

# Multi-View Face Detection in Open Environments using Gabor Features and Neural Networks

R. Mohammadian Fini<sup>1</sup>, M. Mahlouji<sup>2\*</sup> and A. Shahidinejad<sup>1</sup>

1. Department of Computer, Faculty of Engineering, Islamic Azad University, Qom Branch, Qom 3749113191, Iran.

2. Department of Telecommunications, Kashan Branch, Islamic Azad University, Kashan, Iran.

Received 25 August 2019; Revised 31 March 2020; Accepted 28 April 2020

\*Corresponding author: mmahlouji@yahoo.com (M.Mahlouji).

## Abstract

Multi-view face detection in open environments is a challenging task due to the wide variations in illumination, face appearances and occlusion. In this paper, a robust method for multi-view face detection in open environments is presented, using a combination of Gabor features and neural networks. Firstly, the effect of changing the Gabor filter parameters (orientation, frequency, standard deviation, aspect ratio, and phase offset) for an image is analyzed. Secondly, the range of Gabor filter parameter values is determined. Finally, the best values for these parameters are specified. A multi-layer feedforward neural network with a back-propagation algorithm is used as a classifier. The input vector is obtained by convolving the input image and a Gabor filter, with both the angle and frequency values equal to  $\pi/2$ . The proposed algorithm is tested on 1,484 image samples with simple and complex backgrounds. The experimental results show that the proposed detector achieves a great detection accuracy, by comparing it with several popular face-detection algorithms such as the OpenCV's Viola-Jones detector.

**Keywords:** Facial Features, Standard Division, Gabor Energy, Face Components.

## 1. Introduction

Face detection is the earliest and the most important step in processes such as face recognition. Face detection in an image has been a challenging area of research, due to images with multiple faces, the variability in human faces with regard to skin-colour (pinkish, yellowish, etc.), variability in expression (smiling, crying, etc.), illumination (outdoor or indoor lighting conditions, etc.), occlusion (face partially covered by long hair or sunglasses, etc.), and different poses (frontal or profile orientation, etc.) [1].

Over time, the researchers have used various methods to improve the quality of face area detection, reducing the impact of these challenges. These methods can be divided into two main groups. The first group is focused on extracting different types of hand-crafted features using domain experts in computer vision. In what follows, we will discuss some of these studies. One of the earliest and most popular studies in this group presented the Viola-Jones algorithm [2]. This study describes a face detection algorithm that

is capable of processing images rapidly while achieving high detection rates using three key contributions: a new image representation called the "integral image", a learning algorithm based on AdaBoost and a method for combining classifiers in a cascade. In [3], a face occlusion detection algorithm for ATM (Automated teller machine) surveillance has been presented. This scheme has two parts including face detection and verifying whether a detected face is occluded or not. For the face detection part, an energy function is used for elliptical head contour detection, together with a head-tracking algorithm that utilizes gradient and shape cues in a Bayesian framework. In [4], an AdaBoost-based detector has been trained using only the frontal face data, and it is able to detect in-plane and out-of-plane rotated faces without requiring training data from different in-plane or out-of-plane rotation angles. For in-plane rotated faces, this scheme measures the angle  $\theta$  between the principal component and the y-axis, and then it rotates the in-plane rotated face with bias  $\theta$  into a

frontal view of the face. For out-of-plane rotated faces, a flipped technique is used to create a face similar to the frontal one, and then the frontal face detector is used. The skin-colour feature is another type of hand-crafted feature. This type has a poor efficiency in images with low intensity and for faces with make-up or changes in facial skin colour but has the two advantages of high-speed image processing and easy implementation. An approach for face detection under varying illumination has been presented in [5]. This detection method initially takes a colour image, and then creates a colour-balance model to transform the RGB colour space to a YCgCr colour space. The algorithm is implemented by combining a Gaussian skin-colour model, template matching, and face verification.

The second group is focused on deep learning, especially deep convolutional neural networks (CNN). The studies in [6,7] are in this group. In [6], a face detection method has been presented that extends the faster region-based CNN (RCNN) algorithm. This scheme improved the existing faster RCNN scheme by combining several important strategies including feature concatenation, hard negative mining, and multi-scale training. In [7], a fast CNN cascade face detector has been used with multiple task learning, and network acceleration techniques. The first stage of the detector is an elaborately designed fully convolutional network with a novel pyramid architecture, which can generate multi-scale face proposals efficiently, with no more than two image resizing operations.

In this paper, as in the studies in first group, a robust method is proposed for face detection from different views using a combination of 2D Gabor filters and neural networks. To this end, we first analyze the impact of each one of the Gabor filter parameters on an image, and then perform an experimental determination of the best values of these parameters for use in face detection. Precise determination of these values has an essential role in reducing computational complexity, which is one of the main aims of the proposed method.

### 1.1. Motivation

In the recent years, Gabor filters have often been used for face recognition. A face recognition application is a three-stage computer program including face detection, feature extraction, and face identification. Face recognition is a problem with a high computational complexity, requiring a significant commitment of computer resources. The use of a common tool for face detection and feature extraction can reduce the computational complexity of the program. In this paper, the main

motive for studying face detection using Gabor methods is to reduce the computational complexity of face recognition, a topic that can be addressed in the future work.

### 2. Related work

One of the earliest papers for face detection using Gabor filters [8] has proposed a hierarchical face detector using an AdaBoost algorithm and combining Haar features and Gabor features. It ran AdaBoost in 22 stages, with the first 21 stages composed of weak classifiers based on Haar features, while the last stage consisted of weak classifiers based on Gabor features. In this study, Gabor filters with five scales and eight orientations were used. The number of Gabor features in each sample was 16,000, from which the training algorithm selected 96 of the most discriminant ones, to form the last stage of the cascade detector. This scheme has a detection rate of 90.37% with eight false alarms on the MIT and CMU frontal face dataset. In [9], a classification-based face detection method has been presented using a polynomial neural network and Gabor filter features. In this study, Gabor features were generated from four orientations and one radial frequency. In the approach, all four Gabor orientation representations were concatenated to construct a feature vector. In order to reduce the complexity, the Gabor representation was downsampled by a factor  $q = 2$  before the concatenation, and then the raw vector projected onto a linear sub-space with size of 100, via PCA. This had a detection rate of 86.4% with 57 false alarms on the CMU dataset and 100% with three false alarms on 270 Internet images with simple backgrounds. In [10], a neural network-based approach has been presented for finding frontal faces using Gabor features and convolutional neural networks. Gabor features were generated using two different orientations and two different wavelengths. In the system, a skin-colour detector is used as the first classifier. This has a detection rate of 87.5% on an image dataset obtained from Internet images of very good quality. Face detection in [11] has been done using Gabor wavelets and a nearest feature space classifier. The approach firstly generates feature vectors of faces using 40 Gabor wavelets, and then reduces the feature space using the principal component analysis, and the linear discriminant analysis, and finally, classifies face and non-face samples using the nearest feature space classifier. This has a detection rate of 93% on the ORL dataset. In [12], it has been stated that in the face detection approach based on AdaBoost for reducing non-face sample

detection errors, it is necessary to increase the node depth, which causes the complexity of the cascade to increase rapidly, although the performance barely improves. The study tackles this problem using contextual information about faces in the selection of features for the deeper nodes. In order to better represent the contextual information, it uses simplified Gabor wavelets. The simplified Gabor wavelet is a version of the Gabor wavelet where the values are quantized to a certain number of levels and can be used for feature extraction via the integral image method. In [12], four orientations and three frequencies have been used for simplified Gabor feature extraction. This has a detection rate of 99.77% with 69 false alarms on the FERET dataset, 99.41% with 83 false alarms on the BIODID dataset and approximately 96% with 165 false alarms on the MIT and CMU dataset. The rest of this paper is organized as what follows. The Gabor filter and Gabor features are introduced briefly in Section 2. The best values for the effective parameters of the Gabor filter bank are obtained in Section 3. The proposed face detection algorithm is described in Section 4. The results of the proposed method on three face databases including FERET [13], Marcus Weber [14], and Internet images are presented in Section 5. In addition, the results of the proposed algorithm for the three face databases are compared with the results of OpenCV's Viola-Jones algorithm [15]. Finally, conclusions are drawn in Section 6.

### 3. Gabor filter and Gabor features

A Gabor function can be viewed as a sinusoidal wave plane of particular frequency and orientation, modulated by a Gaussian envelope. Gabor filters are generated by Gabor functions. Gabor filters are available in three forms, i.e. one-, two- and three-dimensional filters [16,17,18]. According to [19], a 2D Gabor filter can be expressed as in (1):

$$G(x,y)=w(x,y)\times m(x,y) \quad (1)$$

where  $w(x,y)$  is a 2D Gaussian-shaped function and  $m(x,y)$  is a complex exponential. A 2D elliptic Gaussian centred on the origin of a Cartesian coordinate system can be expressed as (2) [20].

$$w(x,y) = \exp\left(-\frac{1}{2}\left(\frac{x_r^2}{\sigma^2} + \frac{y_r^2}{\beta^2}\right)\right) \quad (2)$$

Here,  $\sigma^2$  and  $\beta^2$  are the variances in the 2D Gaussian function along the x-axis and the y-axis, respectively. Furthermore, the  $x_r$  and  $y_r$  variables can be written as:

$$\begin{aligned} x_r &= x \cos \theta + y \sin \theta \\ y_r &= -x \sin \theta + y \cos \theta \end{aligned} \quad (3)$$

Where  $\theta$  is the orientation of the 2D elliptical Gaussian function axis.

According to [21,22], a complex sine wave is generated, as shown in (4).

$$m(x,y) = \exp\left(i\left(2\pi\frac{x_r}{\lambda} + \varphi\right)\right) \quad (4)$$

Here,  $\varphi$  is the phase offset, which determines the symmetry of the function: for values of  $0^\circ$  and  $180^\circ$ , the Gabor filter is symmetric or even, and for values of  $90^\circ$  and  $-90^\circ$ , it is anti-symmetric or odd [22]. In addition,  $\lambda$  is the wavelength in the spatial frequency domain.

In the 2D Gabor function, the ratio of the standard deviations along the x-axis and y-axis, called the aspect ratio, is indicated by the symbol  $\gamma$ . In [16,22], parameter  $\beta$  of the Gabor filter is

removed by considering  $\frac{\sigma}{\beta} = \gamma$ , and also it can be

used as a scale factor  $k = \frac{\gamma}{2\pi\sigma^2}$  in order to generate a Gabor filter of any amplitude [16,20]. Using these changes and using the Euler equation [20], and also considering  $f = \frac{1}{\lambda}$  [23], the Gabor filter function can be expressed as (5).

$$\begin{aligned} G(x,y) &= \frac{\gamma}{2\pi\sigma^2} \times \exp\left(-\frac{x_r^2 + \gamma^2 y_r^2}{2\sigma^2}\right) \\ &\times (\cos(2\pi f x_r + \varphi) + j \sin(2\pi f x_r + \varphi)) \end{aligned} \quad (5)$$

#### 3.1. Gabor features

Gabor features are extracted by convolving the input image and the Gabor filter bank. The Gabor filter bank is a set of filters with different orientations and frequencies [24]. The Gabor filter outputs of the Gabor filter bank have a complex-number format. Therefore, Gabor features can be generated from the real part (symmetric) [25], the imaginary part (antisymmetric) or the complex part of the Gabor filter function [26]. The real part of the Gabor output will give a large response at the centre of symmetric features such as lines. The imaginary part of the Gabor output will give a large response at the centre of anti-symmetric features such as edges.

Symmetric and anti-symmetric outputs of a Gabor filter can be combined in a single quantity called the Gabor energy. This feature can be expressed as

$$e_{\lambda,\theta} = \sqrt{r_{\lambda,\theta,0}^2(x,y) + r_{\lambda,\theta,\frac{\pi}{2}}^2(x,y)}, \quad \text{where } r_{\lambda,\theta,0}^2(x,y)$$

and  $r_{\lambda, \theta, \frac{\pi}{2}}^2(x, y)$  are the responses of the linear symmetric and anti-symmetric Gabor filters with angle  $\theta$  and frequency  $1/\lambda$  respectively [22]. The Gabor energy will give a large response at the locations of both the edge and line features. Figure 1 shows the real, imaginary and energy parts of a Gabor filter output. As it is clear in the figure, the Gabor energy feature reflects the facial components (eyes, nose, and lips) better than the real and imaginary features. Therefore, the proposed method uses the Gabor energy feature.

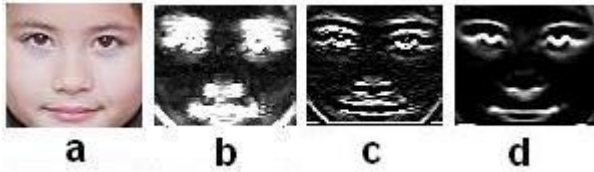


Figure 1. Parts of a Gabor filter output: (a) original image (b) Gabor-energy, (c) Real, and (d) Imaginary. The value of the Gabor filter generation parameters are as follow:

$$f = \frac{\pi}{2}, \sigma = 0.4, \gamma = 1, \theta = \frac{\pi}{2}.$$

#### 4. Setting effective parameters in Gabor filter generation

By analyzing the various references, it can be understood that not only are there no identical values for the parameters of the Gabor filter in different applications but also differences in the values of these parameters exist in similar applications. For example, in [27], it has been shown that the values of the effective parameters of the Gabor filter for face detection are different for different face databases. Differences in the values of these parameters also exist for a face image with different views. It is obvious that in order to obtain a powerful Gabor energy feature for face detection, it is necessary to set the parameters of the Gabor filter banks correctly. Therefore, in the following, the impact of two of the most important Gabor filter parameters (angle and standard deviation) will first be discussed. Then, various effective parameter settings in Gabor filter generation will be presented. Finally, the settings for the Gabor filter parameters in the proposed method will be described.

##### 4.1. Impact of angle and standard deviation parameters in Gabor filter output

The Gabor filter's impact on an input image can be simply represented as colour changes occurring in the input image. The Gabor filter identifies the boundary between two regions with high colour. Two Gabor filter parameters, i.e. angle and standard deviation, are effective in the boundary

determination between regions with different colours. Figure 2 shows the impact of angle in the Gabor filter output. Clearly, the boundary points are black where their angles are the same as the Gabor filter angle. For example, in figure 2(c), the Gabor filter with an angle of  $45^\circ$  has been convolved with the original image. Therefore, every point of the image with an angle of  $45^\circ$  to the horizontal axis is black. In figure 2(f), the Gabor filter with an angle of  $60^\circ$  is convolved with the original image, and there are no black points in the output since there are no points in the original image with an angle of  $60^\circ$  to the horizontal axis.

Figure 3 shows the impact of changing the standard deviation parameter value in the Gabor filter output. The standard deviations in figures 3(b) and 3(c) are 0.4 and  $\pi$ , respectively. It is clear that as the standard deviation value increases, the boundary line becomes thicker.

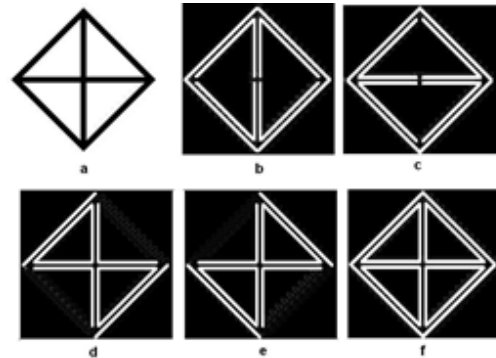


Figure 2. The impact of angle parameter in the Gabor filter output: (a) Original image, the Gabor filter output with the angle value (degrees) of: (b) 0, (c) 90, (d) 45, (e) -45, and (f) 60. The value of the other Gabor filter generation parameters are as follow:  $f = \frac{\pi}{2}, \sigma = 0.5, \gamma = 1$

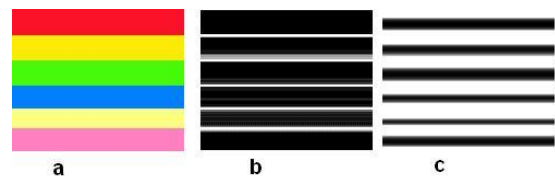


Figure 3. The Impact of standard deviation parameter in the Gabor filter output: (a) Original image, the Gabor filter output with the standard deviation value of: (b) 0.4 and (c)  $\pi$ . The values of other Gabor filter generation

$$\text{parameters are as follow: } f = \frac{\pi}{2}, \gamma = 1, \theta = 0$$

##### 4.2. Analysis of Gabor filter parameter setting methods in face detection

As it can be seen in (7), five parameters, i.e. the angle, frequency, phase offset, and standard deviations along the x-axis and y-axis, must be set to generate the Gabor filter function. Among these parameters, the phase offset, used to generate the

real and imaginary parts of the Gabor filter, is essentially constant, so this parameter value is set to either zero or  $\pi$ , or to  $\pi/2$  or  $-\pi/2$ , respectively [22].

In the proposed method, the phase offset parameter value for the real and imaginary parts is set to 0 and  $\pi/2$  respectively. Table 1 shows the typical values for other Gabor filter parameters. These values are frequently used for face detection via Gabor filters. By analyzing this table, the following results can be discovered concerning the Gabor filter parameters.

The angle parameter often has eight orientations, and its value lies between 0 and  $\frac{7\pi}{8}$  radians. This

parameters can be obtained by  $\theta = \pi \times \frac{i}{8}$

where  $i=1,2,\dots,7$ .

In a number of studies, the frequency parameter value is obtained by  $f = \frac{f_{max}}{f_{rate}^i}$  where  $f_{max}$  is the

maximum frequency,  $f_{rate}$  is the ratio of consecutive frequencies, and  $i=1,2,3,\dots,f_n$ , where  $f_n$  is the number of frequencies.

The standard deviation parameter can be set using various methods. One group of studies initializes the two standard deviations along the x-axis and y-axis [27,28]. Another group of studies uses standard deviations along the x-axis and y-axis using the gamma parameter [16]. Another group utilizes the same value for the standard deviations along the x-axis and y-axis [9,11,12,21]. In this group, these parameters are related to wavelength [16,22]. As it can be seen in table 1, the differences in Gabor filter parameter settings for face detection are often related to the standard deviation parameter value. For this reason, in the next section, the optimal value of this parameter is first obtained for the proposed face detection method.

**Table 1. Some of the effective parameters and their values in the Gabor filter generation used in different papers.**

Sources	Face database	Angle			Frequency		Standard Deviation	Aspect ratio (gamma)	
		No.	Range	Step	No.	Max Step			
[22]	CMU Multi	8	0- $\pi$	$\pi/8$	4	$\lambda = \alpha \times \sigma$ $\alpha = \{0.25, 0.57, 1, 2\}$	$\sigma = \{2, 3.5, 3.5, 2.5\}$	$\gamma = \{1.25, 0.75, 1, 1.5\}$	
[25]	JAFFE & CK	8	0- $\pi$	$\pi/8$	3	$f=1.25 \times \sigma$	$\sigma = \{4.46, 8.62, 13.35\}$	0.5	
[21]	Special	8	0- $\pi$	$\pi/8$	5	$\Pi$	$\sqrt{2}$	$\sigma = \pi$	No
[27] <sup>a</sup>	Special	8	0- $\pi$	$\pi/8$	5	$\pi/2$	$\sqrt{2}$	$\sqrt{2}$	No
[27]	FRGC	6	0- $\pi$	$\pi/6$	5	$\pi/2$	$\sqrt{2}$	$\sigma_x = \sigma_y = \sqrt{2}$	No
[27]	FERET	8	0- $\pi$	$\pi/8$	5	$\pi/2$	$\sqrt{2}$	$\sigma_x = 1, \sigma_y = \sqrt{2}$	No
[27]	XN2VTS	8	0- $\pi$	$\pi/8$	5	$\pi/2$	$\sqrt{2}$	$\sigma_x = \sigma_y = 1$	No

<sup>a</sup> Classical bank

### 4.3. Determining effective parameter values for generating Gabor filter in proposed method

In this section, the procedure for obtaining the effective parameters for Gabor filter generation in the proposed method is first explained. In the following, the reason for selecting a Gabor filter with an angle of  $\pi/2$  and a frequency of  $\pi/2$  for face detection is described, and the Gabor filter performance, tested on different sizes of images, is presented.

In the proposed method, the sigma parameter is independent and is not related to the frequency or filter size. In addition, the values of the other four parameters, i.e. the number of orientations, the maximum frequency, the ratio of consecutive frequencies, and the number of frequencies, are chosen according to the most frequent values of

these parameters in table 1, and are 8,  $\pi/2$ ,  $\sqrt{2}$ , and 5, respectively.

In order to obtain the ranges of two parameter values (sigma and gamma), these parameters are initialized to values of  $\{0.5, 1, 1.9, 3.14, 6.28\}$  and  $\{0.5, 1, 2, 2.5, 5\}$ , respectively, and then  $5 \times 5 = 25$  different Gabor filter banks are generated. The generated Gabor filter banks are convolved with a face image sample of size 70\*70 pixels. The Gabor filter bank with a sigma value greater than 0.5 produces no appropriate output, most of the filters produce white outputs, row one and column two of figure 6(b) shows a sample of these white outputs. After reviewing the gamma parameter, it is found that the Gabor filter bank with a gamma value greater than 2 makes most of the Gabor filter outputs black, and the value of this parameter must be between zero and 2.



This review indicates that the values of the sigma and gamma parameters should be less than 0.5 and 2, respectively. In order to obtain the desirable values for the sigma and gamma parameters, these parameters are initialized to values of {0.1, 0.2, 0.3, 0.4, 0.5} and {0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2}, respectively, and  $5 \times 10 = 50$  new Gabor filter banks are generated.

According to the results obtained, the best values of the gamma and sigma parameters are  $\sigma=0.4$  and  $\gamma=1$ . Figure 4(a) shows the best Gabor filter bank from among 75 different Gabor filter banks. Figure 4(b) shows the classical Gabor filter bank implemented according to table 1. As it can be seen in this figure, some of the classical Gabor filter outputs are almost white, while the outputs of all 40 Gabor filters in the proposed method show all the face components completely.

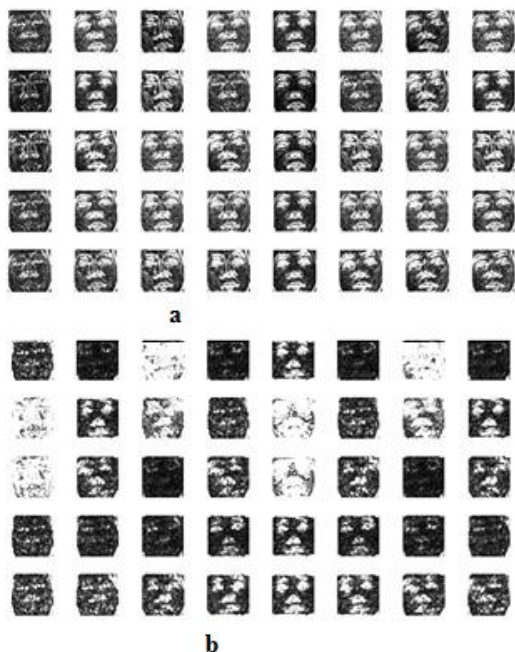


Figure 4. (a) The best Gabor filter bank output in the proposed method after setting the values of sigma and gamma parameters accurately. The values of other Gabor filter generation parameters are as follow:

$$f = \frac{\pi}{2\sqrt{2}^{\sigma}} : \frac{\pi}{2\sqrt{2}^{\gamma}}, \sigma = 0.4, \gamma = 1, \theta = 0 : \frac{7\pi}{8}$$

(b) The classical Gabor filter bank implementation.

#### 4.4. Suitable Gabor filter selection

In order to achieve face detection using Gabor filters, any of the Gabor filters or a combination of two or more of them can be used. Using the latter combination increases the computational complexity and decreases the detection rate. Therefore, in the proposed method, only one Gabor filter has been used. In order to select the best filter, it is necessary to note that the orientation of the eyes, lips, and nose holes are horizontal, and the

orientation of the nose is vertical. As explained in Section 3.1, boundary points are black where their angles are the same as the Gabor filter angle in the Gabor filter output. However, the orientations of eyes, lips, and nose holes are horizontal. Consequently, the Gabor filter with an angle of  $90^{\circ}$  detects face components better than filters with angles of  $0-90^{\circ}$ . For this reason, the proposed method uses a Gabor filter with an angle of  $\pi/2$  and a frequency of  $\pi/2$  for face detection.

Figure 5 show the Gabor filter outputs in the proposed method on some example images. In the figure, parts (a), (b), and (c) are the original image, the Gabor energy, and the Gabor energy converted to a black-and-white image, respectively. Looking closely at figure 5, it can be seen that the Gabor filter can easily detect any face area with skin colours of white, tan, yellow or black. Similarly, it can detect, non-virtual faces and faces with low intensities.

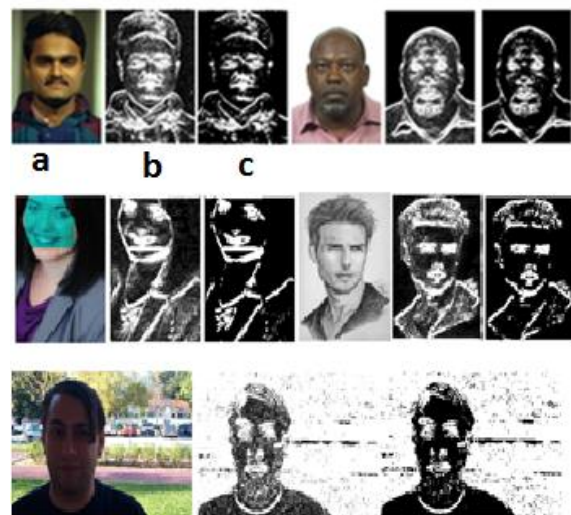


Figure 5. The ability of the selected Gabor filter on the images with different skin colors, virtual and non-virtual images and with low intensities in face area: (a) original image, (b) the Gabor-energy, and (c) the Gabor-energy converted to black and white image.

#### 5. Face detection in proposed method

Face detection in the proposed method uses Gabor features and a neural network. The neural network used in the proposed method is a multi-layer perceptron feedforward back-propagation network. The neural network has three layers, i.e. the input layer, the hidden layer, and the output layer. The hidden layer was initialized with 100 neurons. The transfer function of all layers in the neural network is a tangent sigmoid function. The number of neurons in the input layer depends on the size of the binary input data, i.e. the Gabor filter output converted to a black-and-white image. In the following, firstly, the algorithm of the proposed

face detection method is described, and then the performance of the proposed method on 1,484 images is presented.

### 5.1. Proposed face detection algorithm

Figure 6 shows the face detection algorithm proposed in this paper. As it can be seen, this algorithm includes three procedures called Mask\_gen (line of 1), Image\_seg (line of 18), and Face\_det (line of 33). Mask\_gen procedure receives the Gabor filter parameters including frequency, angle, aspect ratio, standard deviation, and mask size, and generates a Gabor filter mask. Image\_seg procedure receives an image and divides it into a number of sub-images with sizes of 60×60, 50×50, 40×40 and 35×35. Creating sub-images with different sizes helps us to detect faces at different distances from the camera. Face\_det procedure is responsible for applying Gabor filter mask to the input image and detecting faces of it. To this purpose, firstly, a threshold value is assigned to the NN variable (lines of 36-46). When the output of the neural network is greater than the threshold value, the input image is a face. The threshold values in the FERET, Markus Weber, and RFD1 databases are 0.86, 0.93, and 0.93, respectively, since a Gabor filter bank cannot be used on images with different sizes. In the proposed method, size of 192\*128 and 148\*224 pixels has been selected as optimal size for images in the FERET and Marcus Weber databases, respectively (lines of 38 and 42). Then the input image is converted to a grayscale image, and it is convolved with the selected Gabor filter (lines of 47 and 48). In the next step, the Gabor-energy feature is computed and the feature is converted to a black-and-white image (lines of 49 and 50). Using Image\_seg procedure, the black-and-white image is segmented into sub-images with different sizes. Each one of these segmented images is an input of the neural network. In Face\_det procedure, SR\_NN function simulates the response of artificial neural network for a sub-image. When the output of the neural network is greater than the threshold value, the sub-image is a face; otherwise, the sub-image is not a face (lines of 53-58). At this stage, the proposed algorithm can detect any face with different views and different sizes. Figure 7 shows the proposed algorithm output on some images with different views and different sizes. Images with low intensity in the face area (such as row 3 in Figure 5) is a challenge for face detection approaches. As mentioned in Section 4.1, increasing the standard deviation results in an increase in the thickness of the boundary line. Thus to overcome the challenge, in the proposed method,

the value of the sigma parameter is increased from 0.4 to 0.7, and then a new Gabor filter mask is generated, and lastly, the detection process is repeated (lines 59-71).

### 5.2. Performance evaluation of the proposed algorithm on 1,484 images with different Views

In order to evaluate the performance of the proposed algorithm, three sets of images are collected. The first set is a collection of 300 images from the Markus Weber database. The second set is a collection of 250 images including 217 non-repetitive face images and 33 repetitive face images from the FERET database with different distances from the camera and view angles between -15° and +30°. The third set is a new face image database called RFD1. This database includes 112 multi-face images collected from the Internet. RFD1 images are segmented into images of size 192\*128, and contains 934 face images including 577 male images and 357 female images. This image database includes 652 non-repetitive and 282 repetitive face images with different backgrounds. It will use to analyze the performance of the proposed method for face detection on images captured in unconstrained situations.

Table 2 shows the accuracy of the proposed face detection method. In the table, the following parameters are used: TSWN: total sub-window number, DN: detection number, DR: detection rate, FAN: false acceptance number, FRN: false rejection number, FAR: false acceptance rate, and FRR: false rejection rate. The table shows that the proposed algorithm has a face detection rate of 95% for the Markus Weber database, with a threshold value of 0.93. In addition, the proposed algorithm can carry out face detection with an accuracy of 99.66% and an FAR of 1.4E-5 in this image set with a threshold value of 0.76. The proposed algorithm has a face detection rate of 98.4% for the FERET database, with a threshold value of 0.86. In addition, the proposed algorithm can carry out face detection with an accuracy of 99.66% and an FAR of 2.7E-7 in this image set with a threshold value of 0.79. The proposed algorithm has a face detection rate of 97.23% for the RFD1 database with a threshold value of 0.93. In order to determine the accuracy of the proposed face detection method, a comparison between the results of the proposed algorithm and those of the Viola-Jones algorithm is shown in table 2. As it can be seen, the accuracy of the Viola-Jones algorithm for the FERET and Markus Weber databases is better than that of the proposed algorithm. However, the accuracy of the proposed algorithm is greater than that of the Viola-Jones algorithm for

the RFD1 database. It should be noted that the Viola-Jones algorithm uses a 32-layer cascade, while the proposed method uses only one neural network. Furthermore, the Viola-Jones algorithm is trained with 1,000 face and 30,000 non-face samples, while the proposed method is trained with only 10 face and 30 non-face samples. Table 3 shows a comparison between the accuracies of the proposed method and three other methods. It should be noted here that in [5], by combining the Haar features, Gabor filters, and AdaBoost algorithms, an accuracy of 99.77% is obtained, while the proposed method uses only one Gabor filter to obtain an accuracy of 98.4%.

Algorithm 1: Face detection	
1:	<b>Procedure</b> Mask_gen (frequency f, angle T, aspect ratio G, standard deviation S, mask size n)
2:	<b>Input:</b> Gabor filter parameters value
3:	<b>Output:</b> real and imaginary forms of Gabor filter mask
4:	t ← - n
5:	<b>for</b> i ← 0 to 2*n + 1 <b>do</b>
6:	<b>for</b> j ← 0 to 2*n + 1 <b>do</b>
7:	X[i,j] ← t
8:	Y[j,i] ← t
9:	t = t + 1
10:	<b>end for</b>
11:	<b>end for</b>
12:	XP ← X * cos (T) + y * sin (T)
13:	YP ← -X * sin(T) + y * cos(T)
14:	K ← G / (2 * π * S^2)
15:	Real_mask ← K * exp(-
16:	(XP^2 + G^2 * YP^2) / 2 * S^2) * cos(2 * π * f * XP + 0)
17:	Imaginary_mask ← K * exp(-
18:	(XP^2 + G^2 * YP^2) / 2 * S^2) * cos(2 * π * f * XP + π/2)
19:	<b>return</b> Real_mask, Imaginary_mask
20:	<b>Procedure</b> Image_seg(image img)
21:	<b>Input:</b> an image
22:	<b>Output:</b> a list of image segmentations
23:	<b>for each</b> SS ∈ {60,50,40,35} <b>do</b>
24:	i ← 0
25:	j ← 0
26:	<b>while</b> ( i < length(img) <b>do</b>
27:	<b>while</b> ( j < width(img) <b>do</b>
28:	Img_seg_list ←
29:	Img_seg_list U img(i+ ss, j+ ss )
30:	j = j + 2
31:	<b>end while</b>
32:	i = i + 2
33:	<b>end while</b>
34:	<b>end for</b>
35:	<b>return</b> Img_seg_list
36:	<b>Procedure</b> Face_det(image img)
37:	<b>Input:</b> a colour image
38:	<b>Output:</b> detected faces
39:	<b>If</b> img ∈ the FERET dataset <b>then</b>
40:	NN_t = 0.86

```

38:         img ← resize(img, 192, 128)
39:     end if
40:     If img ∈ the Markus Weber dataset then
41:         NN_t = 0.93
42:         img ← resize(img, 224, 148)
43:     end if
44:     If img ∈ the RFD1 dataset then
45:         NN_t = 0.93
46:     end if
47:     img ← greyscale (img)
48:     [real_r, imag_r] ← convolve (Mask_gen (π/2,
π/2, 1, 0.4, 10) , img)
49:     Gabor_energy ← sqrt(real_r^2 + imag_r^2)
50:     Ge_bw ← blackandwhite(Gabor_energy)
51:     img_seg_list ← Image_seg(Ge_bw)
52:     face_detected ← false
53:     for each img_seg ∈ img_seg_list do
54:         if SR_NN(img_seg) > NN_t then
55:             face_detected ← true
56:             break for
57:         end if
58:     end for
59:     if face_detected = false then
60:         [real_r, imag_r] ← convolve ( Mask_gen
(π/2, π/2, 1, 0.7, 10) ,img)
61:         Gabor_energy ← sqrt(real_r^2 + imag_r^2)
62:         Ge_bw ← blackandwhite (Gabor_energy)
63:         img_seg_list ← Image_seg(Ge_bw)
64:         face_detected ← false
65:         for each img_seg ∈ img_seg_list do
66:             if SR_NN(img_seg) > NN_t then
67:                 face_detected ← true
68:                 break for
69:             end if
70:         end for
71:     end if
72:     if face_detected = true then
73:         return location coordinate of the face;
74:     else
75:         return no face detected
76:     end if
```

Figure 6. Face detection algorithm in the proposed method



Figure 7. The results of the proposed face detection algorithm on some image samples.



**Table 2. A comparison between the results of face detection using the proposed and Viola-Jones algorithms on 1484 images collected from FERET, Markus Weber and RFD1 databases.**

Database name	Algorithm	Threshold	DN	FAN	FRN	DR	TSWN	FAR
Markus Weber	Proposed	0.93	294	16	13	%95	3483*300=	1.5E-4
	Proposed	0.76	299	148	1	%99.66	1044900	1.4E-5
	Viola-jones	--	299	57	1	%99.6		
FERET	Proposed	0.86	246	3	4	%98.4	2145*250=	0.5E-7
	Proposed	0.79	249	15	1	%99.6	536250	2.7E-7
	Viola-jones	--	249	6	1	%99.6		
RFD1	Proposed	0.93	909	70	25	%97.23	2145*934=	3.4E-7
	Viola-jones	--	823	14	111	%88.11	2003430 --	--

**Table 3. Comparison between the results of the proposed and three other methods.**

	Proposed method	[12]	[27]	[29]
Face database	FERET	FERET	FERET	FERET
Success rate	%98.4	%99.77	%89.5	%95

### 5.3. The computational complexity of proposed face detection method

In order to determine the computational complexity of the proposed face detection method, the algorithm was carried out on a test image set from the FERET and Markus Weber databases using a computer with specifications as follow: Asus 1015PX mini laptop, Atom N70 CPU model with a 1.66 GHz CPU speed and 2 GB RAM. The sample images from the FERET and Markus Weber databases were segmented into 2,244 and 3,608 new images of size  $60 \times 60 = 3,600$  pixels. The time taken by the proposed face detection algorithm was 103 and 165 seconds for the image samples from the FERET and Markus Weber databases, respectively. Therefore, the time taken for processing any image of size  $60 \times 60$  pixels is 0.045 s. It should be noticed that the proposed algorithm is not a real-time approach, but if a new image segmentation method is selected so that the original image is converted to fewer than 22 image segments, the time taken for processing in the proposed method reduces to less than 1 seconds. This scheme will be addressed in the future work.

### 6. Conclusions

This paper presents a robust face detection algorithm using a combination of a 2D Gabor filter and a multilayer perceptron feedforward back-propagation neural network. The best values for the effective parameters of the Gabor filter banks were specified by analyzing 75 different Gabor filter

banks. It should be noted that the neural network used in this paper was trained with only 10 face and 30 non-face image samples. In addition, one filter (of 40 Gabor filters) was utilized.

The proposed face detection algorithm was tested on 1,484 collected images from three face databases, i.e. FERET, Markus Weber, and RFD1 (the image database created in this study). The images in the Markus Weber database had complex backgrounds. In addition, the images in the RFD1 database were captured in an unconstrained situation. The simulation results showed that the accuracy of the proposed face detection algorithm was 98.4% and 95% for the FERET and Markus Weber databases, respectively. The accuracy of the proposed algorithm for the RFD1 database was 97.23%, while the accuracy of the Viola-Jones algorithm for this database was 88.17%.

The proposed method segments an image into many new images of size  $60 \times 60$ , and therefore, it is not a real-time approach. A real-time version of the method will be considered in the future work.

### References

- [1] Sarkara, R., Bakshi, S. & Sa, P.K. (2012). A real-time model for multiple human face tracking from low-resolution surveillance videos. *Procedia Technology*, vol. 6, pp. 1004-1010.
- [2] Jones, M. J. & Viola, P. (2004). Robust real-time face detection. *International Journal of Computer Vision*, vol.57, no. 2, pp. 137-154.
- [3] Zhang T., et al. (2018). Fast and robust occluded face detection in ATM surveillance. *Pattern Recognition Letters*, vol. 107, pp. 33-40.
- [4] Tsai, Y. H., Lee, Y. C., Ding, J.J., Chang, R.Y. & Hsu, M.C. (2018). Robust in-plane and out-of-plane face detection algorithm using frontal face detector and symmetry extension. *Image and Vision Computing*, vol 78, pp. 26-41.

- [5] Ghazali K.H.B., et al. (2012) . An innovative face detection based on YCgCr color space. *Physics Procedia*, vol. 25, pp. 2116-2124.
- [6] Sun, X. , Wu, P. & Hoi, S.C.H. (2017). Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing*, vol. 299, pp. 42-50.
- [7] Zeng, D., Zhao, F., Ge, S. & Shen, W. (2019). Fast cascade face detection with pyramid network. *Pattern Recognition Letters*, vol 119, pp. 180-186.
- [8] Chen, J. et al. (2004). Novel face detection method based on Gabor features. In: Li, S.Z., Lai, J., Tan, T., Feng, G., Wang, Y. (Eds), *Advances in Biometric Person Authentication*. Springer, Berlin, pp. 90-99.
- [9] Huang, L. L., Shimizu, A. & Kobatake, H. (2005). Robust face detection using Gabor filter features. *Pattern Recognition Letters*, vol. 26, no. 11, pp. 1641-1649.
- [10] Kwolek, B. (2005). Face detection using convolutional neural networks and Gabor filters. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (Eds), *Artificial Neural Networks: Biological Inspirations*. Springer, Berlin, pp. 551-556.
- [11] Sahoolizadeh, H., et al. (2008). Face detection using Gabor wavelets and neural networks. *International Journal of Electrical and Computer Engineering*, vol. 2, pp. 1862-1864.
- [12] Xiaohua, L., et al. (2009). Face detection using simplified Gabor features and hierarchical regions in a cascade of classifiers. *Pattern Recognition Letters*, vol. 30 no. 8 , pp. 717-728.
- [13] FERET image database (2003), Available: <http://www.nist.gov/itl/iad/ig/colorferet.cfm>.
- [14] Markus Weber image database (1999), Available: [http://www.vision.caltech.edu/Image\\_Datasets/faces/faces.tar](http://www.vision.caltech.edu/Image_Datasets/faces/faces.tar).
- [15] OpenCv's Viola-Jones algorithm, Available: [https://docs.opencv.org/3.4.1/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html).
- [16] Oh, J., Choi, S. , Kim, C. , Cho, J. & Choi, C. (2013). Selective generation of Gabor features for fast face recognition on mobile devices. *Pattern Recognition Letters*, vol. 34, no. 13, pp.1540-1547.
- [17] Xu Y., et al. (2016). Image intelligent detection based on the gabor wavelet and the neural network. *Symmetry*, vol. 8, pp. 243-253.
- [18] Theodoridis, S. & Chellappa, R. (2014). *Academic press library in signal processing image and video compression and multimedia*. Academic Press,
- [19] Jones, J. P. & Palmer, L.A. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive rields in cat striate cortex. *Journal of Neurophysiology*, vol. 58 no. 6, pp. 1233-1258.
- [20] Kaushal, A. & Raina, J. P. (2010). Face detection using neural network & Gabor wavelet transform. *International Journal of Computer Science and Technology*, vol. 1, pp. 58-63.
- [21] Grigorescu, S. E., Petkov, N. & Kruizinga, P. (2002). Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing*, vol. 11 no. 10, pp. 1160-1167.
- [22] Roger, J. & Easton, L.(2010). *Fourier methods in imaging*. Rochester, John Wiley and Sons.
- [23] Amayeh, G., Tavakkoli, A. & Bebis, G. (2009). Accurate and efficient computation of Gabor features in real-time applications, in: Bebis, G. (Eds.) *Advances in Visual Computing*, Springer Heidelberg, Berlin, pp. 243-252.
- [24] Gu, W. , et al. (2012). Facial expression recognition using radial encoding of local Gaborfeatures and classifier synthesis. *Pattern Recognition* , vol. 45, no. 1, pp. 80-91.
- [25] Boulgouris, N. V., Plataniotis, K.N. & Tzanakou E.M. (2009). *Biometric: theory method and applications*. John Wiley & Sons.
- [26] Serrano, A., de Diego, I.M., Conde, C. & Cabello, E. (2011). Analysis of variance of Gabor filter banks parameters for optimal face recognition. *Pattern Recognition Letters*, vol. 32, no. 15, pp. 1998-2008.
- [27] Mavaddati, S. (2019). A novel face detection method based on over-complete incoherent dictionary learning. *Journal of AI and Data Mining*, vol. 7, no.2, pp.263-278.
- [28] Meshgini S. et al. (2013). Face recognition using Gabor-based direct linear discriminant analysis and support vector machine. *Computers and Electrical Engineering*, vol. 39, no. 3, pp. 727-745.