ARTICLE IN PRESS

Computers and Electrical Engineering xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A face detection method based on kernel probability map [☆]

Mohammad Reza Mahmoodi [*], Sayed Masoud Sayedi

Department of Electrical and Computer Engineering, Isfahan University of Technology, 8415683111 Isfahan, Iran

ARTICLE INFO

Article history:
Received 4 May 2014
Received in revised form 8 February 2015
Accepted 9 February 2015
Available online xxxx

Keywords:
Face detection
Skin detection
Face segmentation
Face extraction
Image processing
Kernel probability map

ABSTRACT

Face detection is one of the most important parts of biometrics and face analysis science. In this paper, a novel multi-stage face detection method is proposed which can remarkably detect faces in different images with different illumination conditions, variety of poses and disparate sizes. The idea is to utilize a preprocessing step to filter many non-face windows by means of a skin segmentation procedure in order to boost the speed of the detection and also utilize the color information as much as possible. Subsequently, candidate windows are fed to a Local Hierarchical Pattern (LHP) generator unit where a new texture pattern is produced. Based on this pattern, a kernel probability map is calculated for each window, and by summing probabilities of all kernels and comparing it with a predefined threshold, decision is made about content of the window. Not only does this algorithm effectively eliminate many non-face regions, but it is also capable of detecting faces with relatively acceptable rate in different conditions.

1. Introduction

Face detection is one of the most prominent algorithms in biometrics, and a considerable number of practical systems depend on face segmentation algorithms. Face and skin detection applications are myriad including human computer interaction (HCI) [1], face recognition, face tracking, facial expression recognition, fatigue detection, pedestrian and upper body detection [2,3], visual surveillance, gesture recognition [4], robotics, video and image indexing, multimedia and forensic applications. Just to mention in more detail, one specific example of these applications is the driving monitoring system. A glance at actuarial of sleepy people driving cars reported by National Sleep Foundation's 2005 Sleep in America poll shows that, 60% of adult drivers (about 168 million people) have admitted that they have driven a vehicle while feeling drowsy in the past year, and more than one-third (37% or 103 million people) have actually fallen asleep at the wheel [5]. As a matter of fact, more than 1 tenth of those who have nodded off, say they have done so at least once a month and four percentages of them avouched that they have had an accident or near accident since they were too tired to drive. The National Highway Traffic Safety Administration have estimated that 100,000 police-reported crashes are the direct result of driver fatigue each year resulted in an estimated 1550 deaths, 71,000 injuries, and $12.5 billion in monetary losses [5]. Fatigue directly affects the driver's reaction response time and drivers are often too tired to realize the level of inattention. Thus, a driving monitoring system shall be a promising solution to this serious problem. In such systems, several features including different signs and body gestures in the eyes or head such as repetitious yawning, heavy eyes and slow reaction will be exploited as a

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Ferat Sahin.

[*] Corresponding author. Tel.: +98 913 1666 402.
E-mail addresses: mr.mahmoodi@ec.iut.ac.ir (M.R. Mahmoodi), m_sayedi@cc.iut.ac.ir (S.M. Sayedi).

http://dx.doi.org/10.1016/j.compeleceng.2015.02.005
0045-7906/© 2015 Elsevier Ltd. All rights reserved.

drowsiness cue. An essential step before facial feature extraction in such systems is detection of human face. In such applications, both the accuracy and speed of the face detection algorithm is determinative.

Albeit face detection is utilized in many applications and systems, it is still an unsolved problem due to many difficulties and challenges involved in it. Shape of jaw, length of forehead, lips, eyes and other features are varied person to person that make it very difficult to propose a method only based on either the template or shape of the face. In addition, though the color of the face is almost identical for different people (i.e. in specific range of a color space), intensity (or brightness) varies significantly, and nonlinear and uneven illumination makes it worse. Besides various illumination and imaging conditions, disparate responses of camera devices makes it also very hard to use methods just based on skin and face color. Different face poses, occlusion of part of a face, different facial expressions and rotation of faces are other notable challenges [6]. The challenge is to design a system that work reliable in all conditions. Accordingly, recent trends in face detection methods show that the proposed algorithms are quite different from the ones proposed before. Early methods were often based on a single feature or task, while new methods combine different features, e.g., motion, color, shape, edge, and in higher levels, neural networks, Adaboost, SVM, etc., to overcome the challenges.

The proposed method in this work utilizes the color, texture and shape of the face to increase the robustness of the system. The novelty of this method is in its skin segmentation procedure and also in its texture based candidate verifier. The skin classifier is constructed based on a novel algorithm that can efficiently eliminate many non-face regions. It utilizes the idea of ternary image and performs several spatial processing steps to specify high probable skin regions. This algorithm is capable of eliminating 75 percentages of each image in average, while missing a negligible number of faces. After recognizing the potential face regions, a new texture based method is employed to recognize the faces inside these regions. The employed texture method is combined with the concept of kernel probability map which remarkably increases the robustness.

The paper is organized as follows. In Section 2, a brief discussion on face detection algorithms is presented. In Section 3, the proposed algorithm is presented and discussed in detail. Experimental results are provided in Section 4, and finally, conclusion is presented in Section 5.

2. Face detection algorithms

Numerous number of papers have been published in last two decades each proposing a particular method of detecting faces in images. These works are grouped into 4 categories; namely, knowledge-based methods, feature invariant methods, template matching methods, and appearance-based methods [1]. Knowledge-based methods are based on some predefined rules in geometry of the face. Though the simplicity might be an advantage for such algorithms, the robustness is highly questionable. Considering different shapes of the faces (in appearance and size), and difficulties associated with rotation in the head pose or wearing a glass or mustache, make these methods inefficient for use in a general system. Due to lack of robustness, new methods based on exclusive usage of rules in facial appearance have been rarely introduced in recent years. In feature invariant methods, one or several apparent features of the face such as edge, texture, color and even motion is used to localize the position of the face. Skin color based models due to their simplicity are found widely, however yet false detection is too high as skin models are either computationally inefficient or their false positive rates are too high. In [7], Gaussian color model in YC_bC_r color space is utilized to find faces. These models are not applicable in many practical systems as they are limited to their training faces. In [8], an embedded implementation of a skin detection method on FPGAs is proposed by means of YC_bC_r and $I_1I_2I_3$ color spaces. These colors are combined using an explicitly defined method of skin segmentation in order to separate skin pixels from non-skin ones. However, since no illumination cancellation algorithm is considered, the system is not robust against illumination and imaging conditions. In [9], a method based on the edge and skin tone information of the input color image is presented. After conducting skin segmentation in YC_bC_r and RGB space, the result is modified using a skin tone percentage index method, and then, the edge map of the image is combined with the skin tone image to separate all non-face regions from candidate faces. In [6,10,11], skin color models and their applications to face detection are probed and investigated in detail.

In template matching algorithms, faces are found by means of measuring correlation between several predefined or deformable templates and under-test windows. One of the best representatives of this method is the one presented in [12], in which Karen et al. proposed a method of detecting any arbitrary object by using sequential detectors. Detectors or classifiers are developed through a constrained optimization framework where the inner product of template with the under test image is minimized. Appearance-based methods include a large number of pre-test face images utilized to train a classifier; and then, in the test phase, input images are fed into the classifiers. VJ (Viola–Johns) face detector [13] and neural network based classifiers are two well-known methods in the last category. In [14], a segmentation-based illumination normalization method is proposed to compensate non-uniform illumination, and a Haar-classifier is followed for face detection. Neural network based systems are countless. In [15], a convolution based neural network is proposed with a quite good detection rate of 87% and low false positive of 4%. In [16], a face detector is presented which uses principle component analysis in order to extract a feature vector and develop a weak classifier. Subsequently, Adaboost algorithm is applied in order to boost the performance and construct the final strong classifiers. In [17,18], variety of appearance based methods, including different variations of VJ face detectors are elucidated in detail.

Please cite this article in press as: Mahmoodi MR, Sayedi SM. A face detection method based on kernel probability map. Comput Electr Eng (2015), http://dx.doi.org/10.1016/j.compeleceng.2015.02.005

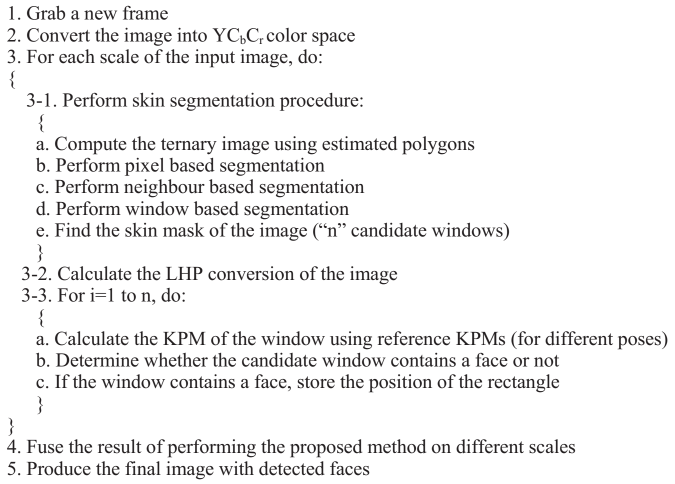
A group of relatively newer face detectors are based on combination of former methods. In [19], a face detection method based on both LUT skin segmentation and boosted cascade of classifiers is proposed. This system is evaluated in different data bases and it is reported that both speed and accuracy are improved compared to those of some former systems. A face detection system for surveillance applications is proposed in [20] based on a skin segmentation and Adaboost classifier. Simulation results in different lighting conditions showed that the system is not very effective in low light and high light conditions and it misses many faces. Wang et al. [21] introduced another system based on modified LBP texture, lip feature extraction and skin color segmentation. These features are combined to detect faces in an image. Even though this system seems to have a high quality with great speed, it is not still efficient as it does not consider the luminance component of YC_bC_r space that will degrade the performance of the system [6]. In [22], an improved version of Gaussian distribution based face detection is introduced. This system is a multi-layer system, consisting of a skin-color detection, labial feature extraction, hole feature extraction, binary image projection, Gaussian distribution and boundary adjusting. The important point here is that how much the estimated Gaussian function is close to the ideal one, and no test has been performed to verify the premise that distribution of skin color pixels in RGB color space is completely Gaussian. In addition, not any formal data base is used to evaluate the performance.

The proposed method in this work utilizes the color information of human skin and also many training faces in order to build a KPM (Kernel Probability Map). It indirectly uses the template of the face when it compares the under test window's KPM with the reference one. The LHP is a particular texture derived from LBP (Local binary Pattern) family of textures. This new feature is not a binary one, but is locally calculated similar to LBP.

3. The proposed method

3.1. An overview of the proposed method

In this section, a general overview of the proposed face detection system is presented and then each part of the system is elucidated in detail. The system utilizes a window-based scanning method which is used in many other image processing tasks too. In this method, the image is segmented into a set of windows in which each may or may not contain a face. For faces with sizes more than a typical window size, the algorithm is performed repeatedly on different scales of an image. Therefore, each face will be included at least in one window. This process is known as generating mosaic images in literature. The proposed method uses a skin segmentation algorithm which is capable of removing more than 70 percentages of non-face portions of an image (in average) while it can preserve more than 97 percentages of faces. Each window is processed, and the final result is a mask which specifies the ROI (Regions of Interest) to search for faces. This is done for all the scales. The resulting mask covers all candidate windows that probably contain faces. In the next stage, the LHP conversion of the image is calculated and then it is utilized in kernel probability map. Each window is processed in some parallel pose units. The structure for each pose unit is the same and the only difference is their kernel reference maps that will be explained later. By using kernel probability map in each kernel, the probability that a window contains a face is calculated and the result will be compared with a threshold value. The reference maps and threshold values are provided in training stage. The output of all pose units will be used in the decision unit to make the final decision. Different steps of the algorithm are summarized in Fig. 1 and block diagram of the design is depicted in Fig. 2.



```
1. Grab a new frame
2. Convert the image into $YC_bC_r$ color space
3. For each scale of the input image, do:
{
   3-1. Perform skin segmentation procedure:
     {
     a. Compute the ternary image using estimated polygons
     b. Perform pixel based segmentation
     c. Perform neighbour based segmentation
     d. Perform window based segmentation
     e. Find the skin mask of the image (“n” candidate windows)
     }
   3-2. Calculate the LHP conversion of the image
   3-3. For i=1 to n, do:
     {
     a. Calculate the KPM of the window using reference KPMs (for different poses)
     b. Determine whether the candidate window contains a face or not
     c. If the window contains a face, store the position of the rectangle
     }
}
4. Fuse the result of performing the proposed method on different scales
5. Produce the final image with detected faces
```

Fig. 1. Different steps of the proposed algorithm.

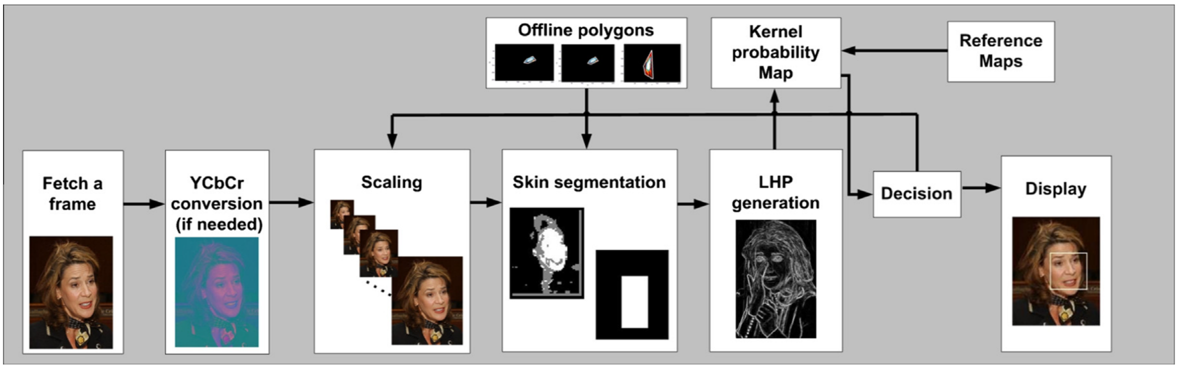


Fig. 2. The proposed algorithm.

In the following, the blocks shown in Fig. 2 are explained in detail.

3.2. Color space conversion

The optimum performance of a skin detection algorithm is not dependant on the chosen color space [23]. However, in this paper, the YC_bC_r color space is used for some reasons. YC_bC_r is similar to the human visual perception process and it is a discrete color space which makes it easy to realize clustering algorithms. One advantage of YC_bC_r color space compared to some other color spaces like HSV and TSL is that it can be obtained easily from RGB data by using a linear transformation. Comparing to RGB, the redundancy of YC_bC_r channels is much lower and its components are highly independent. YC_bC_r is widely used in European television studios, JPEG, MPEG and CCD cameras, so it can be used directly in many applications without transforming the image into another color space. Also, in [24], it has been shown statistically that the performance of YC_bC_r-based methods compared with other color spaces is often superior.

3.3. Scaling

The act of making mosaic images [1] (different scales of one image) is carried out in order to include faces with sizes larger than default window size. The scaling factor is chosen 1.1 since it has been experimentally yielded to acceptable results. The size of an input image is optional, but window's size is set to 48 $*$ 48 pixels. This size is enough for images in the range of 640 $*$ 480, but the algorithm may miss very small faces for lower dimension images. For any given image with an arbitrary size, down sampling continues until the size of the image will be less than window's size. The overall process of face extraction is undertaken for all scales (similar to every other face detection method) and the final result is based on fusion of all outputs.

3.4. Skin segmentation: offline operations

The skin detection process contains a training stage that is performed offline. Necessary statistical data obtained in this stage are preserved for further usages. It is used to partition the color space into 3 sets of pixels. These sets will be used in the

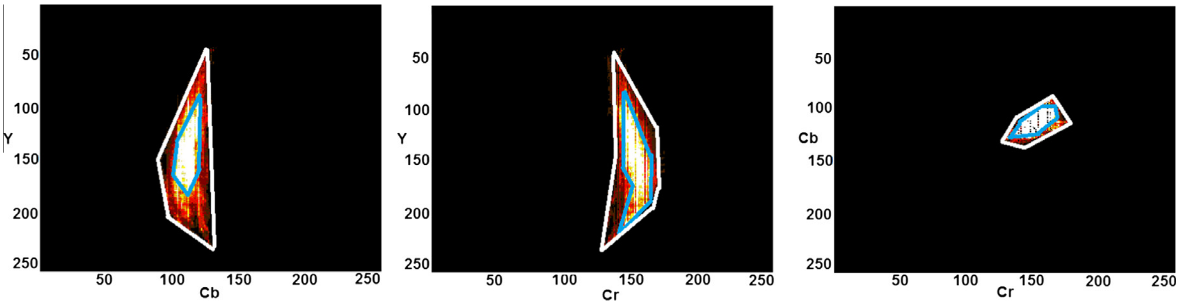


Fig. 3. Intensity heat maps of YC_bC_r color space channels. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

pixel based stage to build the ternary image. For each input image, each pixel is processed in a pixel-based, neighbor-based and a window-based process and the final output will be elected windows.

In order to train the system, about 63 million skin pixels have been collected from World Wide Web. The intensity heat maps of these pixels are depicted in Fig. 3. Each map is divided into 3 non-overlapping partitions. The boundary of each partition is estimated by a polygon in order to keep up both the simplicity and accuracy of the algorithm. For each map, 2 polygons are estimated. Pixels which are observed with high frequency are surrounded by the inner polygon. The outer polygon separates pixels which are not necessarily highly repeated, but, they can still be part of the skin in some conditions, and non-observed pixels. Thus, these two boundaries separate pixels into the pixels observed with low frequency, observed with high frequency, and pixels which are not observed at all. This has to be carried out for each map.

Accordingly, three sets of pixels; T_1, T_2, and T_3 are defined. T_1 contains pixels which are simultaneously located inside inner polygons of all three maps. T_2 contains pixels which are inside the outer polygons and are outside the inner polygons of three maps. And the last one T_3 is filled with pixels which are not located inside the outer polygons of all three maps. As it will be shown later, by this partitioning, dealing with nonlinear illuminations will be much easier. The data needed are coefficients of polygon lines which are extracted in training stage and they are kept in the memory of the processor.

3.5. Skin segmentation: pixel-based segmentation

Segmentation of an image into a ternary one is the first step of the algorithm. Here, contrary to most existing methods, the image is not roughly classified into two groups of skin and non-skin pixels; and the reason is that there are lots of uncertain pixels that cannot easily be put in either of two groups. In the proposed ternary segmentation, if for some reason, like non-uniform illumination, decision on some pixels is difficult, instead of randomly discard or accept them, they are kept aside to be processed in next steps based on their neighbors and region information (spatial based segmentation).

In pixel-based segmentation, each pixel of the input image is classified in accordance with the 3 predefined sets. T_1 pixels are set white (255), T_3 pixels are set black (0) and T_2 pixels are set gray (128). Fig. 4 shows an input image selected from LFW data base and its ternary conversion. The figure shows that the polygon estimation has correctly segmented the image and the idea of ternary image is reasonable. Using simple methods such as explicitly defined regions or even Bayesian classifier (LUT) will not lead to accurate results. Using some of these methods has been yielded to false alarm rate in simulation. In Fig. 4, those skin regions that are not highly affected by nonlinear illumination are detected correctly and these pixels more or less will be specified by most of the methods. These pixels are in T_1 set and shown by white points in the figure. The gray pixels (in set T_2) are those for which a strict decision cannot be made. Some of them are truly skin pixels, while some are not; it is necessary to consider a method for correct classification.

3.6. Neighbor-based Segmentation

It has been observed that there are many skin pixels surrounded by a considerable number of white pixels; but, they have not been set white yet. This may have been due to illumination, a shadow, moustache, etc. The neighbor based procedure compensates these effects to some extent. In addition, isolated pixels mainly caused by noise will be eliminated too.

Each pixel of the ternary image is processed in the neighbor based algorithm. A 3 × 3 and a 5 × 5 neighborhood are considered for all of the pixels except skin pixels which have been verified before (the T_1 pixels) and bordering pixels. For each under test pixel (UTP), based on the number of T_1, T_2 and T_3 type pixels in its neighborhood, and by using a set of rules, a score is calculated. To that end, the numbers of surrounding white, gray and black points are obtained, and subsequently, considering both the simplicity of the algorithm (it should be as fast as possible) and the fact that false rejection rate shall be as low as possible (with the cost of higher false detection), a semi look-up-table procedure is used to calculate the score. It is done for both the 3 × 3 and 5 × 5 neighborhoods. The final score (for each UTP) is computed as:

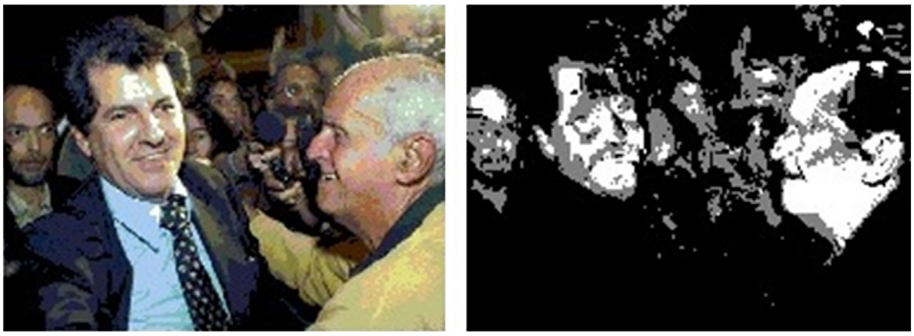


Fig. 4. An input image from LFW and its ternary conversion.

$$\zeta = K \times T + \Phi \tag{1}$$

where ζ is the total score, K is a simple constant to give more weight to nearer pixels, and T and Φ are, respectively, the computed scores (as explained above) for the 3 × 3 and 5 × 5 neighborhoods of the UTP. Using two experimentally chosen threshold values Thr_1 and Thr_2, a hysteresis thresholding is applied to increase the robustness of the algorithm. Based on the value of ζ and Thr_1 and Thr_2 ($\text{Thr}_1 < \text{Thr}_2$), a ternary decision is made on UTP. If ζ is lower than Thr_1, then UTP will be considered black, if it is higher than Thr_2, the pixel will be considered white, and if it is between Thr_1 and Thr_2 then the pixel will be considered gray again. The values of Thr_1 and Thr_2 are statistically determined by applying different values of thresholds and observing the effectiveness of the algorithm in such a way that high false detection is acceptable while false rejection is minimized. Even though it is possible that the algorithm specifies some non-face regions as a face candidate, the follow up texture based segmentation procedure will eliminate most of them. In fact, the output of this stage is another ternary image that for some of its pixels still no decision has been made.

3.7. Window-based segmentation

Without deteriorating the generality of the algorithm, the size of output windows is considered to be 48 × 48. In window based segmentation, each window is determined to be a candidate or not. Decision is made based on the number of white and gray points in the window. If there are no white points, then it is reasonable to bypass it and if it contains many white points, then there is a high probability that the window contains whole or at least a part of human face. Thus, a set of rules are utilized to filter windows. Thresholds could be adaptively determined by the system; however, to boost the speed of the algorithm, they are determined empirically. An important factor considered in setting thresholds and rules in skin segmentation part is minimizing false rejection rate as much as possible. The false positives will be filtered out by subsequent stages. To improve the results, after window processing, a morphological operation is performed to add missing parts of a face that are probably occluded by other objects like glasses or even eyes since they have different colors from skin color. Therefore, windows surrounded by candidate windows will be annexed to previously chosen ones (hole filling). Fig. 5 represents two images selected from LFW data base, and the result of applying the algorithm on them.

3.8. LHP conversion

The LHP conversion unit is designed to convert the luminance of the image into LHP (Local Hierarchical Pattern). As it is observed in Fig. 6, using luminance of an image to build kernel probability map will not produce satisfactory results, and faces and non-faces are not separated at all. However, texture is a great tool that can effectively discriminate between face and non-face windows when combined with the idea of kernel probability map. Fig. 7 represents the probability of a window based on LHP discriminator. In each of the charts, the probability sum in specific window of an image, for identical face and non-face number of images, is represented. Both Figs. 6 and 7 show the advantage of LHP over the luminance, as when using LHP, the distance between probability of faces and non-faces are noticeable and it logically helps better separation of face and non-face windows.

Another important feature of LHP is its capability to cancel illumination effect to some extent. Illumination in nature has low frequency components and it is almost stable all over an image. The fact that LHP is a local operator and it is based on differences makes the illumination cancellation better. Fig. 8 graphically compares luminance of 5 faces with their LHP representations. It can be observed that LHP contains huge section of face information and it is almost unaffected in different illumination conditions. Thus, it can be expected that if in color segmentation part, the algorithm could not remove parts of the image due to uneven and nonlinear illuminations, by this local transformation, it would be possible to counteract this effect and detect faces even in different lightings conditions.

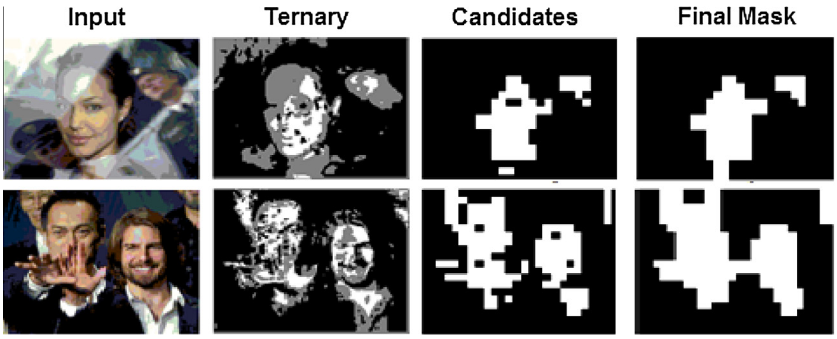


Fig. 5. Example of two challenging LFW images.

Please cite this article in press as: Mahmoodi MR, Sayedi SM. A face detection method based on kernel probability map. Comput Electr Eng (2015), http://dx.doi.org/10.1016/j.compeleceng.2015.02.005

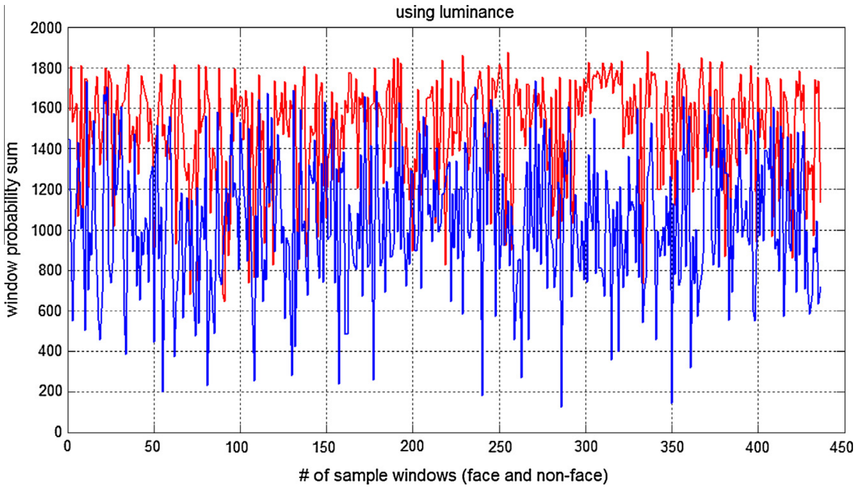


Fig. 6. Using luminance to discriminate face and non-face.

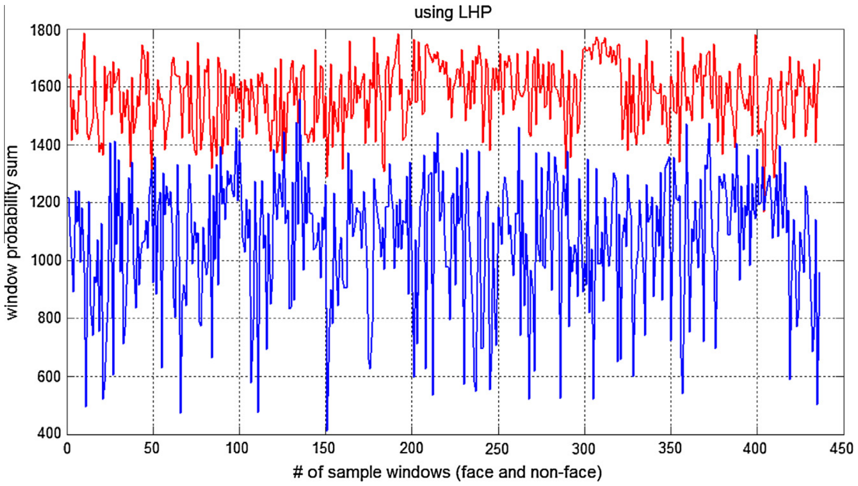


Fig. 7. Using LHP to discriminate face and non-face.

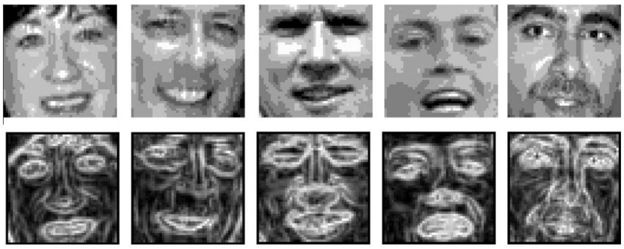


Fig. 8. The luminance and LHP of several faces (upper row: luminance, lower row: LHP).

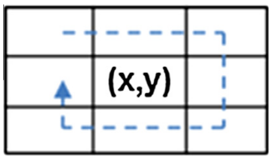


Fig. 9. Calculating LHP for a center pixel.

Please cite this article in press as: Mahmoodi MR, Sayedi SM. A face detection method based on kernel probability map. Comput Electr Eng (2015), http://dx.doi.org/10.1016/j.compeleceng.2015.02.005

Similar to LBP textures, LHP of each pixel depends on the values of its surrounding pixels. For the center pixel, shown in Fig. 9, LHP is calculated as:

$$\text{LHP}\ (x,y) = \sum_{k=1}^{9} \pi_k \tag{2}$$

where π_k $(1 \leqslant k \leqslant 8)$ for each of the neighbors is calculated by:

$$\pi_k = \Psi\ (Y_k - A) \tag{3}$$

in which Y_k is the luminance of neighbor k, A is median of pixels' luminance in the 3 × 3 neighborhood, and Ψ is a roof function with nonlinear coefficients. For each pixel, Eq. (2) is calculated. These coefficients are empirically considered in such a way that illumination will be eliminated as much as possible and maximum separation is obtained when the texture is combined with kernel probability map. The gray level of the LHP image is directly related to simple and small nano-patterns (edges, points, etc.) of the input image. The LHP image will be utilized in order to build the kernel probability map.

3.9. Kernel probability map

KPM (Kernel Probability Map) is a concept used here to measure the probability that a window holds a face. KPM is a matrix with certain number of rows and columns. Rows are related to the number of kernels in a window and columns are related to the number of bins (for each kernel). Each pixel is related to one kernel. In general, a kernel can be considered as the average of an $n \times n$ neighborhood, but in this paper, a one to one relation is utilized i.e., for a 48 × 48 window there would be 2304 possible kernels. In order to build the reference KPM, more than 400 48 × 48 face windows were utilized. For a fixed pose, it is reasonable to suppose that in each kernel of a window, the intensity of the LHP image has a certain value and it can be probabilistically calculated. In other words, for each kernel, by using 400 face window images, a probability function is estimated. Fig. 10 illustrates examples of probability functions (PF) for kernels number 100 and 200. In the figure, Y-axis is the normalized probability and X-axis represents the number of bins.

Bins are used because it was not necessary to consider all 256 possible intensity levels. 256 different gray levels are mapped into 27 bins or classes (non-linearly). PFs are made based on abundance of values of different classes in all 400 images. Then, each PF is normalized to its maximum value. This provides the reference map matrix (Eq. (4)) which is implementable in software or hardware platforms such as FPGAs easily.

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,b} \\ p_{2,1} & p_{2,2} & p_{2,3} & \cdots & p_{2,b} \\ \cdots & & & & \cdots \\ \cdots & & & & \cdots \\ p_{n,1} & p_{n,2} & p_{2,3} & \cdots & p_{n,b} \end{bmatrix} \tag{4}$$

In Eq. (4), matrix P is the output of the training stage for one specific pose. Parameters b and n are the number of bins and number of kernels respectively. After fetching one window from the LHP unit, the intensity of each kernel will be calculated and in accordance with that, the probability of that kernel will be specified by using matrix P. In other words, the value of kernel nth will determine one specific element of the nth kernel (row) of P which gives one number between 0 and 1. It is considered as the probability of kernel nth to be part of a face. Simply, all probability values of a window will be summed up and the final number will specify the possibility that the window contains a face.

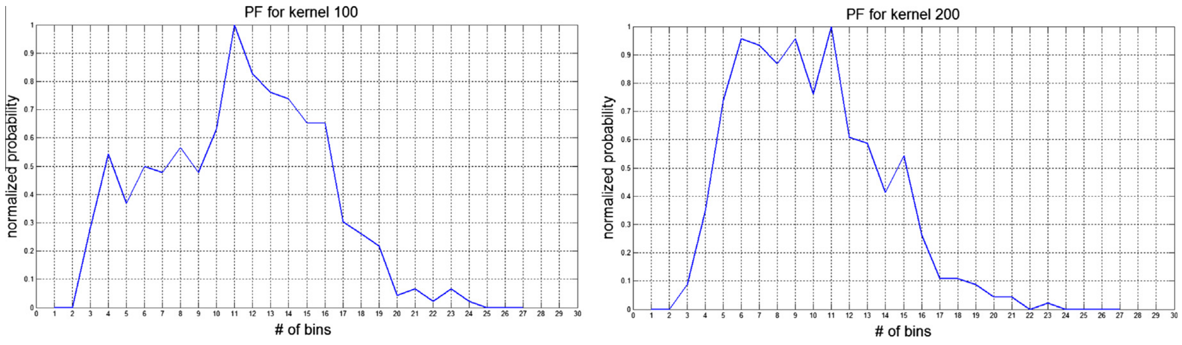


Fig. 10. PF for kernel number 100 and 200.

Not only is the method computationally cost effective and simply implementable, but also as simulation results showed, it is accurate enough to give acceptable detection rate and false alarm rate compared with other methods. It is noteworthy to point out that the above idea was also tested for differential kernels (building KPM based on difference of neighbor kernels instead of their intensity themselves), but the output was not interesting and the windows containing faces were not accurately separated from the non-faces windows. Combination of LHP concept and KPM definition leads to a noticeable performance as it was depicted in Figs. 6 and 7; where the luminance itself was not strong enough to discriminate between face and non-face windows. In other words, variation of Y values is too high to build an efficient KPM based on that. LHP extracts most prominent patterns in a window. In the proposed multi-layer face detection method, both color (skin segmentation) and shape of the face (LHP + KPM matrix) are utilized to detect faces in an image.

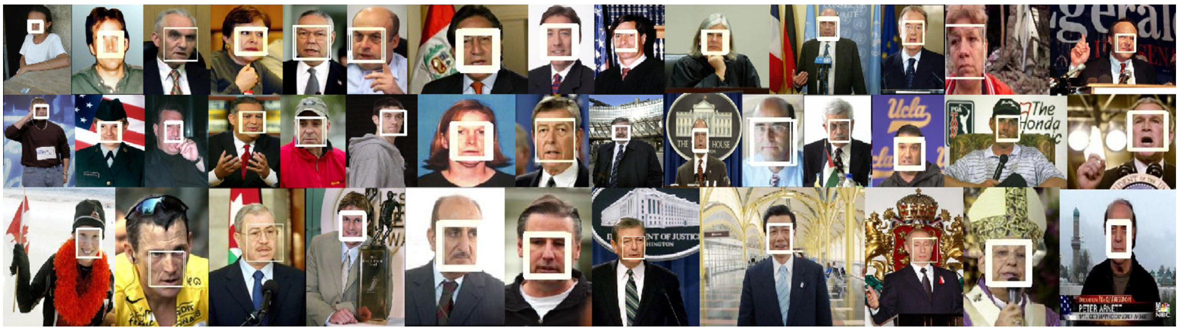


Fig. 11. Images with different lighting conditions.

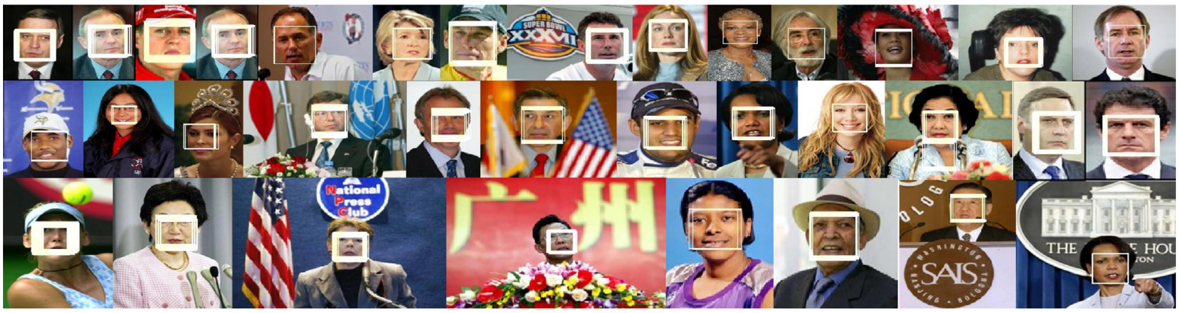


Fig. 12. Images captured from people with various ethnicities and races.

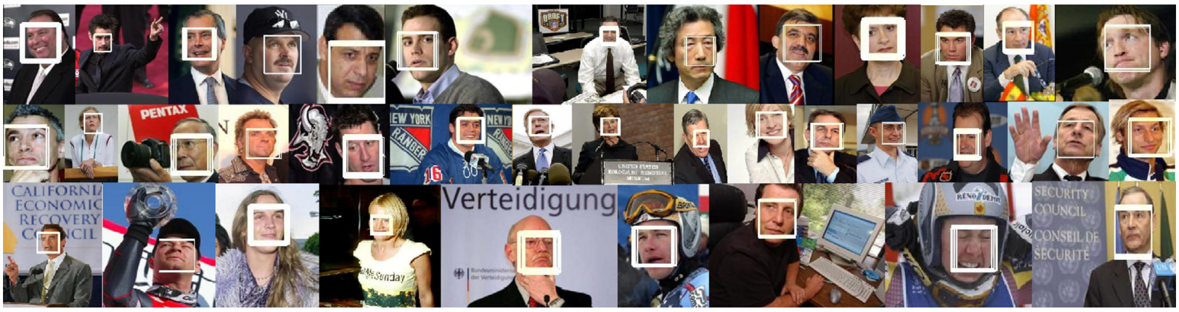


Fig. 13. Images having faces with different pose and orientatations.

4. Results and discussion

4.1. Qualitative results

The proposed algorithm has been developed via MATLAB environment. Qualitative results are shown in Figs. 11–14. In the figures, randomly selected images from LFW database are utilized to demonstrate the effectiveness of the method. In Fig. 11, the result of performing the proposed method on images captured in different lighting and illumination conditions is provided. As it is observed, the proposed method is capable of classifying face regions in color images in most of imaging conditions. Fig. 12 shows images associated with people having various races, ethnicities and nationalities. As it is also observed, the proposed method can handle different images from different groups of people around the world. Fig. 13 provides images useful for assessing the effectiveness of the method in dealing with different poses and orientations. Multifarious images are presented each distinguished with a particular pose or orientation. Faces rotated up to 30° are detected with no critical difficulties. In Fig. 14, the result of applying the method on different facial statuses which could be challenging in many conditions is presented. The faces have been detected with remarkable rate. The results of Figs. 11–14 show the algorithm is capable of processing arbitrary multifarious images with acceptable rate.

4.2. Quantitative results

As mentioned before the proposed algorithm has been developed via MATLAB environment. Images were tested and the number of truly detected windows (associated with faces) and the numbers of incorrectly detected or missed faces have been calculated. In order to measure the performance of the proposed method, 4 standard data bases have been used to test the system. This helps to fairly compare the performance of proposed method with other available methods. Most face detection papers report their results in terms of FP, FN, TP, recall and precision, which are already defined in the literature for assessing any classification problem.

Generally, present algorithm is capable of processing approximately 10 frames (640 × 480) per second in a system with dual 2.17 GHz processor and 3 GB RAM where a 32-bit window 7 is the operating system. Table 1 shows the performance of the proposed system under LFW, Caltech, Bao, and GTDB data bases. As it is shown in Table 1, the proposed algorithm finds faces with a considerable detection rate. Caltech contains approximately 450 images with complex background and varied illumination. The robustness of the system against illumination and imaging conditions was verified through processing of these images. It is observed that only 11 faces are missed which were mostly occluded or very dark. For Bao data base, 210 images were tested. 149 of them were single face and the rest contained multiple faces. In this database, it was observed that most misses were related to faces which have highly rotated. GTDB with 750 faces is a database with images similar to that of Caltech. Best performance is obtained in GTDB. Table 2 provides comparison information between the proposed method and some other recently developed algorithms. As it is observed, in overall, the algorithm is capable of finding faces, with a remarkable rate, and false alarm rate is also acceptable.

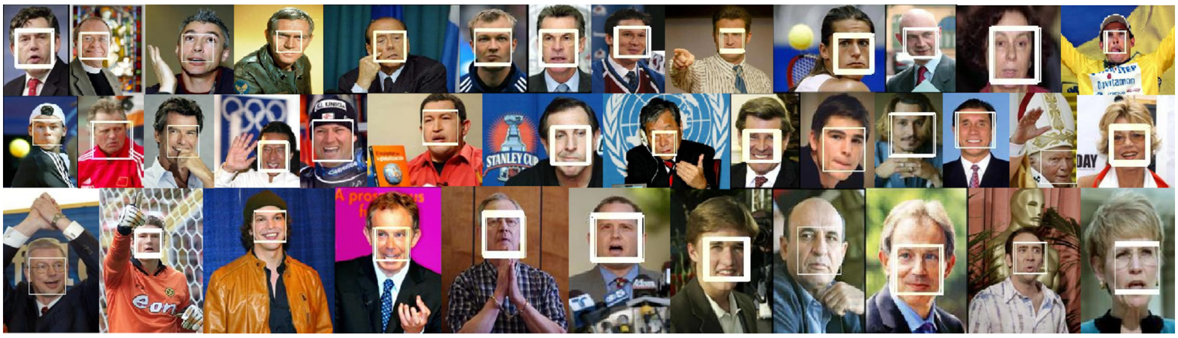


Fig. 14. Images including people with different facial expressions.

Table 1
Perfromance of the proposed alogirthm.

Subset	TP	FN	FP	Recall (%)	Precision (%)
LFW	410	38	45	91	90
Caltech	439	11	43	97	91
Bao	303	23	36	93	89.4
GTDB	739	11	69	98	91.4

Table 2
Comparison table.

Method	PCA + Adaboost [16]	Color segmentation [20]	Adaboost and skin model [25]	[15]	Viola–Johns [13]	Viola–Johns - EER [13]	SVM [16]	Skin Pre-filter [26]	Skin Color [27]	ANN [16]	[28]
Precision	91.7	90.2	92.6	87	94.65	89.64	87.5	88.48	87.6	82.6	90.26
Recall	–	–	–	99	77.43	90.27	–	90.8	90	–	87.1

5. Conclusion

In this paper, a new face detection method which can be utilized in variety of applications is proposed. The method is performed in separate phases. In one phase, a skin segmentation algorithm based on color information is performed in which regions of interest with high probability of existing faces are selected. In this part, a new idea of ternary image is proposed. It is different from the, rough image binary conversion in LUT, explicitly defined, Gaussian based, or even adaptive skin segmentation algorithms. After segmentation, in the second phase, candidate windows are fed to the LHP generator to produce a new pattern of image. This helps building kernel probability map of each window. Finally, the probability that each window contains a face is calculated. If it is more than a threshold value, the window is reported to have a face. This system is robust against different illumination conditions and various poses, which is shown both qualitatively and quantitatively through the results. At software level, the method is limited from the speed point of view. FPGA implementation of the algorithm is underway to make the system work in real-time, where all pixel based operations can be done in parallel in each step of the algorithm. Also the idea of using spatial analysis in skin classification is considered, which can hugely increase the accuracy of the system.

References

[1] Yang M, Kriegman D, Ahuja N. Detecting faces in images: a survey. IEEE Trans Pattern Anal Mach Intell 2002;24(1):34–58.

[2] Barbu T. Pedestrian detection and tracking using temporal differencing and HOG features. Comput Electr Eng 2014;40(4):1072–9.

[3] Duffner S, Odobez J. Leveraging colour segmentation for upper-body detection. Pattern Recogn 2014;47(6):2222–30.

[4] Chaudhary A, Jagdish LR, Kunal S, Raheja S. An ANN based approach to calculate robotic fingers positions. In: Advances in computing and communications. Berlin Heidelberg: Springer; 2011. p. 488–96.

[5] Mahmoodi MR, Sayedi SM. Leveraging spatial analysis on homogonous regions of color images for skin classification. In: 2014 IEEE 4th international e-conference on computer and knowledge engineering (ICCKE); 2014. p. 209–14.

[6] Kakumanu P, Makrogiannis S, Bourbakis N. A survey of skin-color modeling and detection methods. Pattern Recogn 2007;40(3):1106–22.

[7] Ji Z, Qin H. A vehicle surveillance system for face detection. In: IEEE international conference on vehicular electronics and safety (ICVES); 2007. p. 1–4.

[8] Liu G, Zhongke S. Embedded implementation of real-time skin detection system. In: International conference on transportation, mechanical and electrical engineering (TMEE); 2011. p. 2463–6.

[9] Ghimire D, Lee J. A robust face detection method based on skin color and edges. JIPS 2013;9(1):141–56.

[10] Mahmoodi MR, Sayedi SM. Boosting performance of face detection by using an efficient skin segmentation algorithm. In: 2014 IEEE 6th International conference on information technology and electrical engineering (ICITEE); 2014. p. 1–6.

[11] Mahmoodi MR, Sayedi SM. A face detector based on color and texture. In: 2014 IEEE 6th international conference on information technology and electrical engineering (ICITEE); 2014. p. 1–6.

[12] Keren D, Osadchy M, Gotsman C. Antifaces: a novel, fast method for image detection. IEEE Trans Pattern Anal Mach Intelligence 2001;23(7):747–61.

[13] Viola P, Jones M. Robust real-time face detection. Int J Comput Vision 2004;57(2):137–54.

[14] Yao M, Aoki K, Nagahashi H. Segmentation-based illumination normalization for face detection. In: IEEE 6th international workshop on computational intelligence & applications (IWCIA); 2013. p. 95–100.

[15] Farrugia N, Mamalet F, Roux S, Yang F, Paindavoine M. Fast and robust face detection on a parallel optimized architecture implemented on FPGA. IEEE Trans Circuits Syst Video Technol 2009;19(4):597–602.

[16] Shuang L. A novel face detection algorithm based on PCA and Adaboost. In: 2014 IEEE 6th international conference on measuring technology and mechatronics automation (ICMTMA); 2014. p. 38–41.

[17] Hjelmås E, Low BK. Face detection: a survey. Comput Vis Image Underst 2001;83(3):236–74.

[18] Zhang C, Zhang Z. A survey of recent advances in face detection. Tech Rep. Microsoft Research; 2010.

[19] Ban Y, Kim A, Kim S, Toh K, Lee S. Face detection based on skin color likelihood. Pattern Recogn 2014;47(4):1573–85.

[20] Hu K, Pai Y, Ruan S, Naroska E. A hardware-efficient color segmentation algorithm for face detection. In: IEEE Asia Pacific conference on circuits and systems (APCCAS); 2010. p. 688–91.

[21] Wang N, Chang S, Chou P. A real-time multi-face detection system implemented on FPGA. In: International symposium on intelligent signal processing and communications systems (ISPACS); 2012. p. 333–7.

[22] Liu Z, Sha J, Yang P. Multi-face detection based on improved gaussian distribution. In: 5th International conference on intelligent human-machine systems and cybernetics (IHMSC); 2013. p. 54–7.

[23] Albiol A, Torres L, Delp E. Optimum color spaces for skin detection. ICIP 2001:122–4.

[24] Shin M, Chang K, Tsap L. Does color space transformation make any difference on skin detection. In: 6th IEEE workshop on applications of computer vision (WACV 2002); 2002. p. 275–9.

[25] Cui-huan D, Hong Z, Li-ming L, Jie L, Xiang-yang H. Face detection in video based on AdaBoost algorithm and skin model. J China Univ Posts Telecommun 2013;20:6–24.

[26] Tabatabaie Z, Sadri R, Udzir N, Kheirkhah E. A hybrid face detection system using combination of appearance-based and feature-based methodism. Int J Comput Sci Network Security 2009;9(5):181–5.

[27] Wu Y, Ai X. Face detection in color images using AdaBoost algorithm based on skin color information. In: 1st International workshop on knowledge discovery and data mining (WKDD); 2008. p. 339–42.

[28] Phung S, Bouzerdoum A, Chai D. A novel skin color model in YC_bC_r color space and its application to human face detection. In: IEEE international conference on image processing; 2002. p. I–289.

[29] Anghelescu P, Serbanescu I, Ionita S. Surveillance system using IP camera and face-detection algorithm. In: International conference on electronics, computers and artificial intelligence (ECAI); 2013. p. 1–6.

[30] Ooi M. Hardware implementation for face detection on Xilinx Virtex-II FPGA using the reversible component transformation colour space. In: 3rd IEEE international workshop on electronic design, test and applications; 2006. p. 6.

[31] Kalal Z, Matas J, Mikolajczyk K. Weighted sampling for large scale boosting. In: BMVC; 2008.

[32] Cevikalp H, Triggs B. Efficient object detection using cascades of nearest convex model classifiers. In: CVPR; 2012.

Mohammad Reza Mahmoodi received his B.Sc. and M.Sc. degree in Electrical Engineering from IUT (Isfahan University of Technology) in 2012 and 2014 respectively. His research interests include digital/analog circuit design, image processing, and computer vision.

Sayed Masoud Sayedi received his Ph.D. degree in Electronics from Concordia University 1996. From 1988 to 1992, and then since 1997, he has been with IUT, where he is currently an associate professor in the Department of Electrical and Computer Engineering. His areas of interest include digital electronics, VLSI circuits, and data converters.

Please cite this article in press as: Mahmoodi MR, Sayedi SM. A face detection method based on kernel probability map. Comput Electr Eng (2015), http://dx.doi.org/10.1016/j.compeleceng.2015.02.005