A Fast and Accurate Unconstrained Face Detector

Article *in* IEEE Transactions on Pattern Analysis and Machine Intelligence · August 2014

DOI: 10.1109/TPAMI.2015.2448075 · Source: arXiv

Shengcai Liao
Chinese Academy of Sciences
68 PUBLICATIONS **4,432** CITATIONS
SEE PROFILE

Stan Z Li
Chinese Academy of Sciences
790 PUBLICATIONS **25,075** CITATIONS
SEE PROFILE

A Fast and Accurate Unconstrained Face Detector

Shengcai Liao, *Member, IEEE*, Anil K. Jain, *Fellow, IEEE*, and Stan Z. Li, *Fellow, IEEE*

Abstract—We propose a method to address challenges in unconstrained face detection, such as arbitrary pose variations and occlusions. First, a new image feature called Normalized Pixel Difference (NPD) is proposed. NPD feature is computed as the difference to sum ratio between two pixel values, inspired by the Weber Fraction in experimental psychology. The new feature is scale invariant, bounded, and is able to reconstruct the original image. Second, we learn the optimal subset of NPD features and their combinations via regression trees, so that complex face manifolds can be partitioned by the learned rules. This way, only a single cascade classifier is needed to handle unconstrained face detection. Furthermore, we show that the NPD features can be efficiently obtained from a look up table, and the detection template can be easily scaled, making the proposed face detector very fast (about 178 FPS for 640x480 resolution videos and 30 FPS for 1920x1080 resolution videos on a desktop PC, about 6 times faster than OpenCV). Experimental results on three public face datasets (FDDB, GENKI, and CMU-MIT) show that the proposed method outperforms the state-of-the-art methods in detecting unconstrained faces with arbitrary pose variations and occlusions in cluttered scenes.

Index Terms—Unconstrained face detection, normalized pixel difference, regression tree, AdaBoost, cascade classifier

1 Introduction

The objective of face detection is to find and locate faces in an image. It is the first step in automatic face recognition applications. Face detection has been well studied for frontal and near frontal faces. The Viola and Jones' face detector [1] is the most well known face detection algorithm, which is based on Haar-like features and cascade AdaBoost [2] classifier. However, in unconstrained scenes such as faces in a crowd, state-of-the-art face detectors fail to perform well due to large pose variations, illumination variations, occlusions, expression variations, out-of-focus blur, and low image resolution. For example, the Viola-Jones face detector fails to detect most of the face images in the Face Detection Data set and Benchmark (FDDB) database [3] (examples shown in Fig. 1) due to the difficulties mentioned above. In this paper, we refer to face detection with arbitrary facial variations as the unconstrained face detection problem. We are interested in face detection in unconstrained scenarios such as video surveillance or images captured by hand-held devices.

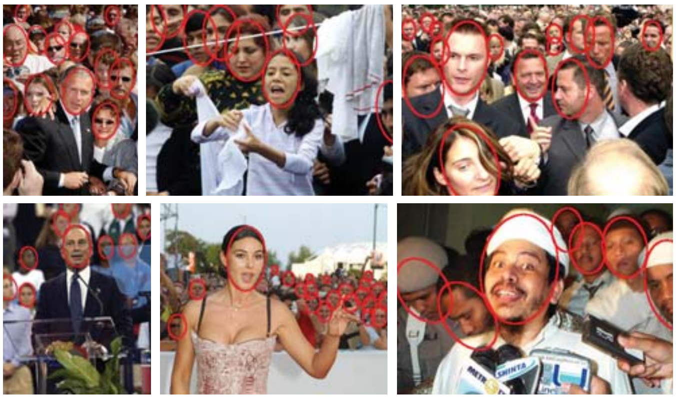


Fig. 1. Face images annotated (red ellipses) in the FDDB database [3].

Numerous face detection methods have been developed following Viola and Jones' work [1], mainly focusing on extracting different types of features and developing different cascade structures. A variety of complex features [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] have been proposed to replace the Haar-like features used in [1]. While these methods can improve the face detection performance to some extent, they generate a very large number (hundreds of thousands) of features and the resulting systems take too much time to train. Another development in face detection has been to learn different cascade structures for multiview face detection, such as parallel cascade [14], pyramid architecture [15], and Width-First-Search (WFS) tree [16]. All these methods need to learn one cascade classifier for each specific facial view (or view range). In unconstrained scenarios, however, it is not easy to define all possible views of a face, and the computational cost increases with an increasing number of classifiers in complex cascade

- *Shengcai Liao and Stan Z. Li are with the National Laboratory of Pattern Recognition and the Center for Biometrics and Security Research, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: {scliao,szli}@nlpr.ia.ac.cn*
- *Anil K. Jain is with the Dept. of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA. He is also affiliated with the Dept. of Brain & Cognitive Engineering, Korea University, Anamdong, Seongbukgu, Seoul 136-713, Republic of Korea. E-mail: jain@cse.msu.edu*

structure. Moreover, these approaches require manual labeling of face pose in each training image.

While some of the available methods [14], [15], [16] can handle multiview faces, they are not able to simultaneously consider other challenges such as occlusion. In fact, since these methods require partitioning multiview data into known poses, occlusion is not easy to handle in this way. On the other hand, while several studies addressed face detection under occlusion [17], [18], [19], [20], [21], they constrained themselves to detect only frontal faces under occlusion. As discussed in [22], a robust face detection algorithm should be effective under arbitrary variations in pose and occlusion, which remains an unresolved challenging problem.

In this paper, we are interested in developing effective features and robust classifiers for unconstrained face detection with arbitrary facial variation. First, we propose a simple pixel-level feature, called the Normalized Pixel Difference (NPD). An NPD is computed as the ratio of the difference between any two pixel intensity values to the sum of their values, in the same form as the Weber Fraction in experimental psychology [23]. The NPD feature has several desirable properties, such as scale invariance, boundedness, and ability to reconstruct the original image. we further show that NPD features can be obtained from a look up table, and the resulting face detection template can be easily scaled for multiscale face detection.

Secondly, we develop a method to construct a single cascade AdaBoost classifier that can effectively deal with complex face manifolds and handle arbitrary pose and occlusion conditions. While the individual NPD feature may have "weak" discriminative ability, our work indicates that a subset of NPD features can be optimally learned and combined to construct more discriminative features in a regression tree. In this way, different types of faces can be automatically divided into different leaves of a regression tree, and the complex face manifold in high dimensional space can be partitioned in the learning process. This is a "divide and conquer" strategy to tackle unconstrained face detection in a single classifier, without pre-labeling of views in the training set of face images. The resulting face detector is robust to variations in pose, occlusion, and illumination, as well as to blur and low image resolution.

The novelty of this work is summarized as follows:

- A new type of feature, called NPD is proposed, which is efficient to compute and has several desirable properties, including scale invariance, boundedness, and enabling reconstruction of the original image.
- A subset of NPD features is automatically learned and combined in regression trees to boost their discriminability. In this way, only a single cascade AdaBoost classifier is needed to handle unconstrained faces with occlusions and arbitrary viewpoints, without pose labeling or clustering in the training stage.

The advantages of the proposed approach include:

- The NPD feature evaluation is extremely fast, requiring a single memory access using a look up table.
- Multiscale face detection can be easily achieved by applying pre-scaled detection templates.
- The unconstrained face detector does not depend on pose specific cascade structure design; pose labeling or clustering in the training stage is also not required.
- The face detector is able to handle illumination variations, pose variations, occlusions, out-of-focus blur, and low resolution face images in unconstrained scenarios.

The remainder of this paper is organized as follows. In Section 2 we review the related work. In Section 3 we introduce the NPD feature space. The proposed NPD based face detection method is presented in Section 4. Experimental results are provided in Section 5. Finally, we summarize the contributions in Section 6.

2 Related work

As indicated in a survey of face detection methods [24], the most popular face detection methods are appearance based, which use local feature representation and classifier learning. Viola and Jones' face detector [1] was the first one to apply rectangular Haar-like features in a cascaded AdaBoost classifier for real-time face detection. Many approaches have been proposed around the Viola-Jones detector to advance the state of the art in face detection. Lienhart and Maydt [4] proposed an extended set of Haar-like features, where 45° rotated rectangular features were introduced. Li et al. [5] proposed another extension of Haar-like features, where the rectangles can be spatially set apart with a flexible distance. A similar feature, called the diagonal filter was also proposed by Jones and Viola [6]. Various other local texture features have been introduced for face detection, such as the modified census transform [7], local binary pattern (LBP) [8], MB-LBP [11], LBP histogram [10], and the locally assembled binary feature [12]. These features have been shown to be robust to illumination variations. Mita et al. [9] proposed the joint Haar-like features to capture the co-occurrence of effective Haar-like features. Huang et al. [16] proposed a sparse feature set in a granular space, where granules were represented by rectangles, and each individual sparse feature was learned as a combination of granules. A problem with the approaches in [9] and [16] is that the joint feature space is very large, making the optimal combination a difficult task.

While more sophisticated features may provide better discrimination power than Haar-like features for the face detection task, they generally increase the computational cost. In contrast, ordinal relationships among image regions are simple yet effective image features [25], [26], [27], [28], [29], [30]. Sinha [25] studied several robust ordinal relationships in face images and developed a face detection method accordingly. Liao et al. [28] further showed that ordinal features can be effectively learned by AdaBoost classifier for face recognition. Sadr et al. [26] showed that pixelwise ordinal features (ordinal relationship between any two pixels) can faithfully encode image structures. Baluja et al. [27] showed that simple pixelwise ordinal features are good enough for discriminating between five facial orientations, a relatively simpler task than face detection. Wang et al. [30] applied the random forest classifier together with pixelwise ordinal features for facial landmark localization. Abramson and Steux [29] proposed a pixel control point based feature for face detection, where each feature is associated with two sets of pixel locations (control points). However, it is not easy to learn control point based features because of the huge number of control point combinations.

Besides different feature representations, some researchers have also tried different AdaBoost algorithms and weak classifiers. For weak classifiers utilized in boosting, Lienhart et al. [31] and Brubaker et al. [32] have shown that classification and regression trees (CART) [33] work better than simple decision stumps. In this paper, we show that the optimal ordinal features and their combinations can be learned by integrating the proposed NPD features in a regression tree. In this way, unconstrained face variations can be automatically partitioned into different leaves of the learned regression tree.

Given that the original Viola-Jones face detector has limitations for multiview face detection [24], various cascade structures have been proposed to tackle multiview face detection [6], [14], [15], [16]. Jones and Viola [6] extended their face detector by training one face detector for each specific pose. To avoid evaluating all face detectors on each scanning subwindow, they developed a pose estimation step (similar to Rowley et al. [34]) before face detection, and then only the face detector trained on that estimated pose was applied. In this two-stage detection structure, if the pose estimation is not reliable, the face is not likely to be detected in the second stage. Wu et al. [14] proposed a parallel cascade structure for multiview face detection, where all face detectors tuned to different views have to be evaluated for each scanning window; they did use the first few cascade layers of all face detectors to estimate the pose for speedup. Li and Zhang [15] proposed a coarse-to-fine pyramid architecture for multiview face detection, where the entire range of face poses was divided into increasingly smaller subranges, resulting in a more efficient detection structure. Huang et al. proposed a WFS tree based multiview face detection approach, which also works in a coarse-to-fine manner. They proposed the Vector Boost algorithm for multiclass learning, which is well suited for multiview pose estimation. However, all these methods need to learn a cascade classifier for each specific view (or view range) of a face, requiring an input face image to go through different branches of the detection structure. Hence, their computational cost generally increases with the number of classifiers in complex cascade structures. Moreover, these approaches require manual labeling of the face pose in each training image.

Instead of designing a detection structure, Lin and Liu [19] proposed to learn the multiview face detector as a single cascade classifier. They derived a multiclass boosting algorithm, called MBHBoost by sharing features among different classes. This is a simpler approach to multiview face detection than designing complex cascade structures. Nevertheless, it still requires manual labeling of poses. In uncontrolled environments, however, it is not easy to define specific views of a face by discretizing the pose space, because a face could be in arbitrary pose simultaneously in yaw (out-of-plane), roll (in-plane), and pitch (up-and-down) angles. To avoid manual labeling, Seemann et al. [35] suggested learning viewpoint clusters automatically for object detection. However, for human faces, Kim and Cipolla [36] showed that clustering by traditional techniques like K-Means does not result in categorized poses. They hence proposed a multi-classifier boosting (MCBoost) for human perceptual clustering of object images, which showed promise for clustering face poses. However, the clusters are not always related to pose variations; in addition to different pose clusters, they also obtained clusters with various illumination variations.

Face detection in presence of occlusion is also an important issue in unconstrained face detection, but it has received less attention compared to multiview face detection. This is probably because, compared to pose variations, it is more difficult to categorize arbitrary occlusions into predefined classes. Hotta [17] proposed a local kernel based SVM method for face detection, which was better than global kernel based SVM in detecting occluded frontal faces. Lin et al. [18] considered 8 kinds of manually defined facial occlusions by training 8 additional cascade classifiers besides the standard face detector. Lin and Liu [19] further proposed the MBHBoost algorithm to handle faces with one of 12 in-plane rotations or one of 8 types of occlusions, with each kind of rotation and occlusion treated as a different class. Chen et al. [20] proposed a modified Viola-Jones face detector, where the trained detector was divided into sub-classifiers related to several predefined local patches, and the outputs of sub-classifiers were fused. Goldmann et al. [21] proposed a component-based approach for face detection, where the two eyes, nose, and mouth were detected separately, and further connected in a topology graph. However, none of the above methods considered face detection with both occlusions and pose variations simultaneously in unconstrained scenarios. As discussed in [22], a robust face detector should be effective under arbitrary variations in pose and occlusion, which has not yet been solved.

Recently, unconstrained face detection has gained attention. Jain and Learned-Miller [3] developed the FDDB database and benchmark for the development of unconstrained face detection algorithms. This database contains images collected from the Internet, and presents challenging scenarios for face detection. Subburaman and Marcel [37] proposed a fast bounding box estimation technique for face detection, where the bounding box is predicted by small patch based local search. Jain and Learned-Miller [38] proposed an online domain adaption approach to improve the performance of the Viola-Jones face detector on the FDDB database. Li et al. [13] proposed the use of SURF feature [39] in an AdaBoost cascade, and area under the curve (AUC) criterion to speed up the face detector training. Zhu and Ramanan [40] proposed to jointly detect face, estimate pose, and localize face landmarks in the wild. Shen et al. [41] proposed an exemplar-based face detection approach, which retrieves images from a large annotated face dataset; facial landmark locations are inferred from the annotations. Li et al. [42] proposed a probabilistic elastic part (PEP) model to adapt any pre-trained face detector to a specific image collection like FDDB. This method extracts the PEP representation for each candidate face detected by a general face detector, and trains a classifier with the top positive and negative samples. Despite the availability of these methods for unconstrained face detection, the detection accuracy is still not satisfactory, especially when the detector is required to have low false alarms.

3 Normalized Pixel Difference Feature Space

The Normalized Pixel Difference (NPD) feature between two pixels in an image is defined as

$$f(x, y) = \frac{x - y}{x + y}, \tag{1}$$

where $x, y \geq 0$ are intensity values of the two pixels[1], and $f(0, 0)$ is defined as 0 when $x = y = 0$.

The NPD feature measures the relative difference between two pixel values. The sign of $f(x, y)$ indicates the ordinal relationship between the two pixels x and y , and the magnitude of $f(x, y)$ measures the relative difference (as a percentage of the joint intensity $x+y$) between x and y. Note that the definition $f(0, 0) \triangleq 0$ is reasonable because, in this case, there is no difference between the two pixels x and y. Compared to the absolute difference $|x - y|$, NPD is invariant to scale change of the pixel intensities.

Weber, a pioneer in experimental psychology, stated that the just-noticeable difference in the magnitude change of a stimulus is proportional to the magnitude of the stimulus, rather than its absolute value [23]. This is known as the Weber's Law. In other words, the human perception of difference in stimulus is often measured as a fraction of the original stimulus, that is, in a form $\Delta I/I$, which is called the Weber Fraction. Chen et al. [43] proposed a local image descriptor, called Weber's Law Descriptor for face recognition, which was computed from Weber Fractions of pixels in a 3×3 window. The proposed feature in Eq. (1) has also been used in other fields such as remote sensing, where the Normalized Difference Vegetation Index (NDVI) [44] is defined as the difference to sum ratio between the visible red and the near infrared spectra to estimate the green vegetation coverage.

The NPD feature has a number of desirable properties. First, the NPD feature is antisymmetric, so either $f(x, y)$ or $f(y, x)$ is adequate for feature representation, resulting in a reduced feature space. Therefore, in an $s \times s$ image patch (vectorized as $p \times 1$, where $p = s \cdot s$), NPD feature $f(x_i, x_j)$ for pixel pairs $1 \leq i < j \leq p$ is computed, resulting in $d = p(p-1)/2$ features. For example, in a 20×20 face template, there are $(20 \times 20) \times (20 \times 20 - 1)/2 = 79,800$ NPD features in total. We call the resulting feature space the NPD feature space, denoted as Ω_{npd} ($\in \mathbb{R}^d$).

Second, the sign of $f(x, y)$ is an indicator of the ordinal relationship between x and y. Ordinal relationship has been shown to be an effective encoding for object detection and recognition [25], [26], [28] because ordinal relationship encodes the intrinsic structure of an object image and it is invariant under various illumination changes [25]. However, simply using the sign to encode the ordinal relationship is likely to be sensitive to noise when x and y have similar values. In the next section we will show how to learn robust ordinal relationships with NPD features.

Third, the NPD feature is scale invariant, which is expected to be robust against illumination changes. This is important for image representation, since illumination change is always a troublesome issue for both object detection and recognition.

Fourth, as shown in Appendix A, the NPD feature f(x,y) is bounded in [-1,1]. The bounded property makes the NPD feature amenable to histogram binning or threshold learning in tree-based classifiers [1]. Fig. 2 shows that $f(x, y)$ is a bounded function and it defines a nonlinear surface.

Theorem 1 (Reconstruction): Given the NPD feature vector $\mathbf{f} = (f(x_1, x_2), f(x_1, x_3), \ldots, f(x_{p-1}, x_p))^T \in \Omega_{npd}$, the original image intensity values $I = (x_1, x_2, \ldots, x_p)^T$ can be reconstructed up to a scale factor.

[1] For ease of representation, sometimes we also denote x and y as pixels instead of pixel values. We use subscripts to differentiate between pixel and pixel values only when pixel locations are under discussion.

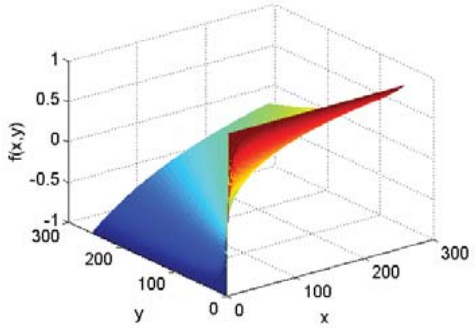


Fig. 2. A plot of the NPD function $f(x, y)$.

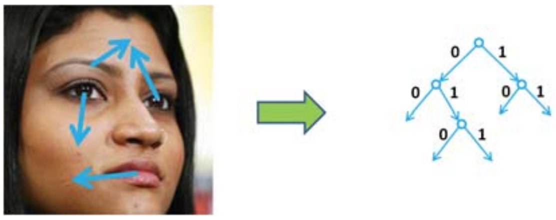


Fig. 3. Learning and combining ordinal features in a regression tree. Left: four pixelwise ordinal features are automatically selected in the learning process. Right: the four features are optimally combined in a regression tree for face/nonface prediction.

The proof of Theorem 1 is shown in Appendix B, which also gives a linear-time approach to reconstruct the original image up to a scale factor. Theorem 1 states that each point in the feature space Ω_{npd} corresponds to a group of intensity-scaled images in the original pixel intensity space. In contrast, the scale invariance property says that all intensity-scaled images are "compressed" to a point in the bounded feature space Ω_{npd}. Therefore, Ω_{npd} is a feature space which is invariant to scale variations, but it carries all the necessary information from the original space.

4 NPD FOR FACE DETECTION

4.1 Learning Object Structures

Ordinal relationship [25] is a well-known simple and basic concept: it compares the brightness of any two image regions, and encodes the result with 1 (brighter) or 0 (darker) accordingly. Sinha [25] showed that ordinal features can represent the intrinsic structure of objects such as a human face, and they are insensitive to illumination changes. Instead of encoding ordinal relationship between two image regions, in this paper, we learn robust ordinal relationships between pairs of pixels via the NPD feature. For a face pattern which is well structured, automatically learned combinations of ordinal features may represent a face better than manual configurations. Therefore, we propose to learn a combination of simple ordinal features by boosted regression trees [33]. By providing a training set of face and nonface images, a weak classifier is learned by a regression tree. At each node, the tree checks the optimal ordinal feature value, and then passes the input data to the next branch accordingly. See Fig. 3. Regression tree is also well suited for face detection with arbitrary pose variations, since similar views can be clustered in the same leaf node of the tree.

Ordinal relationship can always be generated by the default threshold of 0, but it will be sensitive to noise especially when the two pixels to be compared have similar values. In this paper, we learn robust ordinal relationships and their combinations by learning regression trees with NPD features. In this way, regression trees not only learn optimal NPD features at each branch node, but also learn optimal thresholds for splitting. Generally, one of the following two cases are leaned for each NPD feature at a branch node:

$$f(x, y) = \frac{x - y}{x + y} < \theta_1 < 0, \tag{2}$$

$$f(x, y) = \frac{x - y}{x + y} \geq \theta_2 > 0, \tag{3}$$

where θ_1 and θ_2 are the thresholds. Eq. (2) applies if the object pixel x is notably darker than pixel y, while Eq. (3) covers the case when pixel x is notably brighter than pixel y. The learned thresholds allow the ordinal encodings in the learned regression trees to represent the intrinsic object structure. To learn such regression trees, we use the CART algorithm [33] with the NPD features.

4.2 Face Detector

Given that the proposed NPD features contain redundant information, we also apply the AdaBoost algorithm to select the most discriminative features and construct strong classifiers [1]. We adopt the Gentle AdaBoost algorithm [2] to learn the NPD feature based regression trees.

As in [1], a cascade classifier is further learned for rapid face detection. We only learn one single cascade classifier for unconstrained face detection robust to occlusions and pose variations. This implementation has the advantage that there is no need to label the pose of each face image manually or cluster the poses before training the detector. In the learning process, the algorithm automatically divides the whole face manifold into several sub-manifolds by regression trees.

Below is a summary of how the proposed method handles the unconstrained face detection problem.

- **Pose**. Pose variations are handled by learning NPD features in boosted regression trees, where different views can be automatically partitioned into different leaves of the regression trees.
- **Occlusion**. In contrast to Haar-like features that are sensitive to occlusions because of large support [18], NPD features are computed by only

two pixel values, making them robust to occlusion.

- **Illumination**. Since NPD features are scale invariant, they are robust to illumination changes.
- **Blur or low image resolution**. Because the NPD features involve only two pixel values, they do not require rich texture information on the face. This makes NPD features effective in handling blurred or low resolution face images.

4.3 Speed Up

To further speed up the proposed NPD face detector, we develop the following two techniques. First, for 8-bit gray images, we build a 256×256 look up table to store pre-computed NPD features. This way, computing $f(x, y)$ in Eq. 1 only requires one memory access from the look up table.

Second, the learned face detection template (e.g. 20×20 used in this paper) can be easily scaled to enable multiscale face detection. So, we pre-compute multiscale detection templates and apply them to detect faces at various scales. This way, iterative rescaling of images for multiscale detection is avoided.

5 Experiments

We evaluate the performance of the NPD face detector on three public-domain databases, FDDB [3], GENKI [45], and CMU-MIT [34]. We also provide an analysis of the proposed method, report the face detection speed, and report unconstrained face detection performances under illumination variations, pose variations, occlusion, and blur, respectively.

5.1 Implementation of NPD Face Detector

A subset of the training data[2] in [13] was used to train our detector, including 12,102 face images and 12,315 nonface images (some private face images and the Corel5k nonface images were not available, so they could not be used). Fig. 4 shows some example face and nonface images from this training dataset. The detection template is 20×20 pixels. The detector cascade contains 15 stages, and for each stage, the target false accept rate was 0.5, with a detection rate of 0.999. For the depth of the regression tree, we set a constraint that each leaf node must contain at least (1/16)th of the total number of training samples. Under this constraint, the tree depth is at most 5, and in the test phase at most 4 NPD features need to be computed for each regression tree. The first five stages of our detector include 3, 4, 6, 7, 9 weak classifiers, respectively. Fig. 5 shows the NPD features learned in the three regression trees in the first stage. It can be observed that most of the learned features are around eyes, eyebrows, and nose. In addition, the features in the three regression trees are distributed in different parts of the facial region. This is because, in the boosting scheme, all samples are reweighted when a weak classifier is learned, so that the next weak classifier can focus on the training samples that can not be correctly classified in the current step. The face shown in Fig. 5 is a frontal face, but it should be kept in mind that the face can have arbitrary pose variations, and some learned features may be only effective for a specific pose.

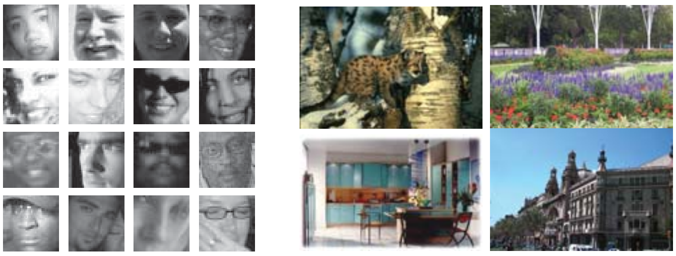


Fig. 4. Example face (left) and nonface (right) images from [13] for face detector training.

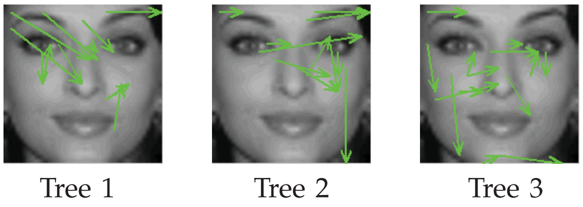


Fig. 5. The learned NPD features by boosting regression trees in the first stage of the cascade.

In the test stage, a scale factor of 1.2 was set for multiscale detection. A postprocessing method similar to the OpenCV face detection module was implemented, which merges nearby detections by the disjoint set algorithm. For each detected face, we summarized the scores of AdaBoost classifiers in all stages of the cascade to be the final score; this score was used to generate the Receiver Operating Characteristic (ROC). We used three public face databases, FDDB [3], GENKI [45], and CMU-MIT [34], to evaluate our face detection algorithm.

5.2 Evaluation on FDDB Database

The FDDB dataset [3] covers challenging scenarios for face detection. Images in FDDB comes from the Faces in the Wild dataset [46], which is a large collection of Internet images collected from the Yahoo News. It contains 2,845 images with a total of 5,171 faces, with a wide range of challenging scenarios including arbitrary pose, occlusions, different lightings, expressions, low resolutions, and out-of-focus faces. All faces in the database have been annotated with elliptical regions. Fig. 1 shows some examples of the annotated faces from the FDDB database.

For benchmark evaluation, Jain and Learned-Miller [3] provided an evaluation code for a compari-

[2] https://sites.google.com/site/leeplus/publications/facedetectionusingsurfcascade

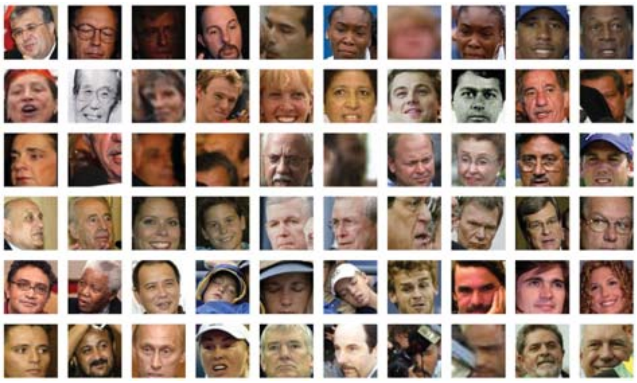


Fig. 6. Face images cropped from the FDDB database [3].

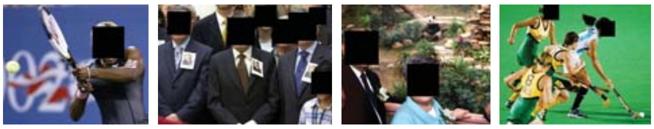


Fig. 7. Modified images from the FDDB database [3] for bootstrapping nonface samples.

son of different face detection algorithms. There are two metrics for performance evaluation based on ROC: discrete score metric and continuous score metric, which correspond to coarse match (similar to previous evaluations in the face detection literature) and precise match, respectively, between the detection and the ground truth. Two experimental setups are proposed in [3]. The first experiment (EXP-1) requires a 10-fold cross-validation, while the second experiment (EXP-2) allows unrestricted training, which means that images outside FDDB can be used for face detector training.

We followed both experimental protocols. For EXP-1, we trained 10 face detectors, with the same settings described in Section 5.1, and tested them separately using 10-fold cross-validation. On average, we used about 4,500 face images annotated in FDDB to train a single face detector. Fig. 6 shows some face images that were cropped from the FDDB database for training our face detectors. Since FDDB does not provide a set of nonface images, we replaced all annotated face regions with black patches in the FDDB images and then used the resulting images to bootstrap nonface samples. Fig. 7 illustrates such modified images.

For EXP-2, we used the detector trained with data outside FDDB, as described in the previous subsection. For evaluation, this detector was applied on each subset of the FDDB database separately, and an average performance is reported.

We compared our method with state-of-the-art results reported on the FDDB website[3]. The ROC curves of various algorithms are depicted in Fig. 8 for the discrete score metric and in Fig. 9 for the continuous score metric. Note that all the baseline results are for EXP-2, because we did not find any result following the EXP-1 protocol. In both Figs. 8 and 9, the curve labels in the legend are sorted in descending order of the detection rates at zero false positives (FP=0). Note also that, on average, FP=285 generally means one false detection per image for the FDDB experiments. Therefore, the useful FPs are in the range [0,500]; we show the X axis in logarithmic scale to emphasize the performance at low FPs. Among the baseline methods, “Olaworks Inc.” and “Illuxtech Inc.” are two commercial detectors. Their methods, as well as the method of “Shenzhen University”, have not been published. “SURF Cascade” is the SURF descriptor based cascade method proposed by Li et al. in [13], which is the best published result at low false positives to date. The method of Zhu-Ramanan [40] was evaluated by the FDDB team, and the result, reported on the FDDB website, is now the state of the art among published methods. For the proposed NPD face detector, besides scaling the detection template in a nearest neighbor fashion, we also tried building the image pyramid representation by the default *imresize* function in MATLAB, and applied the 20×20 detection template. Since this function uses the bicubic interpolation method with antialiasing, we call the resulting detector “Smooth NPD”.

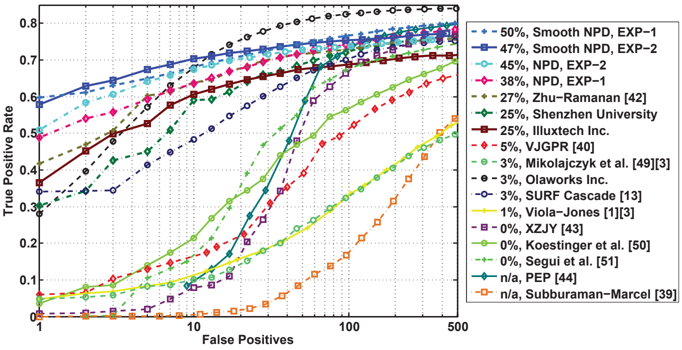


Fig. 8. ROC curves for face detection on the FDDB database [3] with the discrete score metric.

From the discrete score metric results shown in Fig. 8, it can be observed that the proposed method

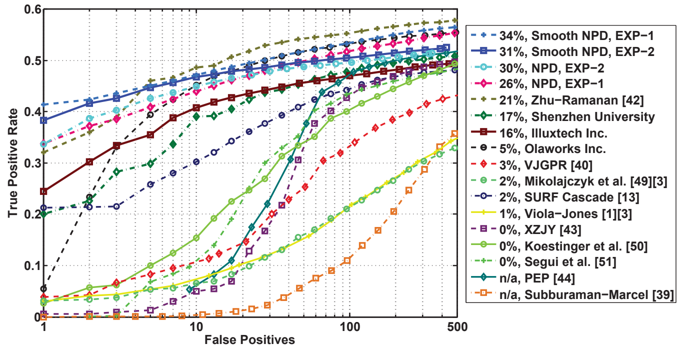


Fig. 9. ROC curves for face detection on the FDDB database [3] with the continuous score metric.

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

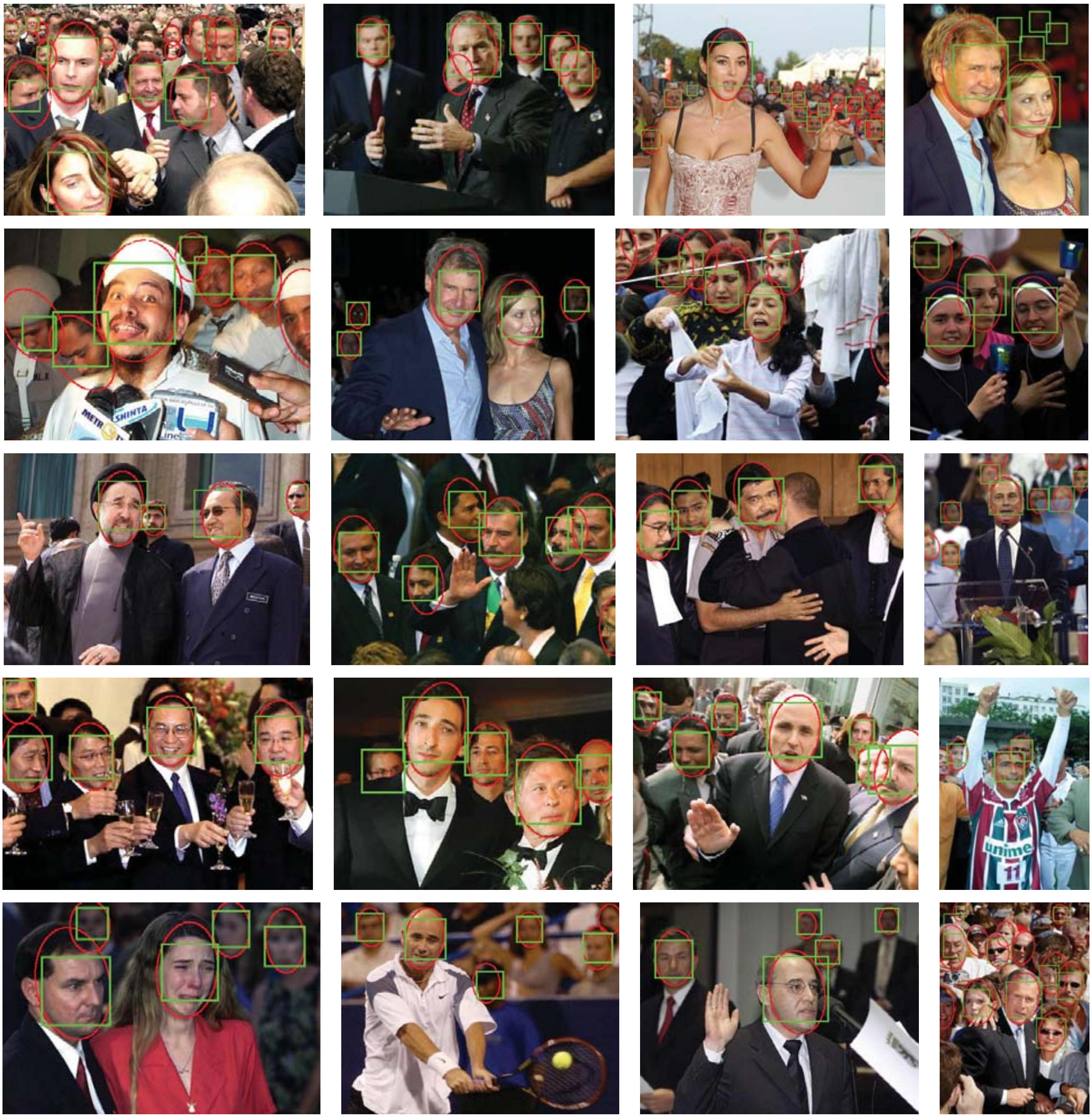


Fig. 10. Detected faces in the FDDB database [3] by the proposed NPD method. Green boxes are detections by the NPD detector, while red ellipses are ground truth annotations.

outperforms all the baseline methods except Olaworks Inc. However, the proposed NPD detector is much better than Olaworks' detector when FP $<$ 10. In fact, when FP=0 (shown in the legend), the proposed NPD detector detects 45% of the annotated FDDB faces in coarse sense (50% overlap with ground truth), while the detection rates of all baseline detectors are below 30%. Note that with a sub training set that was previously used for SURF Cascade [13], NPD for EXP-2 shows much better performance than SURF Cascade. Further, the Smooth NPD is slightly better than NPD, but with an additional cost of smoothing computation. It is also observed that the results of the NPD detectors trained for EXP-1 and EXP-2 are comparable, though the training data size for EXP-2 is several times larger than that for EXP-1. This result indicates that FDDB contains representative images for unconstrained face detection. However, it is not easy to handle all this data in training a single detector (recall the large variations in face appearance in Fig. 6). Note that the generic NPD features are learned in regression trees to divide and conquer the complex face manifolds.

Similar observations can be found in Fig. 9 for the continuous score metric, except that Zhu-Ramanan is slightly better than the proposed method when FP$>$ 5,

TABLE 1
Comparison of detection rates (%) with both discrete and continuous metrics for EXP-2 on the FDDB database [3]*

	Discrete Metric			Continuous Metric		
	FP = 0	FP = 10	FP = 100	FP = 0	FP = 10	FP = 100
Smooth NPD	**47.23**	**70.41**	75.38	**31.26**	46.78	50.60
NPD	45.32	67.47	73.72	29.99	44.95	49.63
Zhu-Ramanan [40]	27.38	63.88	73.08	21.25	**48.62**	**55.40**
Shenzhen University	24.87	59.06	72.50	16.51	39.12	48.05
Illuxtech Inc.	24.56	60.55	68.86	16.50	40.82	47.01
VJGPR [38]	4.58	15.76	51.00	2.95	10.20	33.16
Mikolajczyk et al. [47] [3]	3.25	10.23	33.28	2.10	6.61	21.67
Olaworks Inc.	2.94	67.84	**82.58**	4.79	45.18	53.34
SURF Cascade [13]	2.59	48.27	70.01	1.60	30.21	44.36
Viola-Jones [1] [3]	1.39	10.02	32.64	0.90	6.48	21.26
XZJY [41]	0.31	7.91	67.51	0.19	4.99	43.40
Koestinger et al. [48]	0.19	21.47	57.03	0.14	15.38	40.55
Segui et al. [49]	0.00	15.08	67.94	0.00	9.78	43.76
PEP [42]	n/a	8.43	73.35	n/a	5.38	47.30
Subburaman-Marcel [37]	n/a	0.54	17.25	n/a	0.36	11.27

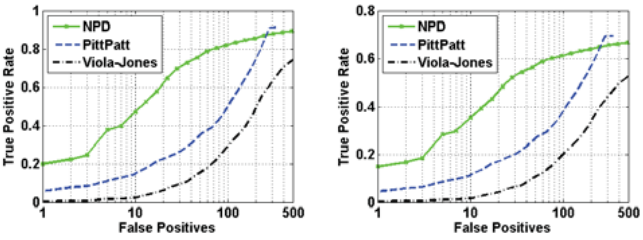
[*] Red numbers represents the best results, while blue numbers are the second best results. Results for Mikolajczyk et al. [47] and Viola-Jones [1] are reported in [3]. Results for Zhu-Ramanan [40] are evaluated by the FDDB team and reported on their website.

and "Smooth NPD, EXP-1" outperforms Olaworks Inc. Table 1 shows a comparison of detection rates for EXP-2 on the FDDB database at FP=0, 10, and 100. It is promising that at low false positives, the proposed method is either much better than the baseline methods, or comparable to the best performers.

Fig. 10 shows some examples of detected faces in the FDDB database by the proposed NPD method. Rotated, occluded, and out-of-focus faces can be successfully detected by the proposed method as shown in Fig. 10. Some occluded faces (e.g. 4th row and 2nd column in Fig. 10) and blurred faces (e.g. top-right image in Fig. 10) that are not annotated in the ground truth can still be detected by the proposed method. However, there are a number of faces that cannot be detected by the proposed method, especially in very crowded scenes (see the 1st image and the 3rd image in row 1, and the bottom-right image in Fig. 10).

5.3 Evaluation on GENKI Database

The GENKI database [45] was collected by the Machine Perception Laboratory, University of California, San Diego. We evaluated the current release of the GENKI database, GENKI-R2009a, on its SZSL subset, which contains 3,500 images collected from the Internet. These images include a wide range of backgrounds, illumination conditions, geographical locations, personal identity, and ethnicity. Some examples of face images from the GENKI database are shown in Fig. 12, with labeled detections by the proposed NPD method. Most images in the GENKI dataset contain only one face. In that sense, the GENKI dataset is not as challenging as the FDDB dataset. Some of the images in the GENKI-SZSL dataset contain faces that are not labeled, therefore they are not suitable for the face detection evaluation task. After removing such unlabeled images, we are left with 3,270 images for face detection evaluation. For performance evaluation, it is not fair to apply the learned detector described in Section 5.1, because the training data used for that detector contained face images from the GENKI database[4]. Therefore, we used the NPD face detector trained on the first fold of the FDDB 10-fold cross validation to evaluate the GENKI database. We also evaluated the Viola-Jones face detector implemented in OpenCV 2.4, and a commercial face detector PittPatt [50]. We again used the benchmark evaluation code by in [3] for performance evaluation, but slightly modified the code for allowing ground truth annotations as rectangles. The ROC curves of the three methods are shown in Fig. 11 for both the discrete and continuous score metrics. The results show that the proposed NPD face detector performs much better than both the Viola-Jones and PittPatt face detectors.



(a) discrete (b) continuous

Fig. 11. ROC curves for face detection on the GENKI-SZSL dataset [45] with (a) discrete and (b) continuous score metrics.

5.4 Evaluation on CMU-MIT Database

The CMU-MIT face dataset [34] is one of the early benchmark for face detection. The CMU-MIT frontal face data set contains 130 gray-scale images with a total of 511 faces, most of which are not occluded. We applied the same NPD detector described in Subsection 5.1 on this database. We also used the modified benchmark evaluation code from [3] with the discrete score metric for performance evaluation. Fig. 13 shows the ROC curves for the proposed NPD face detector, the Soft cascade method [51], the SURF cascade method [13], and the Viola-Jones detector [1]. The results show that, compared to the Viola-Jones frontal face detector, the NPD detector performs better when the number of false positives, $FP < 50$, while it is slightly worse than Viola-Jones at higher FPs. Compared to the SURF cascade detector, the NPD detector is better when $FP < 3$, but SURF cascade method outperforms NPD at higher FPs. Note that

[4] This training data is provided by the authors of [13]. We cannot remove the GENKI face images from this training data, because we have access to only the raw face images in binary format; we do not know the corresponding filenames and sources.

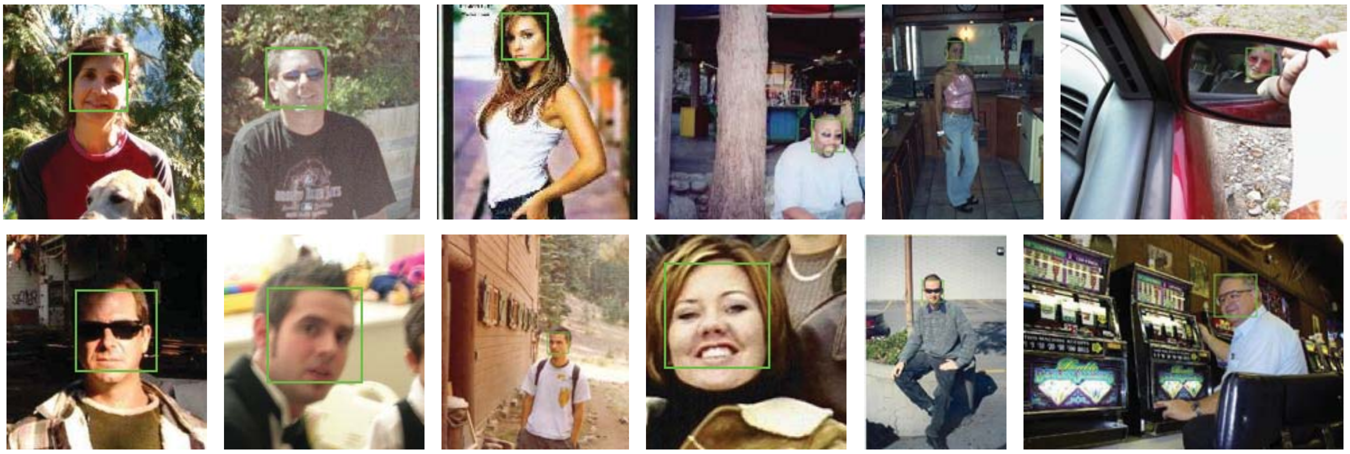


Fig. 12. Detected faces in the GENKI-SZSL dataset [45] by the proposed NPD method.

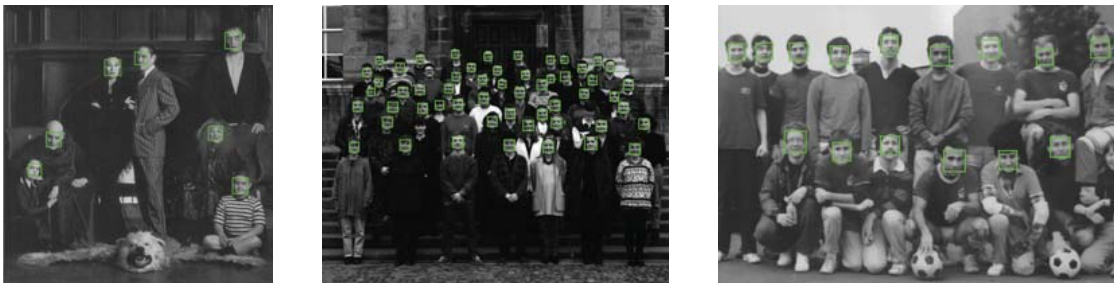


Fig. 14. Detected faces in the CMU-MIT dataset [34] by the proposed NPD method.

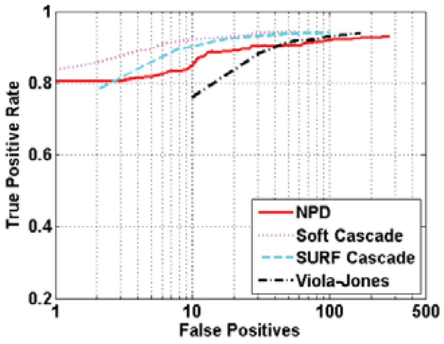


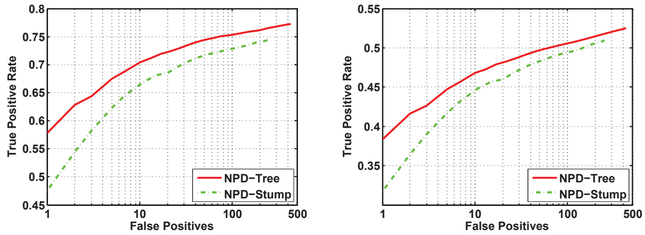
Fig. 13. ROC curves for face detection on the CMU-MIT dataset [34].

the SURF cascade method uses a face template of size 40×40 pixels, which is four times larger than our face detection template (20×20 pixels). Generally, a larger face template contains more features for face description, but is computationally more expensive and may have a limitation in detecting blurred faces. In addition, the proposed NPD method is not as good as the Soft cascade, the state-of-the-art method on the CMU-MIT dataset. Still, the proposed NPD method can detect about 80% of the frontal faces without any false positives, which is promising since we did not train a frontal face detector. Some of the detected faces in the CMU-MIT dataset by the proposed NPD method are shown in Fig. 14.

5.5 Analysis of the Proposed Face Detector

Since the proposed face detector is a combination of regression trees and the NPD features, it is instructive to determine the contribution of each of these two components. In the following, we trained all compared face detectors on the same training set and cascade training settings described in Section 5.1.

First, we trained a face detector based on the NPD features, but with the stump classifier [1], a basic tree classifier with only one splitting node. As shown in Table 2, the stump classifier based detector contains 1,597 weak classifiers. In contrast, the regression tree based detector contains 176 weak classifiers, indicating that combining NPD features in a regression tree is much more effective in constructing a weak classifier for AdaBoost learning. Furthermore, in cascade processing, each scanning subwindow needs to evaluate 36.5 NPD features, on average, for the stump classifier based detector. On the other hand, for the regression tree based detector, only 34.4 NPD features, on average, need to be evaluated, which means that using regression tree does not increase the average computation cost. The face detectors based on the stump classifier and the regression tree were tested on the FDDB database. The ROC curves of these two detectors are shown in Fig. 15 for both the discrete score metric and continuous score metric. As illustrated, using regression trees instead of stump classifier improves the face detection performance by about 2% – 10% for discrete metric and 1% – 7%



(a) discrete (b) continuous

Fig. 15. Comparison of NPD face detectors based on stumps and regression trees on the FDDB database [3] with (a) discrete and (b) continuous score metrics.

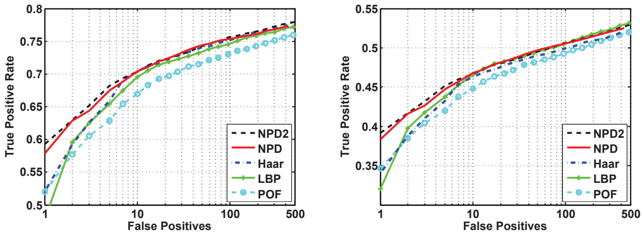
TABLE 2
Comparison of detector complexity.

	Haar	LBP	POF	NPD-stump	NPD-tree
#weak classifiers	150	108	276	1,597	176
#features learned	1,763	1,269	3,082	1,597	2,035
#feature evaluations	33.9	30.4	44.3	36.5	34.4

for continuous metric. The improvement is larger at smaller false positives.

Next, we fixed the regression tree based weak learner, but tried three other local features, namely Haar-like features [1], LBP [52], and pixelwise ordinal feature (POF) [30]. Since LBP is a discrete label, we treated it as a categorical variable in the regression tree learning, that is, for branching at each tree node, the algorithm finds the optimal criterion that splits the discrete LBP codes into two groups. Using the same training set as in Section 5.1, we trained the three detectors using Haar, LBP, and POF, respectively. The model complexity of these detectors is summarized in Table 2. It can be observed that, the NPD model appears to be more efficient than the POF model, though it requires slightly more feature evaluations than the Haar and LBP models. However, it should be noted that the computation of Haar-like features requires computing integral images, while for LBP, each feature needs to compare 8 pairs of pixels and convert the resulting binary string to the corresponding decimal number. In contrast, using look up tables as aforementioned, computing the NPD feature requires only one memory access.

The four detectors with different local features were tested on the FDDB database, and the corresponding ROC curves are shown in Fig. 16 for both the discrete and continuous score metrics. The NPD detector performs better than the Haar, LBP, and POF detectors with the same regression tree based weak learners. The performance improvements due to NPD features over Haar, LBP, and POF features are about 6%, 10%, and 6%, respectively, for discrete metric, and about 4%, 6%, and 4%, respectively, for continuous metric, at FP=1. NPD is better than POF, because with NPD features the regression tree learns optimal thresholds to form more robust ordinal rules. NPD performs better than Haar and LBP, especially at low false positives, indicating that combining optimal pixel-level features in regression trees provides better discrimination between faces and nonfaces. On the other hand, one can also observe that except at low false positives, NPD performs about the same or just slightly better than Haar-like features and LBP.



(a) discrete (b) continuous

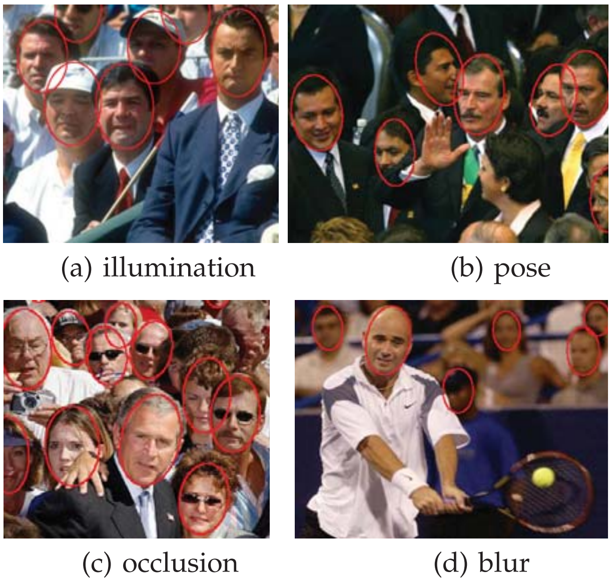
Fig. 16. Comparison of different features in regression tree based face detector on the FDDB database [3] with (a) discrete and (b) continuous score metrics.

We also tried a variation of NPD, defined as $f(x,y) = \frac{x-y}{\sqrt{x^2+y^2}}$. This is denoted as NPD2. With the same setting as NPD, we trained another detector based on NPD2. The testing results on FDDB are also shown in Fig. 16; the performances of NPD and NPD2 are about the same, with NPD2 being slightly better. However, considering that NPD is simpler than NPD2, we still suggest the formulation of Eq. (1).

5.6 Evaluation Under Specific Detection Challenge

In the following, we evaluate how the proposed NPD face detector performs under illumination variation, pose variation, occlusion, and blur (or low resolution). Note that these four challenges are often encountered simultaneously in an image. In our selection of the four subsets, one per specific challenge, we focused on the main source of variation in each image. For each challenge, we selected 100 images from the FDDB database [3] (examples are shown in Fig. 17), and ran the NPD detector described in Subsection 5.1 on each subset separately. Fig. 18 shows that the NPD face detector performs the best on the illumination subset. This is not surprising since the proposed NPD features are robust against illumination variations. Further, the NPD method performs better for face images with pose variation than with occlusion or blur. These results indicate that occlusion and blur are the two major challenges for unconstrained face detection, which have not been well addressed in the literature.

The NPD face detector is also compared with the Viola-Jones face detector implemented in OpenCV 2.4, and the commercial face detector PittPatt on the four subsets of FDDB discussed above. The resulting ROC curves with the discrete score metric are shown in Fig. 19. These plots show that the proposed NPD



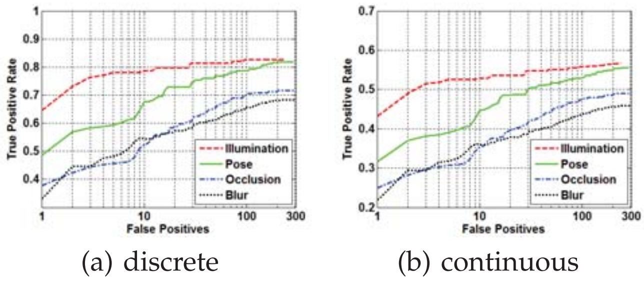
(a) illumination

(b) pose

(c) occlusion

(d) blur

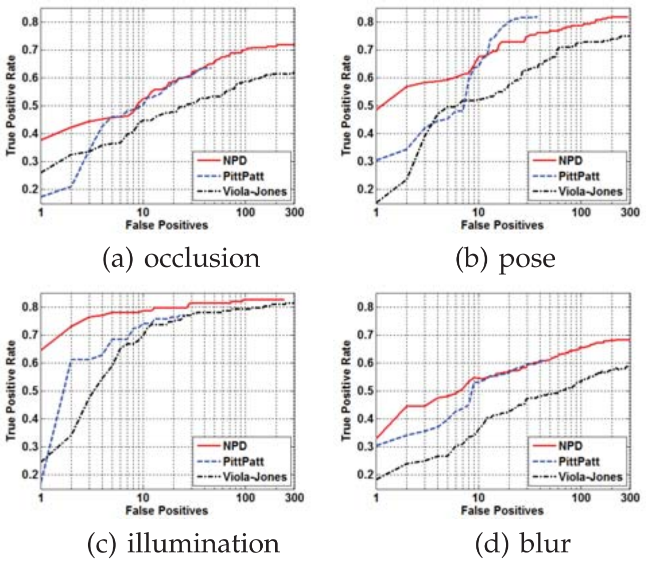
Fig. 17. Example images and annotated faces for four subsets extracted from the FDDB database [3].



(a) discrete

(b) continuous

Fig. 18. ROC curves of the proposed NPD face detector on the four subsets extracted from the FDDB database [3] with (a) discrete and (b) continuous score metrics.



(a) occlusion

(b) pose

(c) illumination

(d) blur

Fig. 19. ROC curves for face detection on four subsets from the FDDB database [3] with the discrete score metric.

face detector outperforms both the Viola-Jones and the PittPatt face detectors on all the four subsets. The reasons for the superior performance of the proposed method under illumination variations, pose variations, occlusions, and blur, were discussed in Subsection 4.2.

5.7 Detection Speed

For handheld devices like mobile phones, the available resources for computation and memory are rather limited. Therefore, face detector's complexity and detection speed are very important for embedded systems. In this subsection, we report the detection speed of the proposed NPD face detector, compared with the Viola-Jones[5] face detector in OpenCV 2.4, which is known to be optimized for speed. The proposed NPD face detector is implemented in C++; the size of the model trained in Section 5.1 is 41KB. Two platforms were selected for this evaluation: (i) a normal desktop PC with the Intel Core i5-2400 @3.1GHz CPU (4 cores, 4 threads), and (ii) a netbook with Intel Atom N450 @1.6GHz processor (1 core, 2 threads), to simulate low-end devices. For face detection evaluation, a video clip of the movie "*Jobs*" was used. This video clip shows a busy campus, with each frame containing from one to tens of faces. The length of the video clip is about 2 minutes, containing 3,950 frames in total. The original resolution is 1280×720. To test the detection speed at various resolutions, the original video clip was cropped and resized to 1920×1080, 800×600, and 640×480. In this evaluation, the minimal face size to detect was set to 40×40 pixels, and the scaling factor was 1.2. The multi threading technique was enabled in both NPD and OpenCV detectors for parallel computation.

The test results (measured in terms of Frame Per Second, FPS) are shown in Table 3. Note that we only calculated the face detection time, regardless of the video decoding time. The detection speed of the SURF cascade [13], a fast face detection algorithm, is also compared in Table 3. The detection speed of the SURF cascade algorithm is taken directly from [13], since we do not have access to the code. The detection parameters in [13] are the same as our algorithm, except that authors in [13] used the Intel Core-i7 CPU for the desktop computer. From Table 3 it can be observed that the NPD detector is much faster than both the OpenCV and SURF cascade detectors. On Atom N450 processor, the detection speed of the NPD detector is about 9 times faster than the detection speed of the OpenCV detector; on i5 processor the speed of the NPD detector is about 7 times the speed of the OpenCV detector.

Table 3 shows that the NPD detector can run in

[5] We have tested four models of the Viola-Jones face detector provided in OpenCV 2.4, and found that the "haarcascade_frontalface_alt" model is the fastest, which was selected here for comparison.

TABLE 3
Comparison of face detection speed (as FPS).

CPU	Resolution	NPD	OpenCV	SURF [13]*
Atom N450	640 × 480	**19.4**	2.1	5.8
	800 × 600	**12.1**	1.3	-
@1.6GHz	1280 × 720	**6.8**	0.7	-
(1 core, 2 threads)	1920 × 1080	**3.0**	0.3	-
i5-2400	640 × 480	**177.6**	24.4	71.3
	800 × 600	**112.6**	16.2	-
@3.1GHz	1280 × 720	**63.3**	8.9	-
(4 cores, 4 threads)	1920 × 1080	**29.6**	3.6	-

* "-" means data is not available for the SURF detector [13]. This is because we do not have access to the code, and [13] only reports detection speed at resolution 640×480 or lower.

real-time (29.6 FPS) on i5 desktop PC for processing 1920 × 1080 high definition videos. For the standard VGA (640 × 480) videos, the NPD detector on i5 processor can detect faces at even faster speed (177.6 FPS). On the low-end Atom platform, the NPD detector can still run in near real-time (19.4 FPS) for processing VGA videos. The reasons for the high processing speed of NPD are two folds. First, the NPD feature is simple, involving only two pixels. Further with the look up table technique, the evaluation of each NPD feature requires only one memory access. Second, the NPD feature can be easily scaled to various sizes of detection templates. Therefore, pre-calculating and storing multiscale templates can speed up detection because rescaling the input image is avoided.

6 Summary and Future Work

We have proposed a fast and accurate method for face detection in cluttered scenes. The method is based on the normalized pixel difference (NPD) feature in conjunction with boosted regression trees. An analysis of NPD feature shows that it possesses properties of scale invariance, boundedness, and reconstruction ability. We have developed a method for learning the optimal set of NPD features and their combinations. As a result, a single cascade AdaBoost classifier is able to achieve promising results for face detection with large pose variations and occlusions. Evaluations on three public domain databases, namely FDDB, GENKI, and CMU-MIT show that the proposed method outperforms state-of-the-art methods for unconstrained face detection. The proposed NPD face detector can process 1920 × 1080 video frames in realtime, which is about 6 times faster than the Viola-Jones face detector implemented in OpenCV 2.4. The reported results also show that occlusions and blur are two big challenges for face detection. Our future work will use the NPD feature and the classifier learning method for other applications such as face attribute classification (e.g. pose estimation, age estimation, and gender classification) and pedestrian detection.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[2] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–374, April 2000.

[3] V. Jain and E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[4] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the IEEE International Conference on Image Processing*, 2002.

[5] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, "Statistical learning of multi-view face detection," in *Proceedings of the 7th European Conference on Computer Vision*, 2002.

[6] M. Jones and P. Viola, "Fast multi-view face detection," *Mitsubishi Electric Research Lab TR-2003-96*, 2003.

[7] B. Froba and A. Ernst, "Face detection with the modified census transform," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[8] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under bayesian framework," in *Proceedings of the 3rd International Conference on Image and Graphics*, 2004.

[9] T. Mita, T. Kaneko, and O. Hori, "Joint Haar-like features for face detection," in *Proceedings of the 10th IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1619–1626.

[10] H. Zhang, W. Gao, X. Chen, and D. Zhao, "Object detection using spatial histogram features," *Image and Vision Computing*, vol. 24, no. 4, pp. 327–341, 2006.

[11] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block LBP representation," in *Proceedings of the IAPR/IEEE International Conference on Biometrics*, 2007.

[12] 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 *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[13] J. Li, T. Wang, and Y. Zhang, "Face detection using SURF cascade," in *ICCV BeFIT workshop*, 2011.

[14] B. Wu, H. Ai, C. Huang, and S. Lao, "Fast rotation invariant multi-view face detection based on real AdaBoost," in *IEEE Conference on Automatic Face and Gesture Recognition*, 2004.

[15] S. Li and Z. Zhang, "Floatboost learning and statistical face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112–1123, 2004.

[16] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 671–686, 2007.

[17] K. Hotta, "A robust face detector under partial occlusion," in *International Conference on Image Processing*, 2004.

[18] Y. Lin, T. Liu, and C. Fuh, "Fast object detection with occlusions," in *Proceedings of the European Conference on Computer Vision*, 2004, pp. 402–413.

[19] Y. Lin and T. Liu, "Robust face detection with multi-class boosting," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[20] J. Chen, S. Shan, S. Yang, X. Chen, and W. Gao, "Modification of the adaboost-based detector for partially occluded faces," in *18th International Conference on Pattern Recognition*, 2006.

[21] L. Goldmann, U. Monich, and T. Sikora, "Components and their topology for robust face detection in the presence of partial occlusions," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 559–569, 2007.

[22] M. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[23] E. H. Weber, "Tastsinn und gemeingefühl," in *Handwörterbuch der Physiologie*, R. Wagner, Ed. Brunswick: Vieweg, 1846, pp. 481–588.

[24] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," Microsoft Research, Tech. Rep. MSR-TR-2010-66, June 2010.

[25] P. Sinha, "Qualitative representations for recognition," in *Biologically Motivated Computer Vision Workshop*, 2002.

[26] J. Sadr, S. Mukherjee, K. Thoresz, , and P. Sinha, "Toward the fidelity of local ordinal encoding," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2001.

[27] S. Baluja, M. Sahami, and H. Rowley, "Efficient face orientation discrimination," in *International Conference on Image Processing*, vol. 1, 2004, pp. 589–592.

[28] S. Liao, Z. Lei, X. Zhu, Z. Sun, S. Z. Li, and T. Tan, "Face recognition using ordinal features," in *Proceedings of the 1st IAPR International Conference on Biometrics*, Hong Kong, 2006.

[29] Y. Abramson, B. Steux, and H. Ghorayeb, "Yet even faster (YEF) real-time object detection," *International Journal of Intelligent Systems Technologies and Applications*, vol. 2, no. 2, pp. 102–112, 2007.

[30] L. Wang, L. Ding, X. Ding, and C. Fang, "2D face fitting-assisted 3D face reconstruction for pose-robust face recognition," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 15, no. 3, pp. 417–428, 2011.

[31] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," MRL, Intel Labs, Tech. Rep., May 2002.

[32] S. Brubaker, J. Wu, J. Sun, M. Mullin, and J. Rehg, "On the design of cascades of boosted ensembles for face detection," Georgia Institute of Technology, Tech. Rep. GIT-GVU-05-28, 2005.

[33] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone, *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

[34] H. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 1998.

[35] E. Seemann, B. Leibe, and B. Schiele, "Multi-aspect detection of articulated objects," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[36] T. Kim and R. Cipolla, "MCBoost: Multiple classifier boosting for perceptual co-clustering of images and visual features," *Proceedings of Neural Information Processing Systems*, 2008.

[37] V. B. Subburaman and S. Marcel, "Fast bounding box estimation based face detection," in *ECCV Workshop on Face Detection: Where we are and what next*, 2010.

[38] V. Jain and E. Learned-Miller, "Online domain adaptation of a pre-trained cascade of classifiers," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[39] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[40] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[41] X. Shen, Z. Lin, J. Brandt, and Y. Wu, "Detecting and aligning faces by image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[42] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic part model for unsupervised face detector adaptation," in *IEEE International Conference on Computer Vision*, 2013.

[43] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao, "WLD: A robust local image descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, Sept. 2010.

[44] F. Kriegler, W. Malila, R. Nalepka, and W. Richardson, "Preprocessing transformations and their effects on multispectral recognition," in *Proceedings of the Sixth International Symposium on Remote Sensing of Environment*, 1969, pp. 97–131.

[45] http://mplab.ucsd.edu, "The MPLab GENKI Database, GENKI-SZSL Subset."

[46] T. L. Berg, A. C. Berg, J. Edwards, and D. Forsyth, "Whos in the picture," *Advances in neural information processing systems*, vol. 17, pp. 137–144, 2004.

[47] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," in *European Conference on Computer Vision (ECCV)*, 2004.

[48] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Robust face detection by simple means," in *DAGM Computer Vision in Applications Workshop*, 2012.

[49] S. Seguí, M. Drozdzal, P. Radeva, and J. Vitrià, "An integrated approach to contextual face detection." in *International Conference on Pattern Recognition Applications and Methods*, 2012.

[50] PittPatt Software Developer Kit, Pittsburgh Pattern Recognition, Inc., http://www.pittpatt.com.

[51] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 236–243.

[52] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

Appendix A
Boundedness of NPD

*Lemma 1 (**Boundedness**):* $\forall x, y \geq 0$, the NPD feature f(x,y) is well bounded in [-1,1]. In addition, $f(x,y) = 1$ if and only if $x > 0$ and $y = 0$; and $f(x,y) = -1$ if and only if $x = 0$ and $y > 0$.

Proof: From the definition of NPD we know that $x \geq 0$, $y \geq 0$, and $f(0,0) = 0 \in [-1,1]$. When either x or y is nonzero, for example, $y \geq 0$ but $x > 0$, Eq. (1) can be reformulated as

$$f(x,y) = \frac{x-y}{x+y} = \frac{2x}{x+y} - 1 = \frac{2}{1+\frac{y}{x}} - 1 \leq 1. \quad \text{(a)}$$

The inequality in Eq. (a) holds because $y \geq 0$, and the last equality holds if and only if $x > 0$ and $y = 0$. Similarly, when $x \geq 0$ but $y > 0$, Eq. (1) can be reformulated as

$$f(x,y) = \frac{x-y}{x+y} = 1 - \frac{2y}{x+y} = 1 - \frac{2}{\frac{x}{y}+1} \geq -1. \quad \text{(b)}$$

The inequality in Eq. (b) holds because $x \geq 0$, and the last equality holds if and only if $x = 0$ and $y > 0$. □

Appendix B
Proof of Theorem 1

Denote $f_{ij} = f(x_i, x_j)$. From Eq. (1) we have

$$f_{ij}(x_i + x_j) = x_i - x_j. \quad \text{(c)}$$

Equivalently,

$$(f_{ij} - 1)x_i + (f_{ij} + 1)x_j = 0. \quad \text{(d)}$$

Therefore, we have the following set of linear equations

$$\mathbf{F}\mathbf{x} = \mathbf{0}, \quad \text{(e)}$$

where

$$\mathbf{F} = \begin{pmatrix} f_{12}-1 & f_{12}+1 & 0 & \cdots & 0 \\ f_{13}-1 & 0 & f_{13}+1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f_{1p}-1 & 0 & 0 & \cdots & f_{1p}+1 \\ 0 & f_{23}-1 & f_{23}+1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & f_{p-1,p}+1 \end{pmatrix} \quad \text{(f)}$$

is a sparse $d \times p$ matrix with each row containing at most two nonzero entries. Furthermore, from the formulation of $\mathbf{F}$ we know that each row of $\mathbf{F}$ contains at least one nonzero entry, because $(f_{ij} - 1) \neq (f_{ij} + 1)$ always holds for all i and j. Without loss of generality, let's assume $f_{12} + 1 \neq 0$. Then it follows that $f_{1j} + 1 \neq 0, \forall j$. Because if $\exists j$ such that $f_{1j} + 1 = 0$, then from Lemma 1 we know that $x_1 = 0$. This will further lead to $f_{12} + 1 = 0$, which violates the assumption that $f_{12} + 1 \neq 0$. Therefore, the first $p-1$ rows in the matrix $\mathbf{F}$ are linearly independent of each other.

We will further prove that $rank(\mathbf{F}) = p - 1$. In fact, any row of the matrix $\mathbf{F}$ can be linearly expressed by the first $p-1$ rows. To show this, let's denote the row containing $f_{ij} - 1$ and $f_{ij} + 1$ by $\mathbf{r}_{ij}$. We will show that

$$\mathbf{r}_{ij} = \frac{f_{ij}-1}{f_{1i}+1}\mathbf{r}_{1i} + \frac{f_{ij}+1}{f_{1j}+1}\mathbf{r}_{1j}, \quad \text{(g)}$$

holds for all $i > 1$ and $j > i$. In fact, it is easy to verify that the above equation holds for all columns of $\mathbf{r}_{ij}$, $\mathbf{r}_{1i}$, and $\mathbf{r}_{1j}$ after the first column. So, we only need to show that, for the first column, we have

$$\frac{(f_{1i}-1)(f_{ij}-1)}{f_{1i}+1} + \frac{(f_{1j}-1)(f_{ij}+1)}{f_{1j}+1} = 0, \quad \text{(h)}$$

which is equivalent to

$$f_{1i}f_{1j}f_{ij} - f_{1i} + f_{1j} - f_{ij} = 0. \quad \text{(i)}$$

This can be verified by substituting each feature with its definition in Eq. (1).

Given that $rank(\mathbf{F}) = p - 1$, we know that the nullspace of $\mathbf{F}$ contains only one nonzero vector, which is a solution to Eq. (e). Furthermore, from Lemma 1 we can infer that $(f_{ij} - 1)(f_{ij} + 1) \leq 0$, hence Eq. (d) tells that $x_i x_j \geq 0, \forall i, j$. Consequently, Eq. (e) always has a nonnegative solution $\hat{\mathbf{x}}$, and all solutions to Eq. (e) must be $c\hat{\mathbf{x}}$, where c is a scale factor. □

Given this proof, we make four observations below:

- For a solution, c can be any real value, but to satisfy the constraint that all pixel intensity values are nonnegative, c should be positive.
- The solution to Eq. (e) spans a one-dimensional subspace (the nullspace).
- A specific solution can be obtained by assigning $x_1 = 1$ and solving for the other variables from the first $p - 1$ rows of Eq. (e) in linear time.
- When the original image is $\mathbf{x} = \mathbf{0}$, it can also be reconstructed by $c\hat{\mathbf{x}}$ where $\hat{\mathbf{x}}_i = 1$, $\forall i$, and $c = 0$. However, in this case a solution with $c > 0$ is not generally regarded as a scaled version of the original image $\mathbf{x} = \mathbf{0}$.