Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090255

A Probabilistic Framework for Robust Face Detection

Mourad Ahmed[1], *M. Hassaballah*[1,*], *Yasser Salama Hassan*[1], *A. H. Abd-Ellah*[2] *and A. S. Abdel Rady*[1]

[1] Department of Mathematics, Faculty of Science, South Valley University, Qena 83523, Egypt.
[2] Department of Mathematics, Faculty of Science, Sohag University, Sohag 82524, Egypt.

Received: 26 Jan. 2014, Revised: 24 Sep. 2014, Accepted: 25 Sep. 2014
Published online: 1 Mar. 2015

Abstract: Due to its wide range of use in human face-related applications, face detection has been considered one of the most important areas of research in computer vision and visual pattern recognition communities. Though current methods perform well on controlled face images, their performance degrades considerably under realistic scenarios that include pose, illumination and blur challenges as well as low-resolution images. This paper proposes an efficient approach for detecting faces in uncontrolled imaging conditions using a probabilistic framework based on Hough forests. Hough forests can be regarded as task-adapted codebooks of local appearance that allow fast supervised training and fast matching at test time, codebooks are built upon a pool of heterogeneous local appearance features, a codebook is learned for the face appearance features that models the spatial distribution and appearance of facial parts of the human face. Extensive evaluation of the proposed method on various databases shows the usefulness of the method. We show that the suggested method improves the detection rate and accuracy outperforming the state-of-the-art methods.

Keywords: Face detection, Face localization, Random Forests, Hough forests

1 Introduction

Because of its various uses, face detection has received considerable attention, especially in the last decade. The human face is the main source of information during human interaction; hence any system integrating Vision-Based Human Computer Interaction requires fast and reliable face detection [1,2]. The first step of any face processing system is detecting the 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. In other words, face detection can be considered the step stone to all facial analysis algorithms, including face alignment, face modeling, face relighting, face recognition, face verication/authentication, head pose tracking, facial expression tracking/recognition, gender/age recognition, and many applications include face detection-based autofocus and white balancing in cameras, new methods for sorting and retrieving images in digital photo management software, image editing software tailored for faces.

According to [3] 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 [4].

Recently, the more practical yet considerably more complicated problem of uncontrolled imaging conditions face detection has gained increasing attention [5]. In this paper, we provide a method for robust face detection under various imaging conditions based on Hough forests that can learn a mapping from local image or depth patches to a probability over the parameter space. Hough forests can be regarded as task-adapted codebooks of local appearance that allow fast supervised training and fast matching at test time. In other words, Hough forests are sets of decision trees learned on the training data. Each tree in the Hough forest maps local appearance of face to its leaves, where each leaf is attributed a probabilistic vote in the Hough space. The set of leaf nodes of each tree in the Hough forest forms a

* Corresponding author e-mail: m.hassaballah@svu.edu.eg

discriminative codebook, where, each leaf node makes a probabilistic decision whether a patch corresponds to the face or to the background, 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 voting is used for face detection task. In this context, 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 a number of widely used face databases 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 detection of faces is introduced in Section 4. Experimental results are reported in Section 5 and finally, the conclusions and future research are given in Section 6.

2 Related work

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. Changes in view can induce substantial variation in a faces visual appearance. In full-face (or frontal) view, for example, faces contain a contiguous pair of eyes, which are located either side of a centrally positioned nose. By comparison, only a single eye is visible in a profile view of the head, and this eye is located much more peripherally than both eyes in a full face. The appearance of other facial landmarks, such as the nose and mouth and more global visual characteristics, such as the head outline and hair, also vary across different face views and can change the overall appearance of a face substantially. This variation is such that observers often fail to match two different views of the same face [6]. Even though these difficulties, the last ten 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 recent surveys by Yang et al. [3] and by Hjelmas and Low [7]. 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 [8].

Viola and Jones [9] 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 the introduction of a pose estimator. Variations of the framework that use different basis sets have been presented; e.g., Gabor wavelets, and local orientations of gradient and Laplacian based filters [10,11].

Li et al. [12] modify the monotonic assumption of the Adaboost algorithm proposed by Viola and Jones [9] 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 [13]. Yang et al. [14] incorporate a genetic algorithm into the AdaBoost training to optimize the detection performance given the number of Haar features for embedded systems.

Liu [15] utilized the 1D Harr wavelet transform to effectively detect the face. He designed two wavelet-based transformed faces, a horizontally corresponding face and a vertically corresponding face, and then combined these two faces to form a histogram for face templates. Finally, a Bayesian classifier is applied to locate the face regions from images. In [16], 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. In this respect, detection of facial features is not an easy task at all [17]. Furthermore, many feature-based methods are unsuitable for detection of low resolution faces.

Chen and Lien [13] 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. The proposed method in this paper builds upon the class-specific Hough forest detection framework [18], The Hough forests framework is based on the generalized Hough transform which is in turn inspired by the implicit shape model detector [19]. Both of those approaches maps the appearance of object parts onto codebook with specific spatial distribution.

3 The Hough Forests

This section describes the necessary background of the Hough forest framework and the notation that we will use in the rest of the paper. Hough forests are in many aspects similar to other random forests in computer vision. Random forests have recently attracted a lot of attention in computer vision [20,21,22,23]. It consists 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 an object, each patch is described with several features, termed channels. Positive samples additionally store an offset vector pointing to the center of the object, the center point in our case is pointing to the center of the face facial parts depending on the pose of the face in the image as shown in Fig. 1. 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.

In the test stage, 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 object 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 object hypotheses.

4 Face Localization with Hough Forests

The main steps of proposed method using Hough forests to detect and localize faces in images are shown in Fig. 2, and can be summarized as follows:

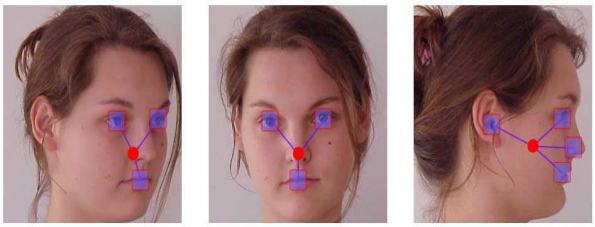


Fig. 1: Patches of facial components in a positive training data.

First, the different views of a human face can be handled by a single codebook, for generating the 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 are the base that used to construct each tree T on the Hough forests. We define a set of patches as:

$$\{P_j^{train} = (a_j, l_j, o_j)\} \tag{1}$$

Where a_j are the extracted image feature channels Γ of the patch (face 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. (Note that the o_j is undefined for a background 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

$$\mu_1(\mathcal{A}) = |\mathcal{A}|.\ \mathcal{E}(\{l_j\}) \tag{2}$$

$$\mu_2(\mathcal{A}) = \sum_{j:l_j=1} \parallel (o_j - \mathcal{O}_m) \parallel^2 \tag{3}$$

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 set $\mathcal{A}$, and $\mathcal{O}_m$ is the mean offset of this set. $\mathcal{E}$ is Shannon entropy we use it

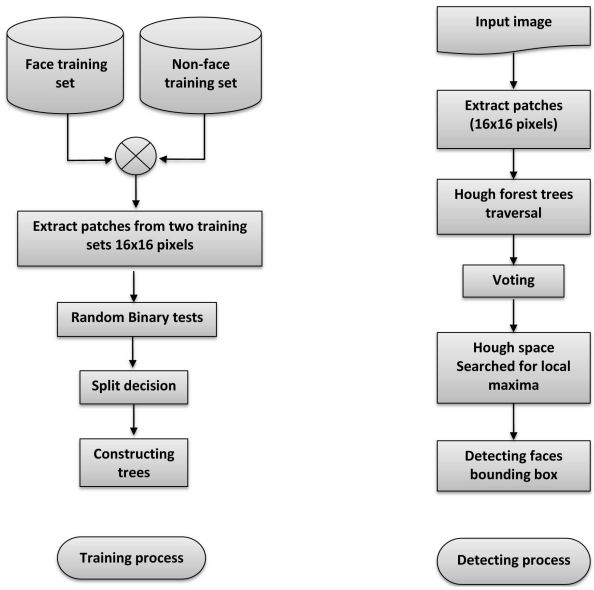


Fig. 2: Flowcharts of the training and detecting processes of the proposed face detection method.

to maximize the calssification information gain. The class label entropy is defined in a standard way:

$$\mathcal{E}(\{l_j\}) = -\sum_{l \in \{0,1\}} \mathcal{P}(l_j|\mathcal{A})\ log\ \Big(\mathcal{P}(l_j|\mathcal{A})\Big) \tag{4}$$

Where $\mathcal{P}(l_j|\mathcal{A})$ is the proportion of patches with class label l_j in 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 patches 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 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 face patches (e.g., $\mathcal{F}_L = 1$ means that only face patches have reached the leaf) and the list $\mathcal{O}_L = o_j$ of the offset vectors corresponding to the face patches. The leaves of the tree thus form a discriminative codebook with the assigned information about possible locations of the centroid. 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 a patch a_j for each non leaf node in each tree is assign a binary test during training. The patches have a fixed size 16×16 pixels at both train and test time, and the appearance is defined by the extracted feature channels. Thus, the appearance of the patch can be written as $a_j = (\Gamma_j^1, \Gamma_j^2, \ldots, \Gamma_j^c)$, where each Γ_j^i is a 16×16 image and c is the number of 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 16×16 image, and a real threshold value r. The test $\mathcal{T}_{(\alpha,p,q,r)}(a)$ can be defined as :

$$\mathcal{T}_{(\alpha,p,q,r)}(a) = \begin{cases} 0, \text{ if } \Gamma^{\alpha}(p) - \Gamma^{\alpha}(q) < r \\ \\ 1, \text{ otherwise} \end{cases} \quad (5)$$

Using (2) and (3) for uncertainty measures μ_1 and μ_2, the binary test $\mathcal{T}$ is chosen as follows. Given a training set of patches P^{train} , firstly, a pool of binary tests $\{\mathcal{T}\}$ were generated by sampling α, p, and q uniformly. 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 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 following minimization target Ω which is sum of the respective uncertainty measures to split the training set, Ω can be defined as:

$$\Omega_k = min\Big(\mu_\gamma\Big(\{\mathcal{P}_j|\mathcal{T}^k(a_j)=0\}\Big) + \mu_\gamma\Big(\{\mathcal{P}_j|\mathcal{T}^k(a_j)=1\}\Big)\Big) \quad (6)$$

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 face facials only).

In general, the tree construction for generating the codebook follows the common Hough forests framework [18]. 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 a 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:

$$\mathcal{P}(E(x)|a(y)) = \mathcal{P}(E(x), l(y)=1|a(y)) = \mathcal{P}(E(x)|l(y)=1, a(y)) \cdot \mathcal{P}(l(y)=1|a(y)) = \mathcal{P}(o(y)=y-x|l(y)=1, a(y)) \cdot \mathcal{P}(l(y)=1|a(y)) \quad (7)$$

Assume 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: Parzen-Window Density Estimation [24] based on the offset vectors $\mathcal{O}_L$ collected in the leaf at train time, while the second factor can be straightforwardly estimated as the proportion $\mathcal{F}_L$ of face patches at train time. For a single tree T, the probability estimate is defined as:

$$\mathcal{P}(E(x)|a(y);\mathrm{T}) = \left[\frac{1}{|\mathcal{O}_L|}\sum_{o\in\mathcal{O}_L}\frac{1}{2\pi\delta^2}\exp\left(-\frac{\|(y-x)-o\|^2}{2\delta^2}\right)\right]\cdot\mathcal{F}_L \quad (8)$$

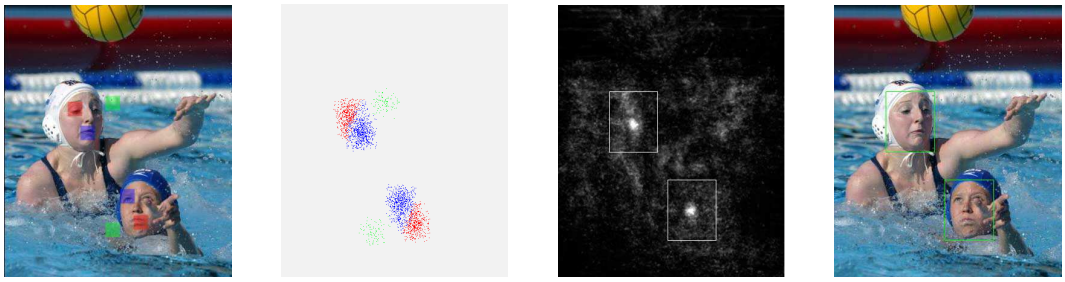
Where $\delta^2\mathrm{I}_{(2\times2)}$ is the covariance of the Gaussian Parzen-Window, for the entire forest $\{\mathrm{T}_t\}_{t=1}^F$, we simply average the probabilities (8) coming from different trees

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

Equations (8) and (9) 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 (admittedly non probabilistic) additive way into a 2D Hough image $H(x)$, which for each pixel-location x sums up the votes (9) coming from the nearby patches by:

$$H(x) = \sum_{y\in\mathcal{B}(x)}\mathcal{P}(E(x)|a(y);\{\mathrm{T}_t\}_{t=1}^F) \quad (10)$$

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 detection 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 other Hough voting-based frameworks [25,26]. 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



(a) (b) (c) (d)

Fig. 3: For each of the three patches emphasized in(a), the face Hough forest casts weighted votes about the possible location of a face (b) (each color channel corresponds to the vote of a sample patch). Note the weakness of the vote from the background patch (green). After that, the votes from all patches are aggregated into a Hough space (c), the faces can be detected as a peak in this image (d).

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 scale vector. The resulting face detection 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. 3. In this work, the first two channels contain the pixel values and normalized ones to avoid the effect of illumination and the rest of channels are the first and second derivatives in x,y directions, and the HOG descriptors respectively.

Fig. 4: Training face samples of different view, expression and poses.

5 Experimental results

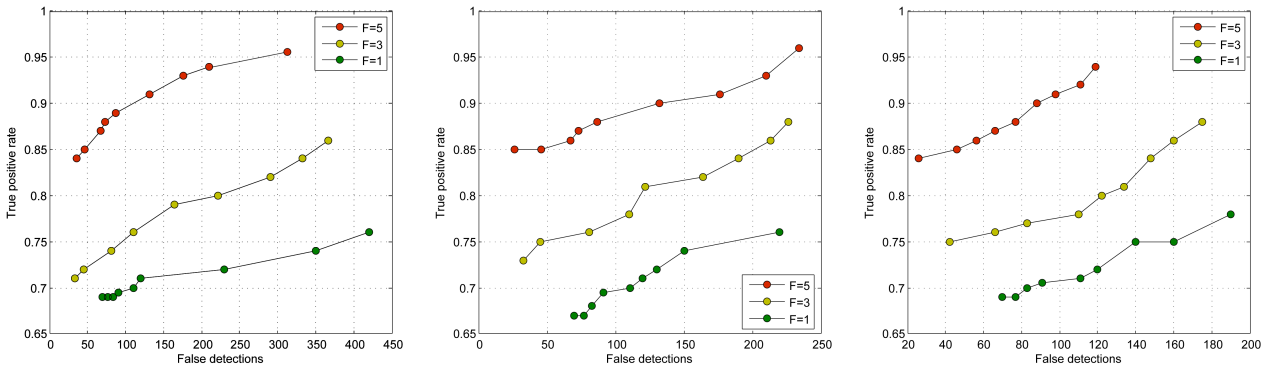
5.1 Performance evaluation measure

Several measures are used to evaluate the performance of face detection systems [2]. Actually, in measuring the performance of a face detection method, 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. Most face detection methods include a threshold, which can be varied to lie at different points in the trade-off between correct and false detections. One method for expressing the trade-off is the receiver operating characteristics (ROC) curve. It allows a better visualization of the performance of a face detector and eases the comparison between several approaches. The ROC curve plots the true positive rate versus the false positive rate, where

$$\text{True positive rate} = \frac{\text{Number of true positives}}{\text{Total number of positives in dataset}} \quad (11)$$

$$\text{False positive rate} = \frac{\text{Number of false positives}}{\text{Total number of negatives in dataset}} \quad (12)$$

The performance of the proposed face detection method is evaluated using variety of image datasets. In the initial evaluation experiment, the proposed face detector is trained for three different Hough forests trees number with the same setting and training data used in constructing the trees. The first detector is trained for Hough forests of one tree, the second detector is trained for Hough forests of three trees, while the last detector is

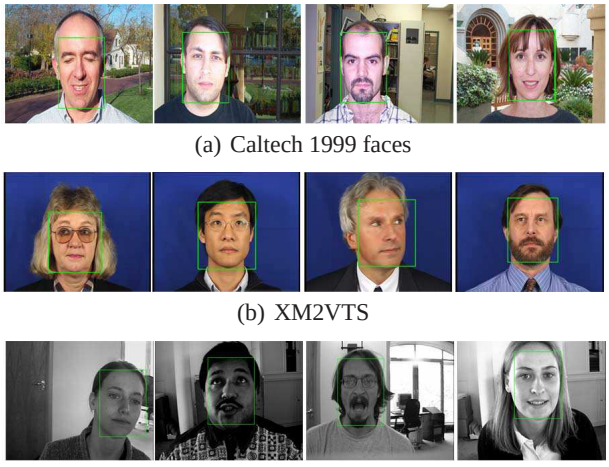


(a) Caltech 1999 (Frontal) (b) XM2VTS (c) BioID

Fig. 5: ROC curves for face detector trained for Hough forests of one, three and five trees tested on (a) Caltech 1999 (Frontal), (b) XM2VTS and (c) BioID database.

trained for Hough forests of five trees, and ROC curves are generated for each one of the face detector. The training samples consist of frontal face and non-frontal faces. In this paper, training stage contains two training data sets, one is for face training dataset and the other for non-face training dataset. For each training data set, we performed a careful selection of images that represent the face training set, as much as possible, the variations of faces. We use faces of males and females, with different ages, of different races, with and without structural components such as glasses and beard, and different lighting conditions and sources, the face training set contains 500 face images cover the different face pose with images of size 85×85 pixels. Also, we added non-face images by randomly selecting regions in images without faces. The training face data set is cropped from images of FDDB database [27]. Figure 4 shows samples of face training data. While non-face training set which were cropped manually and collected by randomly sampling non-face regions of images at Caltech 1999 (Frontal) face database [28], and FDDB database. The non-face training set contains 2,750 images of 35×35 pixels resolutions. We compute the following feature channels Γ: 3 color channels of the CIELAB color space, the absolute values of the two first and two second order derivatives x, y, and the 9 HOG-like channels. Each HOG-like channel was obtained as the soft bin count of gradient orientations in a 5×5 pixels. To increase the invariance under noise, we apply the min and the max filtration, 16 channels for the min filter and 16 channels for the max filter.

First, we evaluate our system with three different databases stressing real world conditions. First, Caltech 1999 (Frontal) face [28] contains 450 face images. The average image resolution is 896×592 pixels. Images in this database are with different lighting, expressions, and backgrounds. We compare the three different trained face



(a) Caltech 1999 faces

(b) XM2VTS

(c) BioID

Fig. 6: Qualitative results on the three databases:(a) Caltech 1999 (Frontal), (b) XM2VTS, (c) BioID.

Hough forests. Second database is the XM2VTS database [29] collected for research and development of identity verification systems. The database contains 295 subjects, each recorded at four sessions over a period of four months. At each session two head rotation shots and six speech shots (subjects reading three sentences twice) were recorded. The third one is the BioID database [30], which consists of 14051 gray scale images of 1199 individuals in front of a uniform background, with views ranging from frontal to left and right profiles. It is designed at first to develop and evaluate face recognition algorithms, but it also can be used to train and test face detection algorithms. Figure 5 shows the ROC curves of the method for three different trained forests number. As

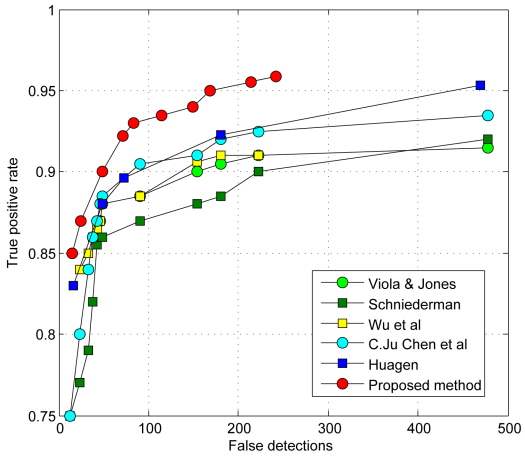


Fig. 7: Comparisons with state of the art methods on CMU database.

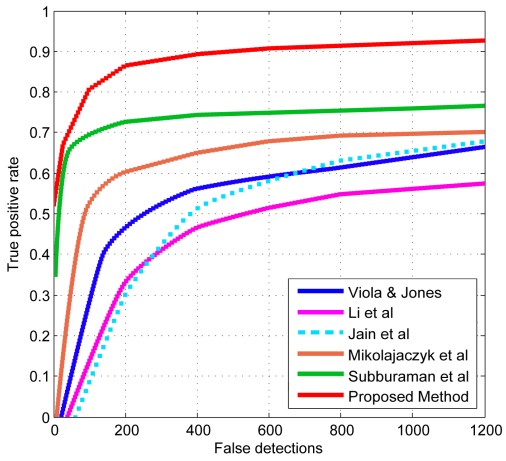


Fig. 8: Comparisons with stat of the art methods on FDDB database.

it is clear that the detection rate increases with the increasing in the number of trees in the trained forests, while the false detections number is decreased. From this experiment, we conclude that the performance of the face detector based Hough forests with trained of five trees has the highest detection rate (i.e., $F = 5$). Some of the qualitative results on the three databases for $F = 5$ are shown in Fig. 6.

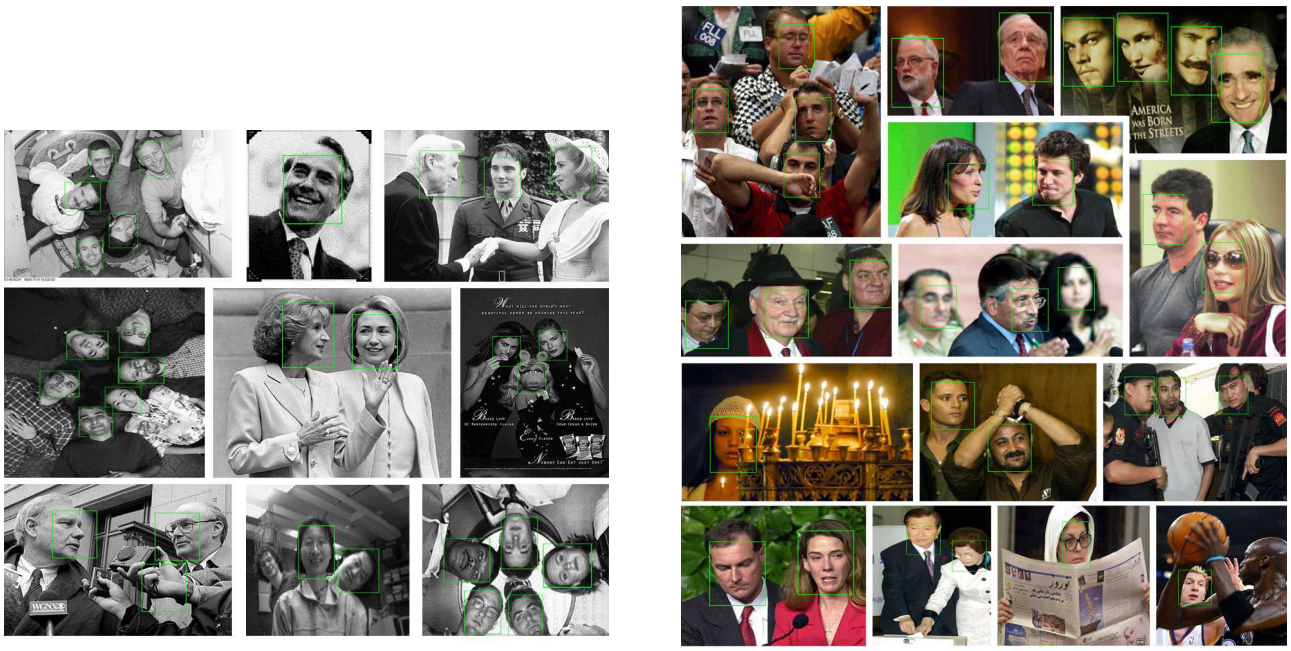
In order to compare our face detection method with state-of-the-art methods, we use same databases used in testing these methods, because we do not have the source code of these methods, as well as to avoid the problem of optimize the parameters of these method. Therefore, other two widely used databases; CMU [31] and FDDB databases are used in this comparison. The proposed face detector based Hough forests is trained with five trees. We compare our results with existing face detection methods such as Viola-Jones face detector [9], Schniederman [31], Wu et al. [32], Huagen [33], and Chen et al [12] using CMU face database, our method has the highest detection rate, with decreasing in the false detections compared to the other methods as shown in Fig. 7. In the case of the FDDB database, the FDDB supporting website has an evaluation toolkit that is based on two types of detection scores: the discrete score is 1 if the ratio of the intersection of a detected region with an annotated face region is greater than 0.5 and 0 otherwise, and the continuous score takes the intersection ratio itself. We adopt the same evaluation discrete score criterion that represents the degree of matching between a detection bounding-box ($\mathcal{B}_i$) and ground truth (G_j) by using the ratio of intersected regions to joined regions as:

$$M(\mathcal{B}_i, G_j) = \frac{A(\mathcal{B}_i) \bigcap A(G_j)}{A(\mathcal{B}_i) \bigcup A(G_j)} > 0.5 \qquad (13)$$

We compared the proposed face detection method with other existing face detection methods such as Viola-Jones, Li et al. [34], Jain et al. [35], Mikolajaczyk et al. [36], and Subburaman et al. [37]. For Viola and Jones' detector, the implementation of OpenCV 2.4.2 [38] with the default face classifier configuration (i.e., *haarcascade_frontalface_default.xml*) is used. The curves of other methods are taken from their published papers without any modification. The proposed method achieves the highest detection rates but lower than this using the CMU database because the face images in the FDDB database have higher variations in pose, illumination, expression, and occlusion than those in the CMU database. Also, the false detections is higher than other methods in the case of CMU database. The ROC curves of this comparison are shown in Fig. 8. Examples of detecting faces form CMU and FDDB databases using the proposed face detection method are shown in Fig. 9.

6 Conclusions

This paper address one of the most difficult task in pattern recognition; namely face detection. It 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. Hough forests are capable to handle large training datasets, high generalization power, fast computation, and ease of implementation. A thorough experimental evaluation is conducted on various benchmark databases for face detection and the results are compared to the existing state of the art. Our method consistently achieves comparable or better results across all experiments than state-of-the-art face detection approaches. There is still a room to further improve the detection performance, so



(a) CMU (b) FDDB

Fig. 9: Examples of detecting faces using the proposed face detection method in (a) CMU and (b) FDDB databases.

our future work includes using non-maxima suppression that can be combined with Hough forests to improve the detection results. Exploiting the relations between sliding window and Hough-based face detection is another promising approach for improving the detection accuracy.

Acknowledgement

The owners of all databases that are used in the experiments in this research; namely, BioID, Caltech, XM2VTS, CMU, and FDDB are gratefully acknowledged.

References

[1] M. Castrillón, O. Déniz, D. Hernández , J. Lorenzo, A comparison of face and facial feature detectors based on the Viola-Jones general object detection framework, Machine Vision and Applications, **22**, 481-494, 2011.

[2] M. Hassaballah, Kenji Murakami, and Shun Ido, Face detection evaluation: A new approach based on the golden ratio Φ, Signal, Image and Video Processing (SIViP) Journal, **7**, 307-316, 2013.

[3] M. Yang, D. Kriegman, and N. Ahuja, Detecting faces in images: A survey, IEEE Trans. on Pattern Analysis and Machine Intelligence, **24**, 34-58, 2002.

[4] W-K. Tsao, A.J.T. Lee, Y-H. Liu, T.-W. Chang, H-H. Lin, A data mining approach to face detection, Pattern Recognition, **43**, 1039-1049, 2010.

[5] C. Huang, H. Ai, Y. Li, S. Lao, High-performance rotation invariant multiview face detection, IEEE Trans. on Pattern Analysis and Machine Intelligence, **29**, 671-686, 2007.

[6] A. M. Burton, M. Bindemann, The role of view in human face detection, Vision Research, **49**, 2026-2036, 2009.

[7] E. Hjelmas and B. Low, Face detection: a survey, Computer Vision and Image Understanding, **83**, 236-274, 2001.

[8] Z. Tabatabaie, R. Rahmat, N. B. Udzir, and E. Kheirkhah, A hybrid face detection system using combination of appearance-based and feature-based methods, International Journal of Computer Science and Network Security, **9**, 181-185, 2009.

[9] P. Viola and M. J. Jones, Robust real-time face detection, International Journal of Computer Vision, **57**, 137-154, 2004.

[10] S. C. Brubaker, J. Wu, J. Sun, M. Mullin, and J. Rehg, On the design of cascades of boosted ensembles for face detection, International Journal of Computer Vision, **77**, 65-86, 2008.

[11] L. Xiaohua, K.-M. Lam, S. Lansun, and Z. Jiliu, Face detection using simplified Gabor features and hierarchical regions in a cascade of classifiers, Pattern Recognition Letters, **30**, 717-728, 2009.

[12] S. Li and Z. Zhang, Floatboost learning and statistical face detection, IEEE Trans. on Pattern Analysis and Machine Intelligence, **26**, 1112-1123, 2004.

[13] J.-C. Chen and J.-J. James Lien, A view-based statistical system for multi-view face detection and pose estimation, Image and Vision Computing, **27**, 1252-1271, 2009.

[14] M. Yang, J. Crenshaw, B. Augustine, R. Mareachen, and Y. Wu, Adaboost based face detection for embedded systems, Computer Vision and Image Understanding, **114**, 1116-1125, 2010.

[15] C. Liu, A Bayesian discriminating features method for face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, **25**, 725740, 2003.
[16] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, H. Klviinen, and J. Matas, Feature-based affine-invariant localization of faces, IEEE Trans. on Pattern Analysis and Machine Intelligence, **27**, 1490-1495, 2005.
[17] M. Hassaballah, Tomonori Kanazawa, S. Ido, and S. Ido, An efficient eye detection method based on gray intensity variance and independent components analysis, IET Computer Vision Journal, **4**, 261-271, 2010.
[18] J. Gall and V. Lempitsky, Class-specific Hough forests for object detection, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), Miami, Florida, USA, 20-25 June, pp. 1022-1029, 2009.
[19] B. Leibe, A. Leonardis, and B. Schiele, Robust object detection with interleaved categorization and segmentation, International Journal of Computer Vision, **77**, 259-289, 2008.
[20] A. Bosch, A. Zisserman, and X. Muñoz, Image classification using random forests and ferns, Proc. IEEE International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brazil, 14-20 October, pp. 1-8, 2007.
[21] F. Schroff, A. Criminisi, and A. Zisserman, Object class segmentation using random forests, British Machine Vision Conference (BMVC'08), Leeds, UK, 1-4 Sep., 2008.
[22] J. Shotton, M. Johnson, and R. Cipolla, Semantic texton forests for image categorization and segmentation, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), Anchorage, Alaska, USA, 23-28 June, pp. 1-8, 2008.
[23] J. M. Winn and J. Shotton, The layout consistent random field for recognizing and segmenting partially occluded objects, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06), NY, USA, 17-22 June, pp. 37-44, 2006.
[24] E. Parzen, On estimation of a probability density function and mode, Annals of Math. Statistics, **33**, 1065-1076, 1962.
[25] O. Barinova, V. Lempitsky, and P. Kohli, On the detection of multiple object instances using Hough transforms, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), San Francisco, CA, USA, June 13-18, pp. 2233-2240 , 2010.
[26] N. Razavi, J. Gall, and L. Van Gool, Scalable multi-class object detection, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), Colorado Springs, USA, 20-25 June, pp. 1505-1512, 2011
[27] V. Jain and E. Learned-Miller, FDDB: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.
[28] Available from: http://www.vision.caltech.edu/archive.html
[29] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, XM2VTSDB: The extended M2VTS database, Proc. Second International Conference on Audio and Video-based Biometric Person Authentication, 1999.
[30] O. Jesorsky, K. Kirchberg, and R. W. Frischholz, Robust face detection using the hausdorff distance. In Third International Conference on Audio and Video-based Biometric Person Authentication, Lecture Notes in Computer Science, pages 9095. Springer, 2001
[31] H. Schneiderman and T. Kanade, A Statistical method for 3D object detection applied to faces and cars, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hilton Head Island, South Carolina, USA, 2000.
[32] B. Wu, H.Z. Ai, C. Huang, S.H. Lao, Fast rotation invariant multi-view face detection based on real AdaBoost, Proc. of Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 79-84.
[33] C. Huang, H.Z. Ai, Y. Li, S.H. Lao, Vector boosting for rotation invariant multi-view face detection, Proc. of IEEE International Conference on Computer Vision (CVPR'05), **1**, 2005, 446-453.
[34] J. Li, T. Wang, and Y. Zhang, Face detection using SURF cascade, Proc. IEEE Intel Conf. Computer Vision Workshops, pp. 2183-2190, 2011.
[35] V. Jain and E. Learned-Miller, Online domain adaptation of a pre-trained cascade of classifiers, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 577-584, 2011.
[36] K. Mikolajczyk, C. Schmid, and A. Zisserman, Human detection based on a probabilistic assembly of robust part detectors, Proc. European Conf. Computer Vision, pp. 69-82, 2004.
[37] V. Subburaman and S. Marcel, Fast bounding box estimation based face detection, Proc. ECCV Workshop Face Detection: Where We Are and What Next, 2010.
[38] Intel, Intel Open Source Computer Vision Library, v2.4.2, http://sourceforge.net/projects/opencvlibrary/

Mourad Ahmed received his B.Sc. degree in Computer Science in 2005 from Faculty of Science, South Valley University. He is an assistance lecturer at the Department of Mathematics, Faculty of Science Egypt. He is currently working toward the M.Sc. degree in Computer Science at Faculty of Science, where he is coadvised by Prof. A.S. Abdel Rady and Dr. M. Hassaballah. His research interests include Artificial Intelligence, face detection, and pattern recognition.

M. Hassaballah was born in 1974, Qena, Egypt. He received the B.Sc. degree in Mathematics in 1997, then M.Sc. degree in Computer Science in 2003, all from South Valley University, Egypt. His M.Sc. research was in the area of fractal image compression. From 1997 to 2007, he worked for the Department of Mathematics, South Valley University, Egypt. In April 2008, he joined the lab of Intelligence Communication, Department of Electrical and Electronic Engineering and Computer Science, Ehime University, Japan as a PhD student, where he received the Doctor of Engineering (D.Eng.) in Computer Science on September 2011 for his work on facial features detection. He is currently a lecturer of Computer Science at the Faculty of Science, South Valley University, Egypt. His research interests include: image processing, computer vision, facial feature extraction, face detection and recognition, object detection, image retrieval, similarity measures, fractal image compression, and high performance computing.

Yasser Salama Hassan was born in 1974, Qena, Egypt. He received a B.Sc. degree in Mathematics in 1996, then M.Sc. degree in statistic in 2007, all from South Valley University, Egypt. His M.Sc. research was in the area of Bayesian Estimation. He is an assistance lecturer at the Department of Mathematics, Faculty of Science, South Valley University, Egypt. He is currently working toward the Ph.D. degree in Applied Statistics at Faculty of Science, South Valley University. His research interests include: bootstrap, random forests, Bayesian estimation, applied statistics, probability, pattern recognition, face detection.

A. H. Abd Ellah Professor of Mathematical statistics, Faculty of Science, Sohag University, Egypt. He supervised many Master and Doctorate students in Mathematics and Computer Science. He has authored several peer-reviewed publications. His research interests include: Mathematical statistics, prediction, estimation,order statistics, application of statistics in computer science.

A. S. Abdel Rady was born in 1942, Minia, Egypt. He received the B.Sc. degree in Mathematics and Education in 1962 from Ministry of Institution, Assiut, Egypt, the B.Sc. degree in Special Mathematics in 1970 from Assiut University, Egypt, then the Ph.D. in Mathematics, Differential Equations in 1977 from by Moscow State University, Moscow, USSR. During his carrier he headed the Department of Mathematics and Vice-Dean for Higher Studies and Researches at Faculty of Science, South Valley University. He has authored several peer-reviewed publications, and supervised many Master and Doctorate students in Mathematics and Computer Science. Currently, he is an Emeritus Professor with the Department of Mathematics, South Valley University, Egypt. His main research focus is on Differential Equations and their applications.