A Random Decision Forests Approach to Face Detection

M. Hassaballah and Mourad Ahmed

Department of Mathematics, Faculty of Science, South Valley University, Qena 83523, Egypt
m.hassaballah@svu.edu.eg; mourad.ahmed@sci.svu.edu.eg

Abstract. Face detection has been considered one of the most important areas of research in computer vision due to its wide range of use in human face-related applications. This paper addresses the problem of face detection using Hough transform employed within the random forests framework. The proposed Hough forests-based method is a task-adapted codebooks of local facial appearance with a randomized selection of features at each split that allow fast supervised training and fast matching at test time, where the codebooks are built upon a pool of heterogeneous local appearance features and the codebook is learned for the face appearance features that models the spatial distribution and appearance of facial parts of the human face. Experimental results are included to verify the effectiveness and feasibility of the proposed method.

1 Introduction

Because of its various uses in several applications, face detection has received a considerable attention in the last decade. The human face is the main source of information during human interaction and in vision-based Human Computer Interaction (HCI) systems. Thus, any system integrating vision-based HCI requires fast and reliable face detection [1]. The first step of any face processing system is detecting locations in images where faces are present. Face detection is also a required preliminary step to automated face recognition whose performance greatly impacts recognition rates.

According to [2] the face detection problem can be described as: given an arbitrary image, determine whether there are any human faces in the images, and if there are, return the location of each face in the image. Generally, face detectors return the image location of a rectangular bounding box containing the face. This bounding box serves as the starting point for the above mentioned applications. Automatic detection of the human face is one of the most difficult problems in pattern recognition and computer vision because the face is a non-rigid object that has a high degree of variability with respect to head poses (off-plane rotations), illumination, facial expression, occlusion, aging, image quality, and cluttered backgrounds may cause great difficulties [3].

On the other hand, random decision forests [4,5] have become popular in many applications of computer vision such as Bioinformatics [6], image classification [7], and computational genomics [8] as well as object detection/tracking [9]. Because it provides a unique combination of prediction accuracy and model interpretability among popular machine learning methods. Random Forest (RF) includes an ensemble of decision trees and incorporates a randomized selection of features at each split and interactions naturally in the learning process, which can deal with small sample size, high-dimensional

Y.J. Zhang and J.M.R.S. Tavares (Eds.): CompIMAGE 2014, LNCS 8641, pp. 375–386, 2014.

feature space, and complex data structures. Gall et al. [10] examined using random forests for three tasks: object detection, tracking, and action recognition. They proved the efficacy of forests-based method for these tasks and specifically Hough forests perform well compared to the state of the art for all the three tasks. In [11], a non-maxima suppression method is proposed for detecting multiple object instances in images using Hough transform. To obtain the probabilistic votes, the Hough forest are learned on a training data set from images with the objects of interest (pedestrians) at a fixed scale and from the set of background images. The Hough-based method copes better with multiple occluding instances; and according to the experiments conducted by the authors, a significant increase in accuracy is obtained.

Motivated by these works, in this paper, we investigate the ability of random forests for detecting the human face in digital images by employing the Hough transform within the random decision forests framework. In this respect, a direct mapping between the facial landmarks appearance and its Hough vote in the Hough space can be learned. This Hough forests-based approach can be regarded as task-adapted codebooks of local facial features appearance that allow fast supervised training and fast matching at test time. The set of leaf nodes of each tree in the Hough forest forms a discriminative codebook, where, each leaf node makes a probabilistic decision whether a patch corresponds to the facial part or not, and casts a probabilistic vote about the centroid position with respect to the patch center. As far as we know, this is the first time that Hough forest is utilized for the face detection problem. The proposed method-based Hough forests is very efficient at runtime, since matching a sample against a tree is logarithmic in the number of leaves. Therefore, the method is able to sample patches densely, while maintaining acceptable computational performance. In contrast to other methods, the proposed method is less sensitive to geometrical distortion, noise and partial occlusion. Experimental results on the widely used face database (i.e., CMU+MIT database) are presented to demonstrate the efficacy of the proposed method.

The rest of this paper is organized as follows. A brief review on existing face detection methods is presented in Section 2. The principles of Hough forests are discussed in Section 3, while the proposed method for detecting faces is introduced in Section 4. Experimental results are provided in Section 5. Conclusion along with future direction is summarized in Section 6.

2 Literature Review

As mentioned before, detection of the human face in an image is a difficult task in pattern recognition because the face is a non-rigid object that has a high degree of variability as well as variations in occlusions, illumination changes, and background clutter. Though the difficulties, the last years have shown a great deal of research effort put into face detection technology. Numerous methods have been proposed to detect faces in images. Many of these methods are reviewed in two surveys by Yang et al. [2] and by Hjelmas and Low [12]. These methods can be broadly classified into two main categories: appearance-based approaches and feature-based approaches. Appearance-based approaches are known to be better suited for detecting non-frontal faces and more successful in complex scenes, however in simple scenes feature-based approaches are more successful. In contrast to the appearance-based approaches, feature-based approaches make explicit use of face knowledge. They are usually based on the detection of local invariant features of the face such as eyes, eyebrows, nose, mouth, and the structural relationship between these facial features. Based on the detected facial features, a statistical model is built to describe their relationships and to verify the existence of a face. There are other face detection methods that use a combination of both approaches in order to achieve a more robust and better performance [13].

Viola and Jones [14] present a machine learning approach for face detection, which has been integrated into OpenCV library with five Haar-cascade classifiers. Their method is probably the best known face detection method and it has gained a wide spread acceptance due to the availability of an open source implementation. The novelty of this method comes from the integration of a new image representation (integral image), a learning algorithm (based on AdaBoost to build a very rapid cascade classifier based on weak classifiers ("Haar-like basis functions"), and a method for combining the classifiers cascade. The original work on frontal faces has been extended to detect tilted and non-frontal faces by extending the set of basic features and by introducing pose estimators. Variations of the framework that use different basis sets have been presented using Gabor wavelets, local orientations of gradient and Laplacian based filters [15,16]. Li et al. [17] modify the monotonic assumption of the Adaboost algorithm proposed by Viola and Jones [14] to develop the so-called Floatboost algorithm for the training of face and non-face classifiers. By implementing these classifiers using a coarse-to-fine and simple-to-complex pyramidal structure, the authors successfully develop a computationally efficient multi-view face detection system. However, the proposed classifiers used in such boosted cascades operate independently of each other and therefore discard useful information between layers, resulting in convergence problems during the training process. In addition, non-face samples collected by the bootstrap procedure are incorporated within the database during the training process and hence increase the complexity of the classification task. Moreover, during the latter stages of the training process, the pattern distributions of the face and non-face regions may become so complicated that it is virtually impossible to distinguish between them on the basis of their Haar-like features as reported in [18].

Chen and Lien [18] develop a statistical system for automatic multi-view face detection and pose estimation consisting of five modules, Their statistical multi-view face detection system is based on significant local facial features (or subregions) rather than the entire face. The low and high frequency feature information of each subregion of the facial image are extracted and projected onto the eigenspace and residual independent basis space in order to create the corresponding PCA (principal component analysis) projection weight vector and ICA (independent component analysis) coefficient vector, respectively. Therefore, the system has an improved tolerance toward different facial expressions, wide viewing angles, partial occlusions and lighting conditions due to projecting on feature subspaces. Furthermore, either projection weight vectors or coefficient vectors in the PCA or ICA space have divergent distributions and are therefore modeled by using the weighted Gaussian mixture model (GMM) rather than a single Gaussian model. The GMM weights and parameters of the GMM are estimated iteratively using the Expectation Maximization (EM) algorithm. Face detection is then performed by conducting a likelihood evaluation process based on the estimated joint probability of the weight and coefficient vectors and the corresponding geometric positions of the subregions. Regarding the overall performance of this multi-view face detection method, as the authors reported the system can successfully function under various imaging conditions with the accurate detection rate of higher than 91% and can estimate the pan-rotation angles of more than 90% of the input patches to within $\pm 10^\circ$ of their ground-truth values. Though this high detection rate, this method depends basically on different types of thresholds and several parameters should be adapted in advance in different databases. So the method is neither simple nor applicable.

Yang et al. [19] incorporate a genetic algorithm into the AdaBoost training to optimize the detection performance given the number of Haar features for embedded systems. While, in [20], a bank of Gabor filters is utilized to search for ten facial features (eye corners, eye centers, nostrils and mouth corners). Each feature is modeled using a Gaussian Mixture Model (GMM) of feature responses. Any triplet of feature detections with an acceptable spatial orientation produce a face location hypothesis. These face candidates are then normalized using an affine transformation and tested using a SVM region classifier. The highest ranking candidate based on the SVM discriminant function is declared the location of the face. The method detects 91% of faces in the XM2VTS database and 65% of BioID database within 10% of the true inter-ocular distance. The proposed approach in this paper is closely related to this family of facial feature-based methods.

3 The Hough Decision Forests

This section describes the necessary general background of the Hough forests framework and the notation that we will use in the rest of the paper. Hough forests consist of a collection of randomized trees where each tree consists of split nodes and leaves. During training, in each splitting node the algorithm tries to split the given training data $\{z_i; v_i\}_{i=1}^{N}$ where $z_i \in R^D$ is a D-dimensional feature vector, $v_i \in \{1, \ldots, C\}$ is the corresponding class label, and N is the number of training samples. By predefined the number of splitting functions, this recursive algorithm continues to split the data until either the maximum depth of the tree is reached; the subset of the data in a node is pure, or the number of samples is below a threshold. If any of these conditions is met, a leaf node is created and the class probability $p(v|z)$ is estimated.

Hough forests work on small patches extracted at random locations within a given bounding box from positive and negative training images of a face, each patch is described with several features, termed channels. Positive samples additionally store an offset vector pointing to the center of the face. Hough Forests then try to separate positive from negative patches and simultaneously cluster together similar positive patches according to their offset vectors. The splitting functions at each node in the Hough Forests randomly selects a feature channel and two pixels within the patch and calculates the difference of the feature values. This difference is then thresholded to determine which patches are forwarded to the left or the right child node. While, in the test phase, each image patch is passed through all trees in parallel, in each non-leaf node, a simple binary test is performed. The test is applied to each patch that arrives in the node, and its output defines the child that the patch will proceed to. The set of leaf nodes of each tree in the Hough forest can be regarded as a discriminative codebook. Each leaf node makes a probabilistic decision whether a patch corresponds to a part of the face or to the background, and casts a probabilistic vote about the centroid position with respect to the patch center in a probabilistic generalized Hough transform, and the maxima in the Hough voting space (Hough image) correspond to face hypotheses.

4 The Proposed Methodology

Figure 1 shows steps of the proposed method using Hough forests to detect faces in images, which can be summarized as follows: first, the different views of a human face can be handled by a single codebook B with entries $B_1, \ldots, B_b$ for each face pose in the images. The training procedure first extracts a set of patches which are sampled from a set of bounding box annotated positive images of facial landmarks and a set of background images. The set of training patches P_j^{train} are randomly sampled from the examples used to construct each tree T on the Hough forests, where the set of patches is $\{P_j^{train} = (a_j, l_j, o_j)\}$, where a_j are the extracted image feature channels Γ of the patch (facial appearance), l_j is the class label for the patch, and o_j is a offset vector from the patch center to the centroid. The patches sampled from the negative set (background patches) are assigned the class label $l_j = 0$, while the patches sampled from the interior of the face bounding boxes are assigned $l_j = 1$. Each face patch is also assigned a 2D offset vector o_j equal to the offset from the centroid of the bounding box to the center of the patch. Based on such a set of patches, the Hough forests trees are then constructed recursively starting from the root.

Second, the selection of random tests is based on how well they separate the input set of patches, the quality of the separation is measured by one of two uncertainty measures: class label uncertainty μ_1 measuring the impurity of the class labels l_j and offset uncertainty μ_2 measuring the impurity of the offset vectors o_j

$$\mu_1(\mathcal{A}) = |\mathcal{A}|.\ \mathcal{E}(\{l_j\}) \tag{1}$$

$$\mu_2(\mathcal{A}) = \sum_{j:l_j=1} \| (o_j - \mathcal{O}_m) \|^2 \tag{2}$$

Where $\mathcal{A}$ is the set of patches assigned to a node $\mathcal{A} = \{P_j^{train}\}$, $|\mathcal{A}|$ is the number of patches in the set $\mathcal{A}$, and $\mathcal{O}_m$ is the mean offset of this set. $\mathcal{E}$ is Shannon entropy, used to maximize the classification information gain. The class label entropy is

$$\mathcal{E}(\{l_j\}) = -\sum_{l\in\{0,1\}} \mathcal{P}(l_j|\mathcal{A})\ log\left(\mathcal{P}(l_j|\mathcal{A})\right) \tag{3}$$

Where $\mathcal{P}(l_j|\mathcal{A})$ is the proportion of patches with class label l_j in the set $\mathcal{A}$. The first measure μ_1 tries to create two subsets of patches that are as pure as possible in terms of their class labels, while the second measure μ_2 forces the patch offsets to be spatially coherent. When the number of patches is below a certain threshold or the maximum predefined height of the tree is reached, the node is declared a leaf. For each leaf node

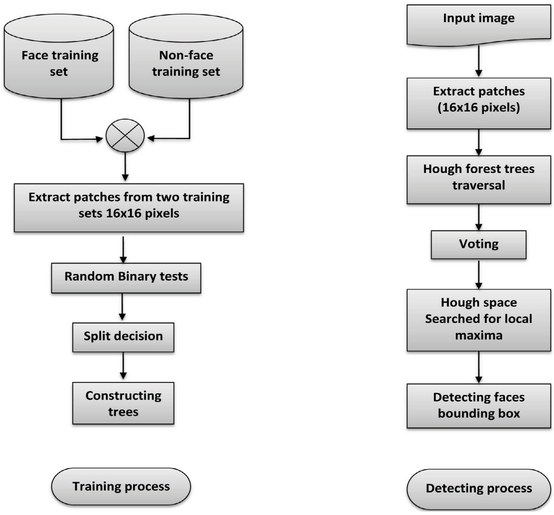


Fig. 1. Flowcharts of the training and detecting processes of the proposed face detection method

L in the constructed tree, the information about the patches that have reached this node at train time is stored. Thus, we store the proportion $\mathcal{F}_L$ of the facial patches (i.e., $\mathcal{F}_L = 1$ means that only facial patches have reached the leaf) and the list $\mathcal{O}_L = o_j$ of the offset vectors corresponding to the facial patches. In this context, the leaves of the tree in the forest form a discriminative codebook with the assigned information about possible locations of the face center. At runtime, this information is used to cast the probabilistic Hough votes about the existence of the face at different positions.

Third, the appearance of the patch a_j for each non leaf node in each tree is assigned a binary test during training. The patches have a fixed size of pixels at both train and test time; the appearance of the patch can be written as $a_j = (\Gamma_j^1, \Gamma_j^2, \ldots, \Gamma_j^c)$, where c is the number of the extracted feature channels. The binary tests on a patch appearance $\mathcal{T}(a) \rightarrow \{0, 1\}$ is defined as simple pixel-based tests. Such a test simply compares the values of a pair of pixels in the same channel with some threshold. The test is defined by a channel $\alpha \in \{1, 2, \ldots, c\}$, two positions p, q in the patch image, and a real threshold value r. The test $\mathcal{T}_{(\alpha,p,q,r)}(a)$ is defined as:

$$\mathcal{T}_{(\alpha,p,q,r)}(a) = \begin{cases} 0, \text{ if } \Gamma^{\alpha}(p) - \Gamma^{\alpha}(q) < r \\ \\ 1, \text{ otherwise} \end{cases} \tag{4}$$

Using (1) and (2) for uncertainty measures μ_1 and μ_2, a pool of binary tests $\{\mathcal{T}\}$ can be generated by sampling α, p, and q uniformly given a training set of patches P^{train}. The threshold value r for each test is chosen uniformly from the range of differences observed on the data randomly. Then, the random decision is made whether should minimize the class label uncertainty μ_1 or the offset uncertainty μ_2 at the non-leaf node.

We choose this with equal probability unless the number of negative patches is small than 5%, in the case of the non-leaf node it is chosen to minimize the offset uncertainty μ_2. Finally, the set of patches arriving at the non-leaf node is evaluated with all binary tests in the pool and the binary test satisfying the minimization target Ω, which is sum of the respective uncertainty measures to split the training set, Ω can be defined as:

$$\Omega_k = min\bigg(\mu_\gamma\Big(\{\mathcal{P}_j|\mathcal{T}^k(a_j) = 0\}\Big) + \mu_\gamma\Big(\{\mathcal{P}_j|\mathcal{T}^k(a_j) = 1\}\Big)\bigg) \quad (5)$$

Where $\mu_\gamma = \mu_1$ or μ_2 depending on the random choice. By choosing the non-leaf nodes that decrease the class label uncertainty μ_1 with the non-leaf nodes that decrease the offset uncertainty μ_2, the tree construction process ensures that the sets that reach the leaf have low variations in both class labels and offsets (leaves represent patches for the facial features only).

In general, the tree construction for generating the codebook follows the common Hough forests framework [9]. During the construction, each node receives a set of training patches. If the depth of the node is equal to the maximal one ($\mathcal{D}_{max} = 15$) or the number of patches is small ($\mathcal{N}_{min} = 20$), the constructed node is declared a leaf and the leaf vote information $(\mathcal{F}_L, \mathcal{O}_L)$ is accumulated and stored. Otherwise, a non-leaf node is created and an optimal binary test is chosen from a large pool of randomly generated binary tests. For detecting the face, image patches are sampled from the test image and passed through the trees, every patch of the test image P_i^{test} is matched against the codebook B and its probabilistic votes are cast to the Hough image, the image patches can be densely sampled or subsampled as for training. Consider a patch $P^{test}(y) = (a(y), l(y), o(y))$ centered at the position y in the test image, where, y lies inside the face bounding box $\mathcal{B}(x)$ centered at x. Here, $a(y)$ is the appearance of the patch, $l(y) = 1$ is the hidden class label and $o(y)$ is the hidden offset vector from the center of the face bounding box to y. Furthermore, $E(x)$ denotes the random event corresponding to the existence of the face centered at the location x in the image. The probabilistic evidence $\mathcal{P}(E(x)|a(y))$ that the appearance $a(y)$ of the patch brings about the availability $E(x)$ at different positions x in the image is defined as:

$$\begin{aligned} \mathcal{P}(E(x)|a(y)) = \mathcal{P}(E(x), l(y) = 1|a(y)) = \\ \mathcal{P}(o(y) = y - x|l(y) = 1, a(y)) \cdot \mathcal{P}(l(y) = 1|a(y)) \end{aligned} \quad (6)$$

Assuming that for a tree T the patch appearance ends up in a leaf L, the first factor can then be approximated using the probability density estimation methods [21] based on the offset vectors D_L collected in the leaf at train time, while the second factor can be straightforwardly estimated as the proportion C_L of face patches at train time. For a single tree T, the probability estimate is

$$\mathcal{P}(E(x)|a(y); \mathrm{T}) = \left[\frac{1}{|\mathcal{O}_L|}\sum_{o\in\mathcal{O}_L}\frac{1}{2\pi\delta^2}\exp(-\frac{\|(y-x)-o\|^2}{2\delta^2})\right]\cdot\mathcal{F}_L \quad (7)$$

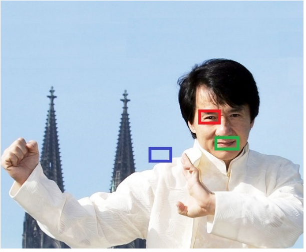
Where $\delta^2 \mathrm{I}_{(2\times 2)}$ is the covariance of the Gaussian Parzen-Window, for the entire forest $\{\mathrm{T}_t\}_{t=1}^{F}$, we simply average the probabilities (7) coming from different trees

$$\mathcal{P}(E(x)|a(y);\{\mathrm{T}_t\}_{t=1}^{F}) = \frac{1}{F}\sum_{t=1}^{F}\mathcal{P}(E(x)|a(y);\mathrm{T}_t) \tag{8}$$

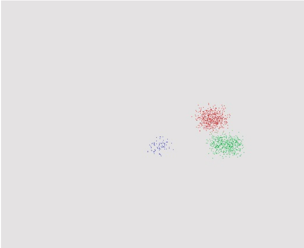
Equations (7) and (8) define the probabilistic vote cast by a single patch about the existence of the face. To integrate the votes coming from different patches, we accumulate them in an additive way into a 2D Hough image $H(x)$ using

$$H(x) = \sum_{y\in\mathcal{B}(x)}\mathcal{P}(E(x)|a(y);\{\mathrm{T}_t\}_{t=1}^{F}) \tag{9}$$

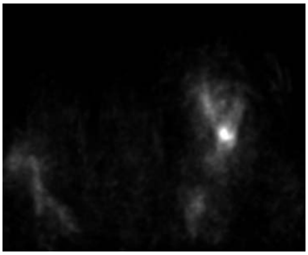
The detection procedure simply computes the Hough image H and returns the set of its maxima locations and values $\{\overline{x}, H(\overline{x})\}$ as the face hypotheses. The Hough image $H(x)$ is then obtained by Gaussian filtering the vote counts accumulated in each pixel. An alternative way to find the maxima of the Hough image would be to use the mean-shift procedure as it is done in [11]. To handle scale variations, let us first assume that the size of the detected face bounding boxes is fixed to $w \times h$ during both training and testing. The test image is resized by a set of scale factors $\sigma_1, \sigma_2, \ldots, \sigma_z$. The Hough images $H^1, H^2, \ldots, H^z$ are then computed independently at each scale. After that, the images are stacked in a 3D scale vector, the Gaussian filtration is performed across the third (scale) dimension, and the maxima of the resulting function are localized in 3D



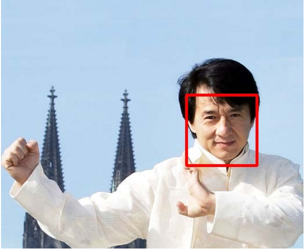
(a) Sample patches in an image



(b) Probability of each patch in (a)



(c) Aggregating all votes



(d) Face location in the image

Fig. 2. Aggregating the votes of patches into the Hough space; the Hough image peak is the face

scale vector. The resulting face hypotheses have the form $(\overline{x}, \overline{\sigma}, H^{\overline{\sigma}}(\overline{x}))$. Finally, the hypothesized bounding box in the original image is then centered at the point $\frac{\overline{x}}{\overline{\sigma}}$, has the size $\frac{w}{\overline{\sigma}} \times \frac{h}{\overline{\sigma}}$, and the face detection confidence $H^{\overline{\sigma}}(\overline{x})$ as illustrated in Fig. 2.

5 Experimental Results

For training, we use a set of 500 face images with a fixed size of 24×24 pixels. While, non-face training set contains 2,750 images cropped manually and collected by random sampling non-face regions of different images downloaded from the Internet. A total of 15000 random binary tests are considered for each node. Furthermore, each tree was trained on 20000 positive and 20000 negative patches. It should be noted that in the proposed approach, the positive patches (facial features) need to collaborate somehow to detect the searched faces. The extracted features of the patches are as follows; the first two channels contain the pixel values and normalized ones to avoid the effect of illumination, the first and second derivatives in x,y directions, and the rest of channels are the HOG descriptors respectively. Other local features descriptors such as SURF and SIFT, or Gabor wavelets may be used, but in this work we examine the HOG descriptor [22]. This is because the definition of split functions (4) is in general based on local image features (i.e., locations and descriptors). Furthermore, for time-efficiency reasons and memory, split functions need to be simple but should also be designed for maximizing the information gain.

The performance of the proposed face detection is evaluated in terms of the receiver operating characteristics (ROC) curve. Where, the two quantities of interest are clearly the number of correct detections, which one wishes to maximize, and the number of false detections, which should be minimize. The ROC curve plots the true positive rate

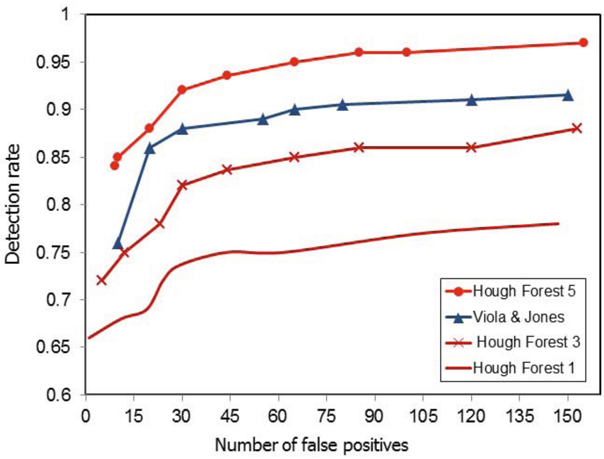


Fig. 3. ROC curves for Hough forests method with different tree number and Viola &Jones [14] method on CMU+MIT database

versus the false positive rate. We investigate the effect of trees number in the forest; thus the detector is trained for three different Hough forests trees number with same setting and training data used in constructing the trees. The first detector is trained for Hough forests of only one tree, the second detector of three trees, while the last detector of five trees. The ROC curves are obtained for each one of the three detector using the CMU+MIT database [23]. From this experiment, we note that there is a significant variation in the performance between the three detectors. The detector of five trees (i.e., Hough Forests 5) performs best compared to the other detectors achieving a high detection rate of 96% at 60 false positives as shown in Fig. 3. The Hough forests detector with five trees in the forest is compared with the baseline face detector of Viola and Jones [14] using CMU+MIT database. Figure 3 shows also the ROC curves for this comparison; the presented results on the curves are extracted from author's publication without any modifications. It is clear that the Hough forests based method with five trees outperforms the compared Viola & Jones' method achieving the highest detection rate of 97.4% at 156 false positives. In particular, the implementation of OpenCV 2.4.2 with the default frontal face classifier configuration (i.e., *haarcascadefrontalface-default.xml*) of Viola & Jones method is used. Some examples that are successfully detected by Hough forests based method but failed in Viola & Jones method are given in Fig. 4.

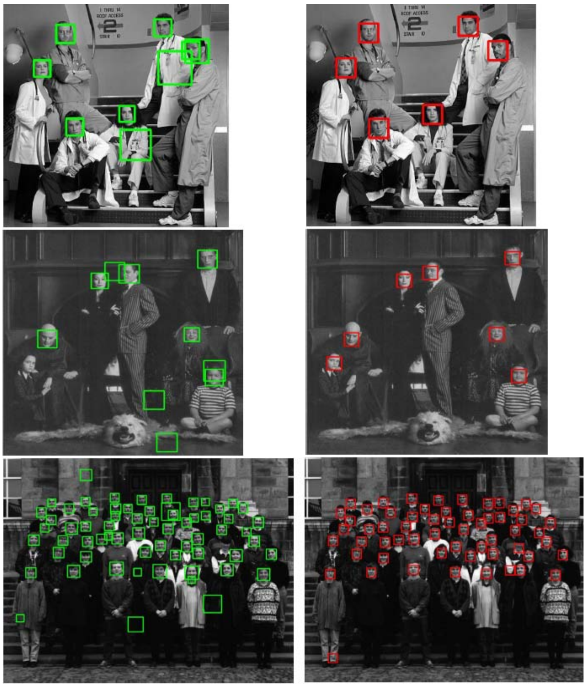


Fig. 4. Comparison detection examples using Viola & Jones' method (green left) and Hough forests method (red right) on test images from the CMU+MIT database

6 Conclusions

This paper introduced a method for face detection based on Hough forests that can learn a mapping from local image or depth patches to a probability over the parameter space. We chose Hough forests approach, because it is capable to handle large training datasets, high generalization power, fast computation, and ease of implementation. A simple experimental evaluation is conducted on the CMU+MIT database for face detection and the obtained results are encouraging. The performance of the proposed method is compared to the baseline face detector of Viola and Jones. There is still a room to further improve the detection performance, so our future work includes using non-maxima suppression that can be combined with Hough forests to improve the detection results. Investigating the aggregating of local descriptors SIFT, SURF, or HOG-LBP into Hough forests is another promising approach for improving the detection accuracy.

References

1. Hassaballah, M., Murakami, K., Ido, S.: Face detection evaluation: A new approach based on the golden ratio Φ. Signal, Image and Video Processing (SIViP) Journal 7(2), 307–316 (2013)
2. Yang, M., Kriegman, D., Ahuja, N.: Detecting faces in images: A survey. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
3. Tsao, W.-K., Lee, A.J.T., Liu, Y.-H., Chang, T.-W., Lin, H.-H.: A data mining approach to face detection. Pattern Recognition 43(3), 1039–1049 (2010)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations and Trends in Computer Graphics and Vision 7(2-3), 81–227 (2011)
6. Barrett, J., Cairns, D.: Application of the random forest classification method to peaks detected from mass spectrometric proteomic profiles of cancer patients and controls. Statistical Applications in Genetics and Molecular Biology 7(2), 1544–6115 (2008)
7. Moosmann, F., Nowak, E., Jurie, F.: Randomized clustering forests for image classification. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(9), 1632–1646 (2008)
8. Chen, X., Ishwaran, H.: Random forests for genomic data analysis. Genomics 99(6), 323–329 (2012)
9. Gall, J., Lempitsky, V.: Class-specific Hough forests for object detection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1022–1029 (2009)
10. Gall, J., Yao, A., Razavi, N., Gool, L.-V., Lempitsky, V.: Hough forests for object detection, tracking, and action Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 33(11), 2188–2202 (2011)
11. Barinova, O., Lempitsky, V., Kohli, P.: On detection of multiple object instances using Hough transforms. IEEE Trans. on Pattern Analysis and Machine Intelligence 34(9), 1773–1784 (2012)
12. Hjelmas, E., Low, B.: Face detection: A survey. Computer Vision and Image Understanding 83(3), 236–274 (2001)
13. Tabatabaie, Z., Rahmat, R., Udzir, N.B., Kheirkhah, E.: A hybrid face detection system using combination of appearance-based and feature-based methods. International Journal of Computer Science and Network Security 9(5), 181–185 (2009)

14. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)
15. Brubaker, S.C., Wu, J., Sun, J., Mullin, M., Rehg, J.: On the design of cascades of boosted ensembles for face detection. International Journal of Computer Vision 77(1-3), 65–86 (2008)
16. Xiaohua, L., Lam, K.-M., Lansun, S., Jiliu, Z.: Face detection using simplified Gabor features and hierarchical regions in a cascade of classifiers. Pattern Recognition Letters 30(8), 717–728 (2009)
17. Li, S., Zhang, Z.: Floatboost learning and statistical face detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(9), 1112–1123 (2004)
18. Chen, J.-C., James Lien, J.-J.: A view-based statistical system for multi-view face detection and pose estimation. Image and Vision Computing 27(9), 1252–1271 (2009)
19. Yang, M., Crenshaw, J., Augustine, B., Mareachen, R., Wu, Y.: Adaboost based face detection for embedded systems. Computer Vision and Image Understanding 114(11), 1116–1125 (2010)
20. Hamouz, M., Kittler, J., Kamarainen, J.-K., Paalanen, P., Kälviäinen, H., Matas, J.: Feature-based affine-invariant localization of faces. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(9), 1490–1495 (2005)
21. Parzen, E.: On estimation of a probability density function and mode. Annals of Math. Statistics 33(3), 1065–1076 (1962)
22. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 886–893 (2005)
23. Schneiderman, H., Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 741–751 (2000)