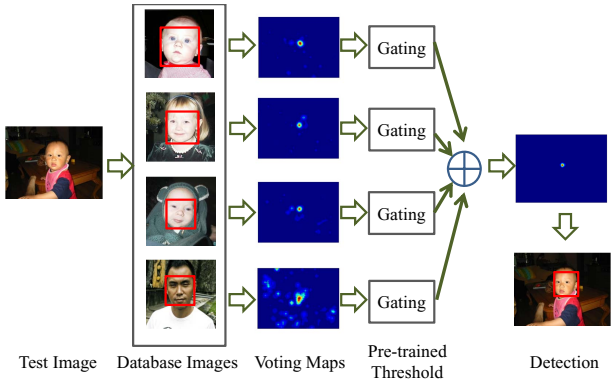


Figure 3. Pipeline of our face detection method. This illustration only shows voting maps at a certain scale, while in practice we generate voting maps at multiple scales.

However, since local features (e.g., SIFT[17]) are quantized for fast retrieval, the similarities between a face exemplar and a non-face test sample can be as high as face-to-face similarities, and the voting maps may be noisy. Therefore, only obtaining and simply aggregating the similarities between test samples and the face exemplars is not sufficient to robustly detect the faces. In fact it only got 58.0% in average precision on the AFW dataset[29].

To this end, we combine image retrieval and discriminative learning, and propose the pipeline of our face detection algorithm as illustrated in Fig.3. Given a test image, we first use all the exemplar faces to vote on the test image and generate corresponding voting maps at multiple scales. Gating is then performed on each voting map, i.e., each map is subtracted by a pre-trained threshold t_i. The threshold t_i corresponding to each exemplar face is discriminatively learned in the training stage, as explained in Section 3.3. The values on the voting maps that are below the threshold are set to zeros. We then aggregate the gated voting maps together to get the final score map. This operation can be interpreted mathematically in the following equation:

$$S(x) = \sum_{i:s_i(x)>t_i} (s_i(x) - t_i) \qquad (2)$$

where $S(x)$ is the final detection score of x, $s_i(x)$ is the similarity score between x and database exemplar c_i, t_i is the corresponding threshold. We can see from Fig.3 that after gating, the noise in the initial voting maps (e.g., in the last row) is filtered out. Based on the aggregated voting maps, we then select the maximal modes from the maps with non-maxima suppression to get the final detection results, as shown in the last column in Fig.3.

The reason we use gating before aggregation is to limit the contributions of irrelevant exemplars to a given test image, or more accurately, to a given sub-rectangle of a test image. The appearance variation of face images can be very large, and we expect that only the exemplars which are very similar to the test region are informative for classification, while the more distant exemplars are uninformative. Therefore our assumption is that, if x is sufficiently similar to c_i, x should be voted as a face with very high probability, while if x is far away from c_i, c_i cannot determine the label of x with any preference. The effect of gating hereby is to determine the effective range of an exemplar. If the similarity $s_i(x)$ is larger than t_i, it means the test sample falls into the close neighborhood of c_i, and accordingly receives a high confidence vote from c_i.

3.3. Naive Bayes Interpretation

The foregoing argument appeals to our intuitive understanding of our algorithm. However it can be more concretely justified in the context of Naive Bayes classification. Suppose that we consider the gated voting of a particular exemplar (in Section 3.2) as a simple classifier. In this context, it is straightforward to show that if we make an independence assumption among the exemplars, then our voting scheme is operating as a Naive Bayes classifier.

Given a set of positive exemplars (i.e., face images) c_i, and a test sample x, let $s_i(x)$ be the similarity between c_i and x. $y \in \{0, 1\}$ is the label of x, $y = 1$ if x is a face. For each positive exemplar c_i, given a small constant value ϵ, suppose there is a threshold t_i such that:

$$\begin{aligned} P(y=1|s_i(x) > t_i) &\geq 1-\epsilon \\ P(y=0|s_i(x) > t_i) &\leq \epsilon \end{aligned} \tag{3}$$

where t_i is a certain threshold, and ϵ is a very small value. It can be considered a hyper-sphere classifier. If x falls into a small hyper-sphere around c_i (i.e., $s_i(x) > t_i$), then it is highly probable that x is a face. If $s_i(x) \leq t_i$, based on our assumption, c_i cannot determine x is a face or not, therefore we assume that c_i has equal contribution to the label of x, i.e., $P(y=1|s_i(x) \leq t_i) = P(y=0|s_i(x) \leq t_i)$.

In the test stage, suppose there are m total exemplar faces, and for simplicity we use s_i to denote the similarity $s_i(x)$, the likelihood ratio can be defined as:

$$L(s_1, ..., s_m) = \frac{P(s_1, ..., s_m|y=1)}{P(s_1, ..., s_m|y=0)} \tag{4}$$

If we assume that the s_i are independent, and take the log operation, we get the Naive Bayes log-likelihood ratio:

$$\begin{aligned} \log L(s_1, ..., s_m) &= \sum_{i=1}^{m} \log \frac{P(s_i|y=1)}{P(s_i|y=0)} \\ &\propto \sum_{i=1}^{m} \log \frac{P(y=1|s_i)}{P(y=0|s_i)} \end{aligned} \tag{5}$$

Suppose there are n exemplars with $s_i(x) > t_i$, based on our assumption, the remaining $m - n$ exemplars with $s_i(x) \leq t_i$ have $\log \frac{P(y=1|s_i)}{P(y=0|s_i)} = 0$. Accordingly we have:

$$\log L(s_1, ..., s_m) = \sum_{i:s_i(x)>t_i} \log \frac{P(y=1|s_i)}{P(y=0|s_i)} \geq n \log \frac{1-\epsilon}{\epsilon} \tag{6}$$

Apparently if more exemplars are close to the test sample (i.e., n is larger), the log-likelihood ratio will be higher, and x is more likely to be a face. Therefore we can use such a log-likelihood ratio to detect faces.

Classification. To calculate the Naive Bayes log-likelihood ratio, we need a detailed form of classifier satisfying Eqn.3. We model the probabilities to be:

$$\begin{aligned} P(y=1|s_i(x) > t_i) &= 1 - \epsilon e^{-f(s_i(x))} \\ P(y=0|s_i(x) > t_i) &= \epsilon e^{-f(s_i(x))} \end{aligned} \tag{7}$$

where $f(s_i(x))$ can be any monotonically increasing function of $s_i(x)$. For a practical purpose, we choose $f(s_i(x)) = s_i(x) - t_i$.[4] Then we have

$$\begin{aligned} \sum_{i:s_i(x)>t_i} \log \frac{P(y=1|s_i)}{P(y=0|s_i)} &= \sum_{i:s_i(x)>t_i} \log \frac{1-\epsilon e^{-(s_i(x)-t_i)}}{\epsilon e^{-(s_i(x)-t_i)}} \\ &= \sum_{i:s_i(x)>t_i} \log(\frac{1}{\epsilon} e^{s_i(x)-t_i} - 1) \end{aligned} \tag{8}$$

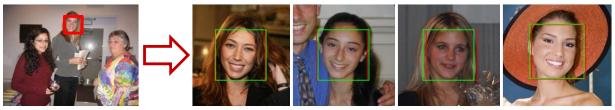
Since ϵ is small, when $s_i(x) - t_i > 0$, we can approximate Eqn.8 and get:

$$\begin{aligned} \sum_{i:s_i(x)>t_i} \log \frac{P(y=1|s_i)}{P(y=0|s_i)} &= \sum_{i:s_i(x)>t_i} \log(\frac{1}{\epsilon} e^{s_i(x)-t_i} - 1) \\ &\approx \sum_{i:s_i(x)>t_i} \log(\frac{1}{\epsilon} e^{s_i(x)-t_i}) \\ &= -n \log \epsilon + \sum_{i:s_i(x)>t_i} (s_i(x) - t_i) \end{aligned} \tag{9}$$

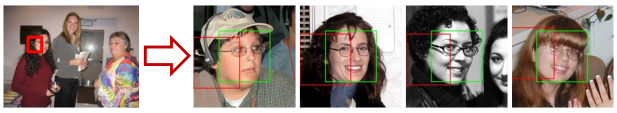
The first term is a constant, and the second term is exactly the aggregated vote score after the gating in Eqn.2.

Classifier training. For each positive exemplar c_i and its corresponding classifier, the threshold t_i needs to be

[4] While $f(s_i(x))$ can take any form as long as it satisfies Eqn.3, we found that $f(s_i(x)) = s_i(x) - t_i$ works quite well in practice.



(a) Validation result of a true positive.



(b) Validation result of a false positive.

Figure 4. The validation step consists of running a second search using the detected window as a query. Valid faces tend to retrieval similar faces and accurately localize on these faces, while invalid detections produce inconsistent search and localization results.

determined. In order to discriminatively learn the threshold, besides the existing positive training samples, we also collected a negative training set $\mathcal{N}$[5].

Given the negative sample set, and a particular exemplar c_i we need to determine a t_i such that $P(s_i(x) > t_i | x \in \mathcal{N})$ is minimized. It is straightforward to see that $P(s_i(x) > t_i | x \in \mathcal{N}) = 0$ if

$$t_i \geq \max_{j \in \mathcal{N}} s_i(x_j). \tag{10}$$

Once satisfying the constraint in Eqn.10, we would like to enlarge the effective hyper-sphere of c_i without losing classification accuracy, i.e., to include as many positive training samples from the positive set $\mathcal{P}$:

$$\begin{aligned} \overline{t_i} \quad &= \quad \arg\max_{t_i} P(s_i(x) > t_i | x \in \mathcal{P}) \\ s.t. \quad & t_i \geq \max_{j \in \mathcal{N}} s_i(x_j) \end{aligned} \tag{11}$$

Apparently when t_i is smaller, the objective function in Eqn.11 is larger. Thus we choose the final threshold as:

$$t_i = \max_{j \in \mathcal{N}} s_i(x_j) \tag{12}$$

This means the threshold is the maximum similarity score between exemplar c_i and any negative training samples.

4. Face Validation

After the face detection step, several candidate face rectangles are obtained. Some of them may not be true faces. Therefore we propose a face validation step using image retrieval again to identify and filter out these false positives and further improve the detection accuracy. We use each detected face window to perform search and localization on a validation face database using the same similarity measure as in Eqn.1 and the similar voting approach as in [20]. The validation database is set as the same as our face database for detection, but it can also be augmented with non-face images for improved discriminability. If the candidate region is a true face, it will retrieve faces with similar poses and meanwhile accurately localize the faces, as shown in Fig.4(a). If it is not a face, then the overlap between the localized rectangle and ground truth rectangle tends to be low, as seen in Fig.4(b). Therefore we use such information to generate the validation score and further refine our face detection results.

Consider that top-k images are retrieved for a detected candidate window x, with a localized rectangle obtained in each retrieved image, we calculate the overlap ratio between the localized rectangle l_i and ground truth rectangle g_i for each retrieved image $I_i (i = 1...k)$:

$$R_i(x) = \frac{l_i \bigcap g_i}{l_i \bigcup g_i} \tag{13}$$

If there are no faces in the retrieved image, then $R_i(x) = 0$.

The validation score is then determined by:

$$V(x) = \sum_{\substack{i=1 \\ R_i(x) > \theta}}^{k} s_i(x) \times R_i(x) \tag{14}$$

where $s_i(x)$ is the similarity score between the test sample x and the i-th retrieved image. The constraint $R_i(x) > \theta$ means that we only consider the retrieved image with overlap ratio greater than θ. In practice, we choose $\theta = 0.6$.

After we obtain the validation score, and the final detection score can be calculated as:

$$D(x) = \alpha S(x) + (1 - \alpha) V(x) \tag{15}$$

which is a linear combination of the initial detection score and the validation score. α is a weight to control the combination, which is determined experimentally through cross validation, and then fixed for all the experiments.

5. Face Alignment

In addition to bounding rectangles, our database faces are annotated with landmark locations. Therefore, we can transfer the facial landmark locations from the images retrieved during validation to the test image. In this way, face alignment can be performed without any additional search cost, which is an additional benefit of our method.

We localize each landmark using a modified version of our voting scheme in face detection, and generate voting maps for each landmark separately. To vote on a landmark, when we find a matched feature pair between the test sample and an exemplar face, we calculate the relative location of the feature to the landmark in the exemplar face image, and vote on the estimated location of that landmark

[5] We collected $\sim$ 5000 images without faces, and use the same voting-based method to calculate the similarities between the positive exemplar and the sub-rectangles in the negative images, which is equivalent to generating negative training samples by multi-scale dense sampling.

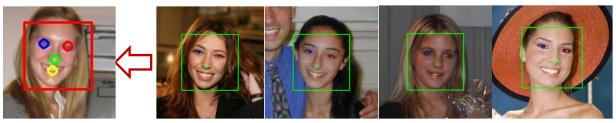


Figure 5. Face alignment and pose estimation using top retrieved face images. The locations of two eyes as well as the mouth and the nose are accurately localized.

in the test sample accordingly. Meanwhile, similar as in face detection, the vote is weighted by the relative distance from the feature to the landmark in the exemplar face. Features closer to the landmark have higher weight. After voting, the peak location in each individual voting map is the estimated landmark location based on c_i.

For a particular landmark, each database image gives us an estimated location e_i. If we have k-top retrieved images, then the final estimated location of that landmark is determined as the per-component median value of $e_1, e_2, ..., e_k$, see Fig.5 for an example.

If the exemplar faces in the database are annotated with additional information (e.g., attributes such as age, gender and expressions), we can use the the top retrieved face images and the same methodology to estimate these attributes in the test image through label transfer..

6. Experiments

6.1. Implementation details

We used combined sparse and dense SIFT[17] as features, and fast approximate k-means[19] to build a 100k vocabulary. The maximum dimension of the exemplar images is 640. To ensure performance, smaller test images are resized to have 1280 pixels as their maximum dimension, while larger images are kept the same. In face detection, the smallest scale on which we vote is 80×80 (in a 1280-pixel dimension image). We vote on 15 scales, and each scale is 1.2 times larger than the previous one.

To speed up the process and reduce the memory, given a test image, we first use the bag-of-words model[22] to retrieve 3000 similar images from the database, and then do voting and face detection using only those retrieved images. Without code optimization, the entire face detection, validation and alignment finishes in less than 10 seconds in C++ implementation. The voting and validation tasks can be parallelized to further reduce the detection time, which shows its potential in real time processing.

6.2. Results

We evaluated our approach on two public datasets with annotated faces in the wild: AFW[29] and FDDB[9]. Both datasets contain faces in uncontrolled conditions with cluttered backgrounds and large variations in both face viewpoint and appearance, and thus bring forward great challenges to the current face detection algorithms.

In the AFW dataset, the results of the following face detection methods are reported in [29]: (1) OpenCV implementations of 2-view Viola-Jones, (2) Boosted 2-view face detector of [11], (3) Deformable part model(DPM)[6], (4) Mixture of trees[29], (5) face.com's face detector and (6) Google Picasa's face detector. Among the academic solutions, [29] significantly outperforms others, and is only slightly below the commercial systems.

The precision-recall curves of our method (face detection with and without validation) on this dataset along with others are shown in Fig.6(a). The results of other methods are provided by [29]. We can see that in our approach, the performance of the initial detection step (without validation) is already among the state-of-the-art. After face validation, our method further outperforms [29], achieving the state-of-the-art in research approaches, and closing the gap with face.com and Google Picasa.

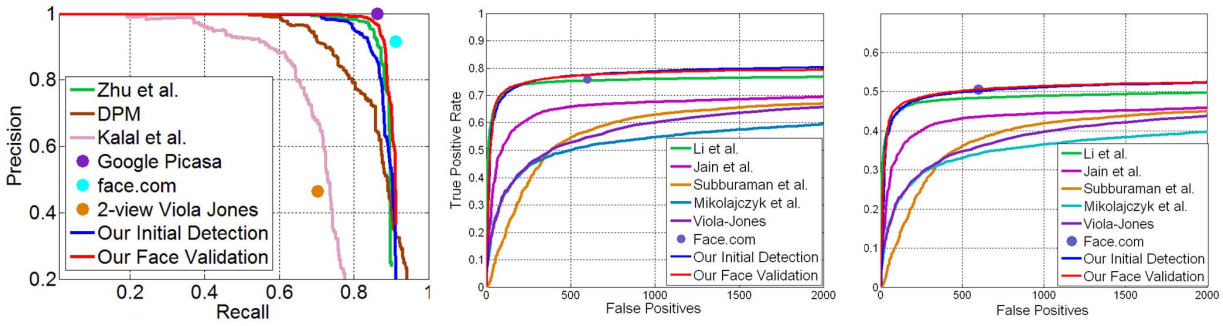
The FDDB benchmark reports the performance of several published methods in the research community on their dataset[6], including: (1) OpenCV implementation of Viola-Jones, (2) Mikolajczyk et al[18], (3) Subburaman et al[23], (4) Jain et al[10] and (5) Li et al[15]. We also report the face detector of face.com on this dataset. The FDDB benchmark includes two methodologies for evaluation: discrete ROC, and continuous ROC[9]. The evaluation of discrete ROC is a common protocol (i.e. requiring at least 50% overlap ratio of the intersection of two regions against the union of the two regions), while in continuous ROC, the overlap ratio is used as a weight to measure the matching quality.

The ROC curves of our approach and others are shown in Fig.6(b) and (c) respectively. On this dataset, our initial face detection has already achieved quite good performance, and face validation does not show much improvement. The performance of our method is even slightly better than face.com's detector. It should be noted that in FDDB, the ground truth are elliptical regions, while the output of our method (as well as face.com) are rectangles. Therefore the overlap of two regions will be smaller than usual, and in fact we have observed some good detections marked as false positives when the rectangles are slightly off centered. Moreover, there are many small faces in the ground-truth files which our method will not detect (the minimum resolution of the ground-truth faces is 20 pixels, while the minimum scale of our detection is 80 pixels in a 1280-resolution image).[7] Nevertheless, our method has already achieved very good results on this benchmark.

Fig.7 shows some examples of our detection results. Our method can accurately detect faces with different

[6] http://vis-www.cs.umass.edu/fddb/results.html.

[7] As argued in [29], relatively large faces in high-resolution images are common given HD photo and video recordings. Meanwhile, smaller faces can be detected by further up-scaling the test images.



(a) PR Curve on AFW (b) Discrete ROC on FDDB (c) Continuous ROC on FDDB

Figure 6. Performance evaluation on two public datasets. We compared with Zhu et al[29], DPM[6], Kalal et al[11], Viola-Jones[24], face.com and Google Picasa on AFW. On FDDB, we compared with Li et al[15], Jain et al[10], Subburaman et al[23], Mikolajczyk et al[18], Viola-Jones and face.com.

Figure 7. Examples of face detection results. Our method can accurately detect faces with large facial appearance variations.

resolutions, poses and attributes, in severe occlusions and cluttered background, as well as blurred face images.

Although the main focus of this paper is face detection, the proposed framework allows us to perform face alignment using the same methodology, as described in Section 5. Our preliminary results show that, in most cases, the localization of the four landmarks was reasonably accurate. From Fig.8 we can see that our approach can accurately localize the landmarks under large facial appearance variations, which shows great potential in more complete face alignment (e.g., eye corners and mouth corners) given the availability of more precise landmark annotations on our exemplar face database [8].

6.3. Discussions

Currently, we include only 18486 face images in the database, without specifically selecting the types of faces, yet our method has already achieved the state-of-the-art performance. In principal, adding more faces to the database will further improve performance since the larger database will better span the face appearance variations. Fortunately, our framework allows us to incrementally add more exemplars in a convenient way, and our approach can be easily extended to an online setting. Meanwhile, how to design a better database for face detection is an interesting problem that merits further study.

7. Conclusions

In this paper, we propose a robust face detector by combining state-of-the-art visual search with discriminative learning. Simple discriminative classifiers are learned for the exemplar face images in the database and collaboratively cast their prediction scores on the test image. Face detection is then efficiently performed by selecting modes from multi-scale voting maps. A face validation step using image retrieval is further proposed, and face alignment can

[8]Please see http://users.eecs.northwestern.edu/~xsh835/CVPR13Sup.zip for more results.

Figure 8. Examples of face alignment. The landmarks are accurately localized in different conditions.

be performed at the same time without additional cost. The evaluation on two public face detection datasets shows that our approach outperforms other state-of-the-art methods. Moreover, our framework can potentially be extended to other face-related tasks and general object detection, which leads to interesting future work.

Acknowledgements. This work is done partially when the first author was an intern at Adobe, and in part supported by National Science Foundation grant IIS-0916607, IIS-1217302, US Army Research Laboratory and the US Army Research Office under grant ARO W911NF-08-1-0504, and DARPA Award FA 8650-11-1-7149.

References

[1] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011.

[2] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *CVPR*, 2005.

[3] 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. *IJCV*, 77, 2008.

[4] H. Cevikalp and B. Triggs. Efficient object detection using cascades of nearest convex model classifiers. In *CVPR*, 2012.

[5] S. Dai, M. Yang, Y. Wu, and A. K. Katsaggelos. Detector ensemble. In *CVPR*, 2007.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2009.

[7] B. Heisele, T. Serre, and T. Poggio. A component-based framework for face detection and identification. *IJCV*, 74(2), 2007.

[8] C. Huang, H. Ai, Y. Li, and S. Lao. High-performance rotation invariant multiview face detection. *PAMI*, 2007.

[9] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, 2010.

[10] V. Jain and E. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR*, 2011.

[11] Z. Kalal, J. Matas, and K. Mikolajczyk. Weighted sampling for large-scale boosting. In *BMVC*, 2008.

[12] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[13] C. H. Lampert. Detecting objects in large image collections and videos by efficient subimage retrieval. In *ICCV*, 2009.

[14] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[15] J. Li, T. Wang, and Y. Zhang. Face detection using surf cascade. In *ICCV Workshops*, 2011.

[16] Z. Lin and J. Brandt. A local bag-of-features model for large-scale object retrieval. In *ECCV*, 2010.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

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

[19] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

[20] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn reranking. In *CVPR*, 2012.

[21] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Mobile product image search by automatic query object extraction. In *ECCV*, 2012.

[22] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[23] B. S. Venkatesh and S. Marcel. Fast bounding box estimation based face detection. In *ECCV Workshop on Face Detection*, 2010.

[24] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[25] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009.

[26] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *FG*, 2004.

[27] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum. Scalable face image retrieval with identity-based quantization and multireference reranking. *PAMI*, 33(10), 2011.

[28] C. Zhang and Z. Zhang. A survey of recent advances in face detection. *Technical Report, MSR-TR-2010-66*, 2010.

[29] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.