FDDB: A Benchmark for Face Detection in Unconstrained Settings

Vidit Jain
University of Massachusetts Amherst
Amherst MA 01003
vidit@cs.umass.edu

Erik Learned-Miller
University of Massachusetts Amherst
Amherst MA 01003
elm@cs.umass.edu

Abstract

Despite the maturity of face detection research, it remains difficult to compare different algorithms for face detection. This is partly due to the lack of common evaluation schemes. Also, existing data sets for evaluating face detection algorithms do not capture some aspects of face appearances that are manifested in real-world scenarios. In this work, we address both of these issues. We present a new data set of face images with more faces and more accurate annotations for face regions than in previous data sets. We also propose two rigorous and precise methods for evaluating the performance of face detection algorithms. We report results of several standard algorithms on the new benchmark.

1. Introduction

Face detection has been a core problem in computer vision for more than a decade. Not only has there been substantial progress in research, but many techniques for face detection have also made their way into commercial products such as digital cameras. Despite this maturity, algorithms for face detection remain difficult to compare, and are somewhat brittle to the specific conditions under which they are applied. One difficulty in comparing different face detection algorithms is the lack of enough detail to reproduce the published results. Ideally, algorithms should be published with sufficient detail to replicate the reported performance, or with an executable binary. However, in the absence of these alternatives, it is important to establish better benchmarks of performance.

For a data set to be useful for evaluating face detection, the locations of all faces in these images need to be annotated. Sung et al. [24] built one such data set. Although this data set included images from a wide range of sources including scanned newspapers, all of the faces appearing in these images were upright and frontal. Later, Rowley et al. [18] created a similar data set with images that included faces with in-plane rotation. Schneiderman et al. [20, 21] combined these two data sets with an additional collection of profile face images, which is commonly known as the MIT+CMU data set. Since this resulting collection contains only grayscale images, it is not applicable for evaluating face detection systems that employ color information as well [6]. Some of the subsequent face detection data sets included color images, but they also had several shortcomings. For instance, the GENKI data set [25] includes color images that show a range of head poses (yaw, pitch $\pm 45^\circ$. roll $\pm 20^\circ$), but every image in this collection contains exactly one face. Similarly, the Kodak [13], UCD [23] and VT-AAST [1] data sets included images of faces with occlusions, but the small sizes of these data sets limit their utility in creating effective benchmarks for face detection algorithms.

One contribution of this work is the creation of a new data set that addresses the above-mentioned issues. Our data set includes

- 2845 images with a total of 5171 faces;
- a wide range of difficulties including occlusions, difficult poses, and low resolution and out-of-focus faces;
- the specification of face regions as elliptical regions; and
- both grayscale and color images.

Another limitation of the existing benchmarks is the lack of a specification for evaluating the output of an algorithm on a collection of images. In particular, as noted by Yang et al. [28], the reported performance measures depend on the definition of a "correct" detection result. The definition of correctness can be subtle. For example, how should we score an algorithm which provides two detections, each of which covers exactly 50% of a face region in an image? Since the evaluation process varies across the published results, a comparison of different algorithms remains difficult. We address this issue by presenting a new evaluation scheme with the following components:

- An algorithm to find correspondences between a face detector's output regions and the annotated face regions.
- Two separate rigorous and precise methods for evaluating any algorithm's performance on the data set. These two methods are intended for different applications.
- Source code for implementing these procedures.

We hope that our new data set, the proposed evaluation scheme, and the publicly available evaluation software will make it easier to precisely compare the performance of algorithms, which will further prompt researchers to work on more difficult versions of the face detection problem.

The report is organized as follows. In Section 2, we discuss the challenges associated with comparing different face detection approaches. In Section 3, we outline the construction of our data set. Next, in Section 4, we describe a semi-automatic approach for removing duplicate images in a data set. In Section 5, we present the details of the annotation process, and finally in Section 6, we present our evaluation scheme.

2. Comparing face detection approaches

Based of the range of acceptable head poses, face detection approaches can be categorized as

- **single pose**: the head is assumed to be in a single, upright pose (frontal [24, 18, 26] or profile [21]);
- **rotation-invariant**: in-plane rotations of the head are allowed [8, 19];
- **multi-view**: out-of-plane rotations are binned into a pre-determined set of views [7, 9, 12];
- **pose-invariant**: no restrictions on the orientation of the head [16, 22].

Moving forward from previous comparisons [28] of approaches that focus on limited head orientations, we intend to evaluate different approaches for the most general, i.e., the pose-invariant, face detection task.

One challenge in comparing face detection systems is the lack of agreement on the desired output. In particular, while many approaches specify image regions – e.g., rectangular regions [26] or image patches with arbitrary shape [17] – as hypotheses for face regions, others idetify the locations of various facial landmarks such as the eyes [27]. Still others give an estimate of head pose [16] as well.

The scope of this work is limited to the evaluation of region-based output alone (although we intend to follow this report in the near future with a similar evaluation of 3D pose estimation algorithms). To this end, we annotate each face region with an ellipse of arbitrary size, shape, and orientation, showing the approximate face region for each face in the image. Compared to the traditional rectangular annotation of faces, ellipses are generally a better fit to face regions and still maintain a simple parametric shape to describe the face. We discuss the details of the annotation process in Section 5. Note that our data set is amenable to any additional annotations including facial landmarks and head pose information, which would be beneficial for benchmarking the next generation of face detection algorithms.

Next we discuss the origins and construction of our database.

3. FDDB: Face Detection Data set and Benchmark

Berg et al. [2] created a data set that contains images and associated captions extracted from news articles (see Figure 1). The images in this collection display large variation in pose, lighting, background and appearance. Some of these variations in face appearance are due to factors such as motion, occlusions, and facial expressions, which are characteristic of the unconstrained setting for image acquisition. The annotated faces in this data set were selected based on the output of an automatic face detector. An evaluation of face detection algorithms on the existing set of annotated faces would favor the approaches with outputs highly correlated with this base detection algorithm. This property of the existing annotations makes them unsuitable for evaluating different approaches for face detection. The richness of the images included in this collection, however, motivated us to build an index of *all* of the faces present in a subset of images from this collection. We believe that benchmarking face detection algorithms on this data set will provide good estimates of their expected performance in unconstrained settings.

3.1. Construction of the data set

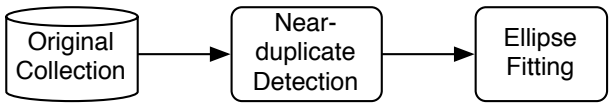


Figure 2. *Outline of the labeling process*. Semi-automatic approaches are developed for both of these steps.

The images in Berg et al.'s data set were collected from the Yahoo! news website,[1] which accumulates news articles from different sources. Although different news organizations may cover a news event independently of each other, they often share photographs from common sources such as the Associated Press or Reuters. The published

[1] http://news.yahoo.com

Figure 1. Example images from Berg et al.'s data set.

photographs, however, may not be digitally identical to each other because they are often modified (e.g., cropped or contrast-corrected) before publication. This process has led to the presence of multiple copies of *near-duplicate* images in Berg et al.'s data set. Note that the presence of such near-duplicate images is limited to a few data collection domains such as news photos and those on the internet, and is not a characteristic of most practical face detection application scenarios. For example, it is uncommon to find near-duplicate images in a personal photo collection. Thus, an evaluation of face detection algorithms on a data set with multiple copies of near-duplicate images may not generalize well across domains. For this reason, we decided to identify and remove as many near duplicates from our collection as possible. We now present the details of the duplicate detection.

4. Near-duplicate detection

We selected a total of 3527 images (based on the chronological ordering) from the image-caption pairs of Berg et al. [2]. Examining pairs for possible duplicates in this collection in the naïve fashion would require approximately 12.5 million annotations. An alternative arrangement would be to display a set of images and manually identify groups of images in this set, where images in a single group are near-duplicates of each other. Due to the large number of images in our collection, it is unclear how to display all the images simultaneously to enable this manual identification of near-duplicates in this fashion.

Identification of near-duplicate images has been studied for web search [3, 4, 5]. However, in the web search domain, scalability issues are often more important than the detection of all near-duplicate images in the collection. Since we are interested in discovering *all* of the near-duplicates in our data set, these approaches are not directly applicable to our task. Zhang et al. [29] presented a more computationally intensive approach based on stochastic attribute relational graph (ARG) matching. Their approach

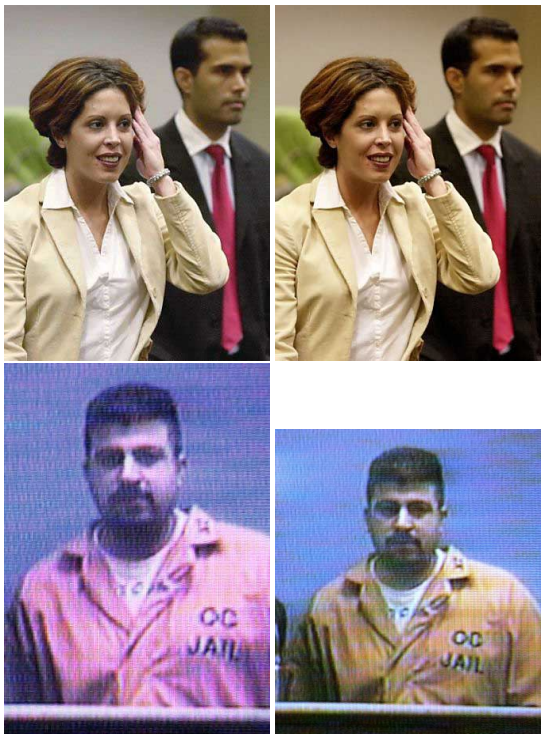


Figure 3. *Near-duplicate images*. (**Positive**) The first two images differ from each other slightly in the resolution and the color and intensity distributions, but the pose and expression of the faces are identical, suggesting that they were derived from a single photograph. (**Negative**) In the last two images, since the pose is different, we do *not* consider them as near-identical images.

was shown to perform well on a related problem of detecting near-identical frames in news video databases. These ARGs represent the compositional parts and part-relations of image scenes over several interest points detected in an image. To compute a matching score between the ARGs constructed for two different images, a generative model for the graph transformation process is employed. This approach has been observed to achieve high recall of near-

duplicates, which makes it appropriate for detecting similar images in our data set.

As with most automatic approaches for duplicate detection, this approach has a trade-off among false positives and false negatives. To restrict the number of false positives, while maintaining a high true positive rate, we follow an iterative approach (outlined in Algorithm 1) that alternates between clustering and manual inspection of the clusters. We cluster (steps 3-5 of Algorithm 1) using a spectral graph-clustering approach [15]. Then, we manually label each non-singleton cluster from the preceding step as either *uniform*, meaning that it contains images that are all near duplicates of each other, or *non-uniform*, meaning that at least one pair of images in the cluster are not near duplicates of each other. Finally, we replace each uniform cluster with one of the images belonging to it.

For the clustering step, in particular, we construct a fully-connected undirected graph G over all the images in the collection, where the ARG-matching scores are used as weights for the edges between each pair of images. Following the spectral graph-clustering approach [15], we compute the (unnormalized) Laplacian L_G of graph G as

$$L_G = diag(\mathbf{d}) - W_G, \quad (1)$$

where $\mathbf{d}$ is the set of degrees of all the nodes in G, and W_G is the adjacency matrix of G. A projection of the graph G into a subspace spanned by the top few eigenvectors of L_G provides an effective distance metric between all pairs of nodes (images, in our case). We perform mean-shift clustering with a narrow kernel in this projected space to obtain clusters of images.

Algorithm 1 Identifying near-duplicate images in a collection

1: Construct a graph $G = \{V, E\}$, where V is the set of images, and E are all pairwise edges with weights as the ARG matching scores.
2: **repeat**
3: Compute the Laplacian of G, L_G.
4: Use the top m eigenvectors of L_G to project each image onto $\mathbb{R}^m$.
5: Cluster the projected data points using mean-shift clustering with a small-width kernel.
6: Manually label each cluster as either *uniform* or *non-uniform*.
7: Collapse the *uniform* clusters onto their centroids, and update G.
8: **until** none of the clusters can be collapsed.

Using this procedure, we were able to arrange the images according to their mutual similarities. Annotators were asked to identify clusters in which all images were derived from the same source. Each of these clusters was replaced by a single exemplar from the cluster. In this process we manually discovered 103 uniform clusters over seven iterations, with 682 images that were near-duplicates. Additional manual inspections were performed to find an additional three cases of duplication.

Next we describe our annotation of face regions.

5. Annotating face regions

As a preliminary annotation, we drew bounding boxes around all the faces in 2845 images. From this set of annotations, all of the face regions with height or width less than 20 pixels were excluded, resulting in a total of 5171 face annotations in our collection.

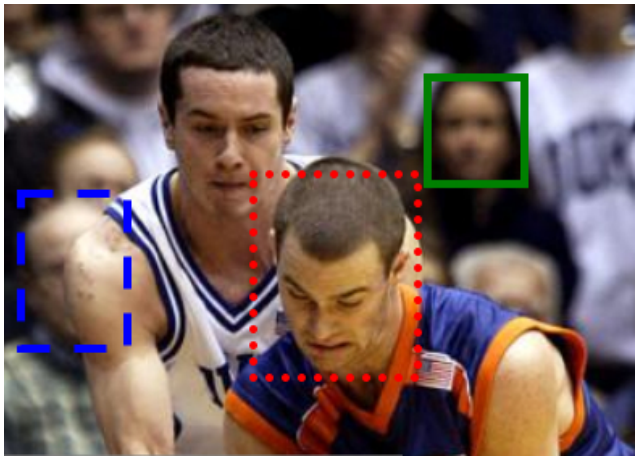


Figure 4. *Challenges in face labeling*. For some image regions, deciding whether or not it represents a "face" can be challenging. Several factors such as low resolution (green, solid), occlusion (blue, dashed), and pose of the head (red, dotted) may make this determination ambiguous.

For several image regions, the decision of labeling them as face regions or non-face regions remains ambiguous due to factors such as low resolution, occlusion, and head-pose (e.g., see Figure 4). One possible approach for handling these ambiguities would be to compute a quantitative measure of the "quality" of the face regions, and reject the image regions with the value below a pre-determined threshold. We were not able, however, to construct a satisfactory set of objective criteria for making this determination. For example, it is difficult to characterize the spatial resolution needed to characterize an image patch as a face. Similarly, for occluded face regions, while a threshold based on the fraction of the face pixels visible could be used as a criterion, it can be argued that some parts of the face (e.g., eyes) are more informative than other parts. Also, note that for the current set of images, all of the regions with faces looking away from the camera have been labeled as non-face regions. In other words, the faces with the angle between the nose (specified as radially outward perpendicular to the

head) and the ray from the camera to the person's head is less than 90 degrees. Estimating this angle precisely from an image is difficult.

Due to the lack of an objective criterion for including (or excluding) a face region, we resort to human judgments for this decision. Since a single human decision for determining the label for some image regions is likely to be inconsistent, we used an approach based on the agreement statistics among multiple human annotators. All of these face regions were presented to different people through a web interface to obtain multiple independent decisions about the validity of these image regions as face regions. The annotators were instructed to reject the face regions for which neither of the two eyes (or glasses) were visible in the image. They were also requested to reject a face region if they were unable to (qualitatively) estimate its position, size, or orientation. The guidelines provided to the annotators are described in Appendix A.

5.1. Elliptical Face Regions

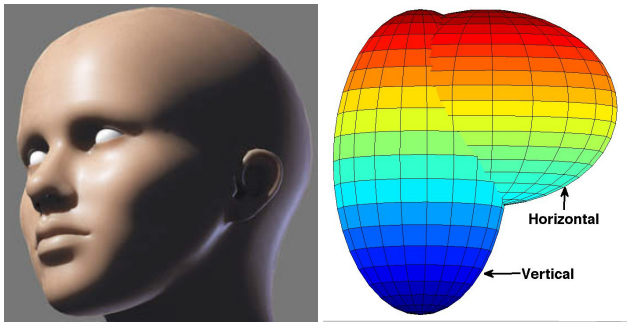


Figure 5. *Shape of a human head*. The shape of a human head (**left**) can be approximated as the union of two ellipsoids (**right**). We refer to these ellipses as vertical and horizontal ellipsoids.

As shown in Figure 5,[2] the shape of a human head can be approximated using two three-dimensional ellipsoids. We call these ellipsoids the *vertical* and *horizontal* ellipsoids. Since the horizontal ellipsoid provides little information about the features of the face region, we estimate a 2D ellipse for the orthographic projection of the hypothesized vertical ellipsoid in the image plane. We believe that the resulting representation of a face region as an ellipse provides a more accurate specification than a bounding box without introducing any additional parameters.

We specified each face region using an ellipse parameterized by the location of its center, the lengths of its major and minor axes, and its orientation. Since a 2D orthographic projection of the human face is often not elliptical, fitting an ellipse around the face regions in an image is challenging. To make consistent annotations for all the faces in our data set, the human annotators are instructed to follow the guidelines shown in Figure 6. Figure 7 shows some sample annotations. The next step is to produce a consistent and reasonable evaluation criterion.

[2]Reproduced with permission from Dimitar Nikolov, Lead Animator, Haemimont Games.

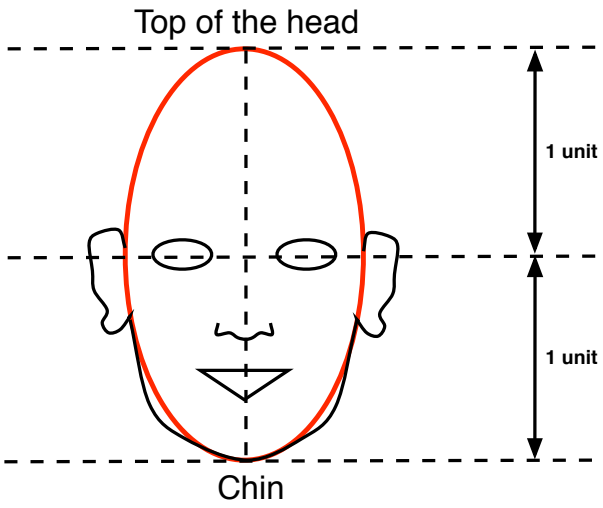


Figure 6. *Guidelines for drawing ellipses around face regions*. The extreme points of the major axis of the ellipse are respectively matched to the chin and the topmost point of the hypothetical vertical ellipsoid used for approximating the human head (see Figure 5). Note that this ellipse does not include the ears. Also, for a non-frontal face, at least one of the lateral extremes (left or right) of this ellipse are matched to the boundary between the face region and the corresponding (left or right) ear. The details of our specifications are included in Appendix A.

Figure 7. *Sample Annotations*. The two red ellipses specify the location of the two faces present in this image. Note that for a non-frontal face (**right**), the ellipse traces the boundary between the face and the visible ear. As a result, the elliptical region includes pixels that are not a part of the face.

6. Evaluation

To establish an evaluation criterion for detection algorithms, we first specify some assumptions we make about their outputs. We assume that

- A detection corresponds to a contiguous image region.
- Any post-processing required to merge overlapping or similar detections has already been done.
- Each detection corresponds to exactly one entire face, no more, no less. In other words, a detection cannot be considered to detect two faces at once, and two detections cannot be used together to detect a single face. We further argue that if an algorithm detects multiple disjoint parts of a face as separate detections, only one of them should contribute towards a positive detection and the remaining detections should be considered as false positives.

To represent the degree of match between a detection d_i and an annotated region l_j, we employ the commonly used ratio of intersected areas to joined areas:

$$S(d_i, l_j) = \frac{area(d_i) \cap area(l_j)}{area(d_i) \cup area(l_j)}. \quad (2)$$

To specify a more accurate annotation for the image regions corresponding to human faces than is obtained with the commonly used rectangular regions, we define an elliptical region around the pixels corresponding to these faces. While this representation is not as accurate as a pixel-level annotation, it is a clear improvement over the rectangular annotations in existing data sets.

To facilitate manual labeling, we start with an automated guess about face locations. To estimate the elliptical boundary for a face region, we first apply a skin classifier on the image pixels that uses their hue and saturation values. Next, the holes in the resulting face region are filled using a flood-fill implementation in MATLAB. Finally, a moments-based fit is performed on this region to obtain the parameters of the desired ellipse. The parameters of all of these ellipses are manually verified and adjusted in the final stage.

6.1. Matching detections and annotations

A major remaining question is how to establish a correspondence between a set of detections and a set of annotations. While for very good results on a given image, this problem is easy, it can be subtle and tricky for large numbers of false positives or multiple overlapping detections (see Figure 8 for an example). Below, we formulate this problem of matching annotations and detections as finding a maximum weighted matching in a bipartite graph (as shown in Figure 9).

Figure 8. *Matching detections and annotations*. In this image, the ellipses specify the face annotations and the five rectangles denote a face detector's output. Note that the second face from left has two detections overlapping with it. We require a valid matching to accept only one of these detections as the true match, and to consider the other detection as a false positive. Also, note that the third face from the left has no detection overlapping with it, so no detection should be matched with this face. The blue rectangles denote the true positives and yellow rectangles denote the false positives in the desired matching.

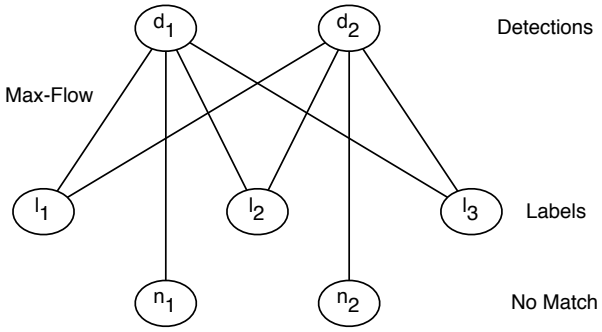


Figure 9. *Maximum weight matching in a bipartite graph*. We make an injective (one-to-one) mapping from the set of detected image regions d_i to the set of image regions l_i annotated as face regions. The property of the resulting mapping is that it maximizes the cumulative similarity score for all the detected image regions.

Let L be the set of annotated face regions (or labels) and D be the set of detections. We construct a graph G with the set of nodes $V = L \cup D$. Each node d_i is connected to each label $l_j \in L$ with an edge weight w_{ij} as the score computed in Equation 2. For each detection $d_i \in D$, we further introduce a node n_i to correspond to the case when this detection d_i has no matching face region in L.

A matching of detections to face regions in this graph corresponds to the selection of a set of edges $M \subseteq E$. In the desired matching of nodes, we want every detection to be matched to at most one labeled face region, and every labeled face region to be matched to at most one detection.

Note that the nodes n_k have a degree equal to one, so they can be connected to at most one detection through M as well. Mathematically, the desired matching M maximizes the cumulative matching score while satisfying the following constraints:

$$\forall d \in D, \exists l \in \{L \cup N\}, \quad d \xrightarrow{M} l \tag{3}$$

$$\forall l \in L, \nexists d, d' \in D, \quad d \xrightarrow{M} l \wedge d' \xrightarrow{M} l \tag{4}$$

The determination of the minimum weight matching in a weighted bipartite graph has an equivalent dual formulation as finding the solution of the minimum weighted (vertex) cover problem on a related graph. This dual formulation is exploited by the Hungarian algorithm [11] to obtain the solution for the former problem. For a given image, we employ this method to determine the matching detections and ground-truth annotations. The resulting similarity score is used for evaluating the performance of the detection algorithm on this image.

6.2. Evaluation metrics

Let d_i and v_i denote the i^{th} detection and the corresponding matching node in the matching M obtained by the algorithm described in Section 6.1, respectively. We propose the following two metrics for specifying the score y_i for this detection:

- **Discrete score (DS)** : $y_i = \delta_{S(d_i,v_i)>0.5}$.
- **Continuous score (CS)**: $y_i = S(d_i, v_i)$.

For both of these choice of scoring the detections, we recommend analyzing the Receiver Operating Characteristic (ROC) curves to compare the performance of different approaches on this data set. Although comparing the area under the ROC curve is equivalent to a non-parametric statistical hypothesis test (Wilcoxon signed-rank test), it is plausible that the cumulative performances of none of the compared approaches is better than the rest with statistical significance. Furthermore, it is likely that for some range of performance, one approach could outperform another, whereas the relative comparison is reversed for a different range. For instance, one detection algorithm might be able to maintain a high level of precision for low recall values, but the precision drops sharply after a point. This trend may suggest that this detector would be useful for application domains such as biometrics-based access controls, which may require high precision values, but can tolerate low recall levels. The same detector may not be useful in a setting (e.g., surveillance) that would requires the retrieval of all the faces in an image or scene. Hence, the analysis of the entire range of ROC curves should be done for determining the strengths of different approaches.

7. Experimental Setup

For an accurate and useful comparison of different approaches, we recommend a distinction based on the training data used for estimating their parameters. In particular, we propose the following experiments:

EXP-1: 10-fold cross-validation

For this experiment, a 10-fold cross-validation is performed using a fixed partitioning of the data set into ten folds.[3] The cumulative performance is reported as the average curve of the ten ROC curves, each of which is obtained for a different fold as the validation set.

EXP-2: Unrestricted training

For this experiment, data outside the FDDB data set is permitted to be included in the training set. The above-mentioned ten folds of the data set are separately used as validation sets to obtain ten different ROC curves. The cumulative performance is reported as the average curve of these ten ROC curves.

8. Benchmark

For a proper use of our data set, we provide the implementation (C++ source code) of the algorithms for matching detections and annotations (Section 6.1), and computing the resulting scores (Section 6.2) to generate the performance curves at `http://vis-www.cs.umass.edu/fddb/results.html`. To use our software, the user needs to create a file containing a list of the output of this detector. The format of this input file is described in Appendix B.

In Figure 10, we present the results for the following approaches for the above-mentioned EXP-2 experimental setting:

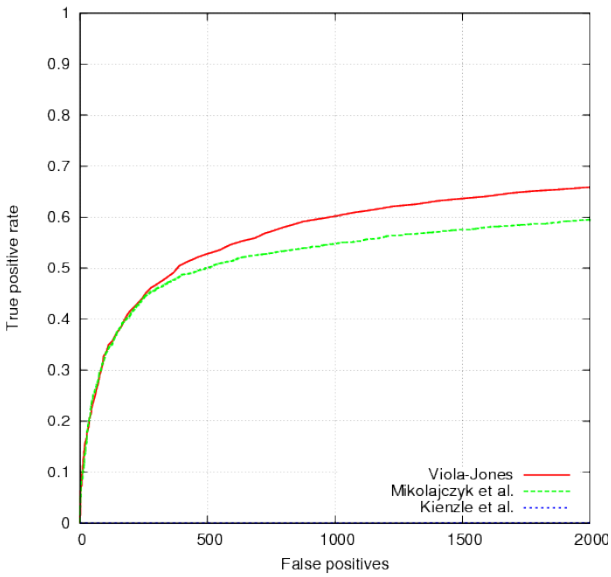
- Viola-Jones detector [26] – we used the OpenCV[4] implementation of this approach. We set the scale-factor and minimum number of neighbors parameters to 1.2 and 0, respectively.
- Mikolajczyk's face detector [14] [5] – we set the parameter for the minimum distance between eyes in a detected face to 5 pixels.
- Kienzle et al.'s [10] face detection library (*fdlib* [6]).

[3] The ten folds used in the proposed experiments are available at `http://vis-www.cs.umass.edu/fddb/FDDB-folds.tgz`

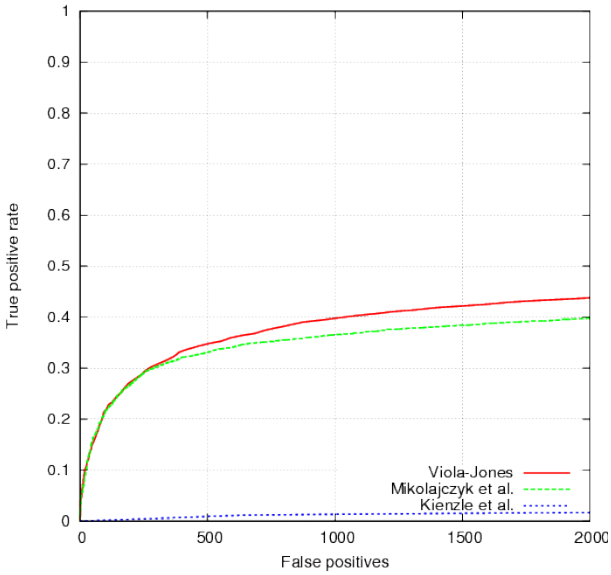
[4] `http://sourceforge.net/projects/opencvlibrary/`

[5] `http://www.robots.ox.ac.uk/~vgg/research/affine/face_detectors.html`

[6] `http://www.kyb.mpg.de/bs/people/kienzle/fdlib/fdlib.htm`



(a) ROC curves based on discrete score (DS)



(b) ROC curves based on continuous score (CS)

Figure 10. *FDDB baselines*. These are the ROC curves for different face detection algorithms. Both of these scores (DS and CS) are described in Section 6.2, whereas the implementation details of these algorithms are included in Section 8.

As seen in Figure 10, the number of false positives obtained from all of these face detection systems increases rapidly as the true positive rate increases. Note that the performances of all of these systems on the new benchmark are much worse than those on the previous benchmarks, where they obtain less than 100 false positives at a true positive rate of 0.9. Also note that although our data set includes images of frontal and non-frontal faces, the above experiments are limited to the approaches that were developed for frontal face detection. This limitation is due to the unavailability of a public implementation of multi-pose or pose-invariant face detection system. Nevertheless, the new benchmark includes more challenging examples of face appearances than the previous benchmarks. We hope that our benchmark will further prompt researchers to explore new research directions in face detection.

Acknowledgements

We thank Allen Hanson, Andras Ferencz, Jacqueline Feild, and Gary Huang for useful discussions and suggestions. This work was supported by the National Science Foundation under CAREER award IIS-0546666. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

[1] A. S. Abdallah, M. A. El-nasr, and A. L. Abbott. A new color image database for benchmarking of automatic face detection and human skin segmentation techniques, to appear. In *International Conference on Machine Learning and Pattern Recognition*, 2007. 1

[2] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 848–854, 2004. 2, 3

[3] O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In *ACM International Conference on Image and Video Retrieval*, pages 549–556, New York, NY, USA, 2007. ACM. 3

[4] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *British Machine Vision Conference*, 2008. 3

[5] J. J. Foo, J. Zobel, R. Sinha, and S. M. M. Tahaghoghi. Detection of near-duplicate images for web search. In *ACM International Conference on Image and Video Retrieval*, pages 557–564, New York, NY, USA, 2007. ACM. 3

[6] R.-L. Hsu, M. Abdel-Mottaleb, and A. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706, May 2002. 1

[7] C. Huang, H. Ai, Y. Li, and S. Lao. High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):671–686, 2007. 2

[8] B. H. Jeon, S. U. Lee, and K. M. Lee. Rotation invariant face detection using a model-based clustering algorithm. In *IEEE International Conference on Multimedia and Expo*, volume 2, pages 1149–1152 vol.2, 2000. 2

[9] M. J. Jones and P. A. Viola. Fast multi-view face detection. Technical Report TR2003-96, Mitsubishi Electric Research Laboratories, August 2003. 2

[10] W. Kienzle, G. H. Bakır, M. O. Franz, and B. Schölkopf. Face detection — efficient and rank deficient. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, pages 673–680, Cambridge, MA, 2005. MIT Press. 7

[11] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955. 7

[12] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. In *European Conference on Computer Vision*, pages 67–81, London, UK, 2002. Springer-Verlag. 2

[13] A. Loui, C. Judice, and S. Liu. An image database for benchmarking of automatic face detection and recognition algorithms. In *IEEE International Conference on Image Processing*, volume 1, pages 146–150 vol.1, Oct 1998. 1

[14] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision*, pages 69–82, 2004. 7

[15] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001. 4

[16] M. Osadchy, Y. LeCun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 8:1197–1215, 2007. 2

[17] J. Rihan, P. Kohli, and P. Torr. OBJCUT for face detection. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 576–584, 2006. 2

[18] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998. 1, 2

[19] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 38, Washington, DC, USA, 1998. IEEE Computer Society. 2

[20] H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 45, Washington, DC, USA, 1998. IEEE Computer Society. 1

[21] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 746–751 vol.1, 2000. 1, 2

[22] M. Seshadrinathan and J. Ben-Arie. Pose invariant face detection. In *Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on*, volume 1, pages 405–410 vol.1, July 2003. 2

[23] P. Sharma and R. Reilly. A colour face image database for benchmarking of automatic face detection algorithms. In *EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, volume 1, pages 423–428 vol.1, July 2003. 1

[24] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998. 1, 2

[25] http://mplab.ucsd.edu. The MPLab GENKI Database, GENKI-4K Subset. 1

[26] P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004. 2, 7

[27] P. Wang and Q. Ji. Multi-view face and eye detection using discriminant features. *Computer Vision and Image Understanding*, 105(2):99–111, 2007. 2

[28] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002. 1, 2

[29] D.-Q. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *ACM International Conference on Multimedia*, pages 877–884, 2004. 3

A. Guidelines for annotating faces using ellipses

To ensure consistency across multiple human annotators, we developed a set of instructions (shown in Figure 11). These instructions specify how to use facial landmarks to fit an ellipse depending on the pose of the head. Figure 12 presents an illustration of the resulting ellipses on line drawings of a human head. The annotators were futher instructed to follow a combination of these guidelines to fit ellipses to faces with complex head poses.

The illustrations shown in Figure 12 use faces with neutral expressions. A presence of some expressions such as laughter, often changes the shape of the face significantly. Moreover, even bearing a neutral expression, some faces have shapes markedly different from the average face shape used in these illustrations. Such faces (e.g., faces with square-jaw or double-chin) are difficult to approximate using ellipses. To annotate faces with such complexities, the annotators were instructed to refer to the following guidelines:

- **Facial expression**. Since the distance from the eyes to the chin in a face with facial expression is not necessarily equal to the distance between the eyes and the top of the head (an assumption made for the ideal head), the eyes do not need to be aligned to the minor axis for this face.

- **Double-chin**. For faces with a double chin, the average of the two chins is considered as the lowest point of the face, and is matched to the bottom extreme of the major axis of the ellipse.

- **Square jaw**. For a face with a square jaw, the ellipse traces the boundary between the face and the ears, while some part of the jaws may be excluded from the ellipse.

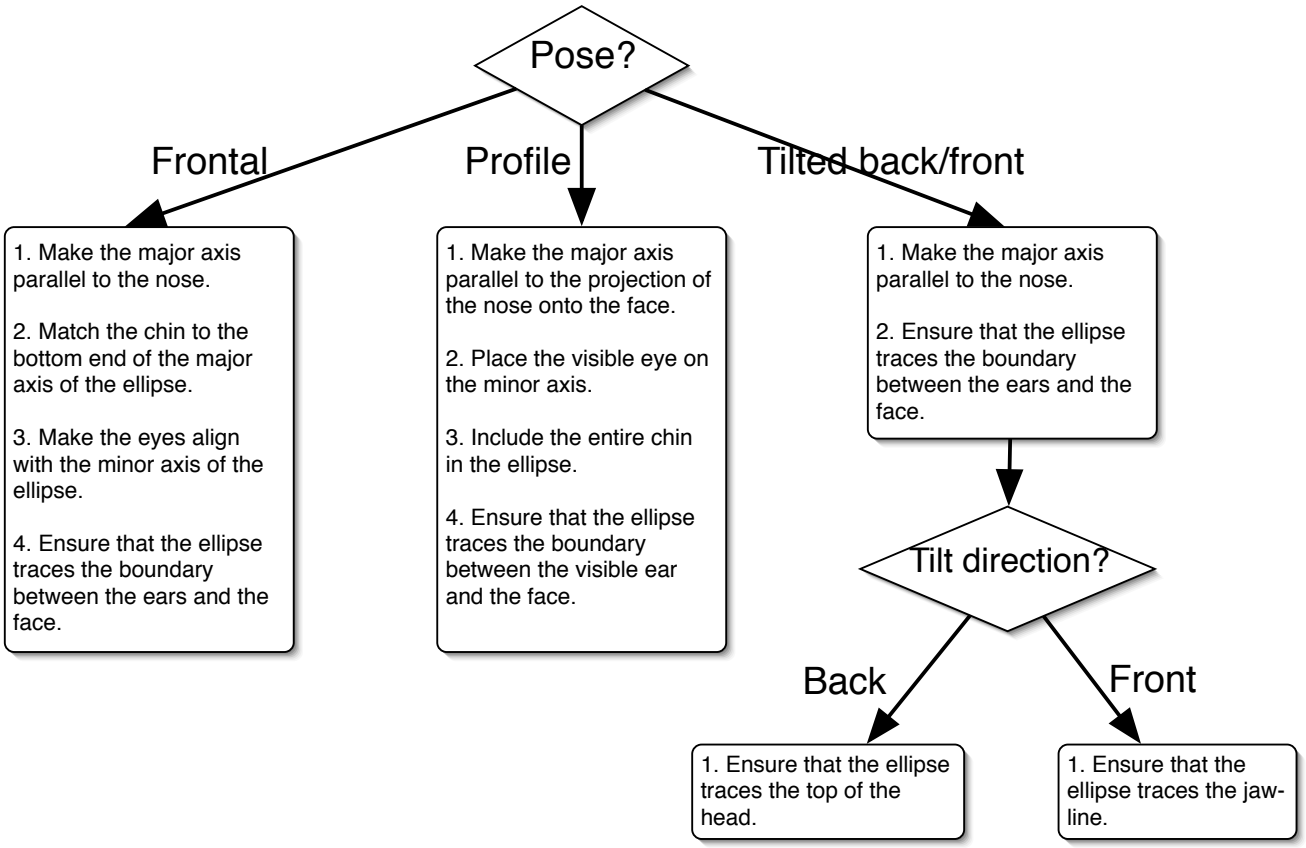


Figure 11. *Procedure for drawing ellipses around an average face region*. The annotators were instructed to follow this flowchart to draw ellipses around the face regions. The annotation steps are a little different for different poses. Here, we present the steps for three canonical poses: frontal, profile and tilted back/front. The annotators were instructed to use a combination of these steps for labeling faces with derived, intermediate head poses. For instance, to label a head facing slightly towards its right and titled back, a combination of the steps corresponding to the profile and tilted-back poses are used.

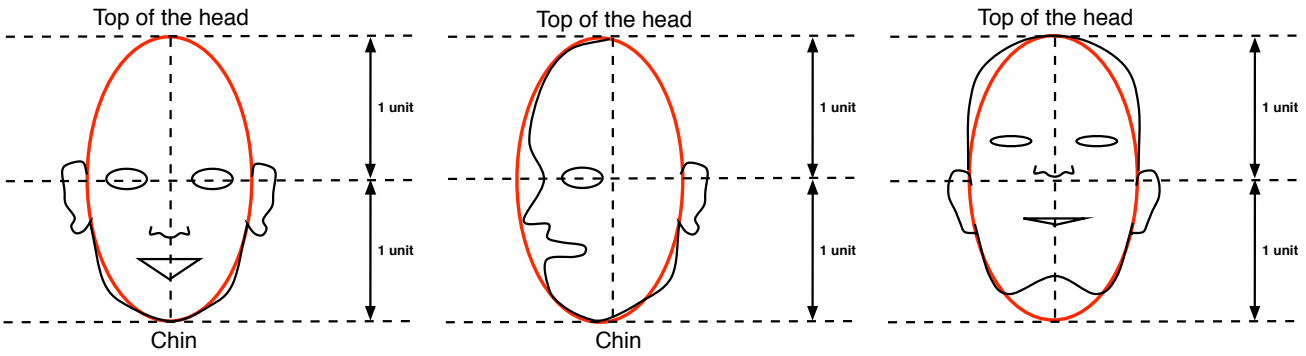


Figure 12. *Illustrations of ellipse labeling on line drawings of human head*. The black curves show the boundaries of a human head in *frontal* (**left**), *profile* (**center**), and *tilted-back* (**right**) poses. The red ellipses illustrate the desired annotations as per the procedure shown in Figure 11. Note that these head shapes are approximations to an average human head, and the shape of an actual human head may deviate from this mean shape. The shape of a human head may also be affected by the presence of factors such as emotions. The guidelines on annotating face regions influenced by these factors are specified in Appendix A.

- **Hair**. Ignore the hair and fit the ellipse around the hypothetical bald head.
- **Occlusion**. Hypothesize the full face behind the occluding object, and match all of the visible features.

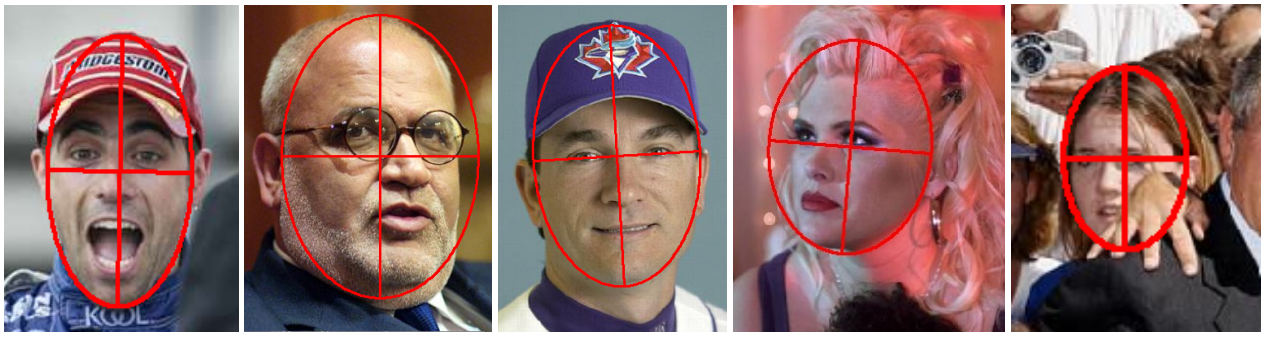


Figure 13. *Illustrations of labeling for complex face appearances*. These images show example annotations for human heads with shapes different from an average human head due to the presence of *facial expression*, *double chin*, *square jaw*, *hair-do*, and *occlusion*, respectively.

Figure 13 shows some example annotations for complex face shapes.

B. Data formats

The original set of images can be downloaded as `originalPics.tar.gz` from http://tamaraberg.com/faceDataset/. Uncompressing this tar-file organizes the images as `originalPics/year/month/day/big/*.jpg`.

The ten folds described in the EXP-1 experiments (Section 7) are available at http://vis-www.cs.umass.edu/fddb/FDDB-folds.tgz. Uncompressing the `FDDB-folds.tgz` file creates a directory `FDDB-folds`, which contains files with names: `FDDB-fold-xx.txt` and `FDDB-fold-xx-ellipseList.txt`, where `xx` = {`01, 02, ..., 10`} represents the fold-index.

Each line in the `FDDB-fold-xx.txt` file specifies a path to an image in the above-mentioned data set. For instance, the entry `2002/07/19/big/img_130` corresponds to `originalPics/2002/07/19/big/img_130.jpg`. The corresponding annotations are included in the file `FDDB-fold-xx-ellipseList.txt`. These annotations are specified according to the format shown in Table 1. Each of the annotation face regions are represented as an elliptical region, which is denoted by a 6-tuple

$$(r_a, r_b, \theta, c_x, c_y, 1), \tag{5}$$

where r_a and r_b refer to the half-length of the major and minor axes; θ is the angle of the major axis with the horizontal axis; and c_x and c_y are the x and y coordinates of the center of this ellipse.

The detection output should also follow the format described in Table 1. The representation of each of the detected face regions, however, could either be denoted using a rectangle or an ellipse. The exact specification for these two types of representations is as following:

- **Rectangular regions**

...
name of the i^{th} image
number of faces in the i^{th} image $= m$
face f_1
face f_2
...
face f_m
...

Table 1. *Format used for the specification of annotations and detections* .

Each face region is represented as a 5-tuple

$$(x, y, w, h, s), \tag{6}$$

where x, y are the coordinates of the top-left corner; w and h are the width and height; and $s \in \{-\infty, \infty\}$ is the confidence score associated with the detection of this rectangular region.

- **Elliptical regions**
  Each face region is represented as a 6-tuple

$$(r_a, r_b, \theta, c_x, c_y, s), \tag{7}$$

where r_a and r_b refer to the half-length of the major and minor axes; θ is the angle of the major axis with the horizontal axis; and c_x and c_y are the x and y coordinates of the center; and $s \in \{-\infty, \infty\}$ is the confidence score associated with the detection of this elliptical region.

Note that the order of images in the output file is expected to be the same as the order in the file `annotatedList.txt`.