Efficient algorithms for detection of face, eye and eye state

Hashem Kalbkhani¹, Mahrokh G. Shayesteh¹,², Seyyed Mohsen Mousavi³

¹Department of Electrical Engineering, Urmia University, Urmia, Iran
²Wireless Research Laboratory, ACRI, Electrical Engineering Department, Sharif University of Technology, Tehran, Iran
³Department of Computer Engineering, Urmia University, Urmia, Iran
E-mail: m.shayesteh@urmia.ac.ir

Abstract: Eye state analysis (open or closed) is an important step in fatigue detection. In this study, an efficient algorithm for eye state detection is proposed. At first, a new face detection method is presented for noisy images that finds the face area in the input image well. Then, novel algorithms for detection of eye region and eye state are introduced. The performance of the proposed method is evaluated on four different databases namely FERET, Aberdeen, IMM and CVL which contain more than 5700 images with different descents, positions, light conditions and glasses. The results show that the new method achieves more accuracy rate than the previously presented algorithms, while it does not need training data and is also computationally efficient.

1 Introduction

Driving with drowsiness is one of the main causes of traffic accidents. A great number of fatalities occurring in traffic accidents could be avoided if driver drowsiness is detected and alarm signals are provided to the driver.

There are several methods for drowsiness detection. They can be divided into three types [1]: biological indicators, vehicle behaviour and face analysis. The first type measures biological indicators such as brain waves and heart rate. These techniques have the best detection accuracy, but they require physical contact with the driver. They are intrusive, so they cannot be used in practice. The second type measures vehicle behaviours such as speed, lateral position and turning angle. These methods can be implemented non-intrusively, but they have several limitations such as the vehicle type, driver experience and driving conditions. Furthermore, they need special equipments and can be expensive. The third type is face analysis. Although it can be less accurate than the indicators, this type is non-intrusive and easily implemented. It can be employed independent of driver experience and vehicle type. Further, it is both more practical and accurate than the vehicle behaviour analysis. However, it is limited by lighting conditions and the driver’s distance from the camera.

In this paper, we aim to determine eye state (open or closed), which is a key step in driver fatigue detection. For this purpose, we first need to detect face region, then we find eyes and after that, we detect eye state. In the following, a brief review on the previous algorithms for detection of eye regions and eye state is presented.

Many eye detection methods have been exploited during the near two decades. They can be grossly divided into two categories: image-based passive approaches and infrared (IR)-based active ones. The image-based is partitioned into three major approaches: template-based [2, 3], feature-based [4, 5] and appearance-based [6, 7]. In the template-based methods, at first, a generic eye model based on the eye shape is designed. Then, template matching is used to search the image for eyes. This method searches whole of image for eye regions, hence it is time consuming. The feature-based approaches explore the characteristics of eyes to identify some distinctive features around the eyes. However, eyebrow and face orientation may also degrade the performance of the discriminative function, since its performance depends on the accuracy of candidate eye window detection. The appearance-based methods detect eyes based on their photometric appearance. These methods usually need to collect a large amount of training data, representing the eyes of different subjects, under different face orientations and illumination conditions. These data are used to train a classifier such as a neural network or the support vector machine. In summary, the above three mentioned methods detect the eyes by exploiting eyes differences in appearance and geometric structure from the rest of the face. The special features of eye such as dark pupil, white sclera, circular iris, eye corners, eye shape etc. are utilised to distinguish the human eye from other objects. However, because of changing of lighting conditions and face pose, these differences will be too trivial to distinguish. Particularly, illumination variations in eye detection applications could be greatly sensitive. In some applications, Hough transform is used to detect eye regions in face images [8]. Hough transform needs high computations, so it is time consuming; consequently it is not suitable for real time applications. IR-based methods [9, 10] are only restricted to some specific applications since they need the assistance of IR illuminating devices.
former is relatively application independent and more widely used with merit that no extra equipment is needed.

For eye state detection, the method of [11] uses the distance between two eyelids. For a person being able to see, the upper eyelid should not cover the pupil. Thus, if the distance between the two eyelids is less than the iris radius, the eye is closed. However, this method is very sensitive to pupil centre location. In [1], eye property in saturation channel of HSI colour space is utilised for eye state detection. This property is used to extract iris region from skin region in iris circle. After choosing threshold value and creating binary image of iris circle, eye state is set to open if the number of white pixels in iris circle is more than the black pixels.

In this study, for face detection, we assume that there is one face region in the input image. We create a rectangle with pre-defined length and width, because the driver has specific distance from the camera. After finding face region, we obtain eyes regions. We use EyeMap algorithm introduced in [12] for this purpose. Although this algorithm has very good performance for detection of eye regions in different light conditions, but its disadvantage is that the other dark points are appeared in the detection. Hence, we propose a new method to remove them. Then, we find pupil centre and circle of iris. For detection of eye state, we separate iris pixels from skin pixels in the detected eye area. The authors in [13] used three-dimensional distance in RGB colour space to extract iris pixels from skin pixels. In RGB colour space, a little variation in lighting conditions causes large changes in RGB components. However, in YC_bC_r colour space, the luminance component is separated from the chrominance components. So, we propose to use YC_bC_r colour space for eye state detection. We examine the proposed algorithm on four different databases including FERET [14], Aberdeen [15], IMM [16] and CVL [17] databases, and compare its accuracy rate with other methods. The results indicate performance improvement of the new method over the existing algorithms.

The rest of this paper is organised as follows: Section 2 describes the proposed face detection algorithm. In Section 3, we explain eye regions detection method and our proposed rules to achieve high accuracy. Section 4 contains finding pupil centre, iris circle and eye state steps. In Section 5, we evaluate the performance of the new method using experimental results. Finally, we present conclusion in Section 6.

2 Proposed face detection algorithm

The block diagram of the proposed eye state (open/closed) detection algorithm is illustrated in Fig. 1. In the following, we explain each part in detail.

At first step, we acquire image. In this research, images are read from database.

2.1 Colour compensation

The illumination of skin colour is mainly because of the highlights and shadows caused by light source. For colour compensation, we use the algorithm of [18] that includes the following steps:

1. Get the intensity component of the colour image as

\[ I = \frac{R + G + B}{3} \quad (1) \]

2. Sort the intensity components of all pixels from the largest one to the smallest one.

3. Set to 255, all RGB components of the pixels whose light values are in the largest q% of region. Suppose that the largest intensity value of q% is I_0, convert the RGB components of the rest 1-q% pixels to

\[ R' = R \times 255/I_0; \quad G' = G \times 255/I_0; \quad B' = B \times 255/I_0 \quad (2) \]

2.2 Face detection

For eye detection with high accuracy rate, we need an image in which face area is isolated from background. The characteristics of face skin of different humans are almost similar. The only major difference between people with lighter complexion and those of darker complexion is that, in different lighting conditions, images pixels have different values. In order to implement the new method in real-time system, we adopt skin colour detection as the first step of face detection.

Typically, images are acquired in the RGB colour space. However, in the RGB model, brightness (luminance) is not independent of colour. That is, one specific colour under different lighting conditions is appeared as different colours. This is a problem in detection of human faces, which causes RGB model not useful for face detection. Colour spaces used in skin colour segmentation are YC_bC_r [12, 19, 20], HSV [21] and normalised RGB (rgb) [22, 23]. Since

\[
\text{Image Acquisition} \\
\downarrow \\
\text{Colour Compensation} \\
\downarrow \\
\text{Face Detection} \\
\downarrow \\
\text{Eye Region Detection} \\
\downarrow \\
\text{Estimate Pupil Centre} \\
\downarrow \\
\text{Refine Pupil Centre and Iris Circle Detection} \\
\downarrow \\
\text{Determine Eye State}
\]

Fig. 1 Block diagram of the proposed eye state analysis algorithm
The colour space transform is faster than the other approaches, we use it to detect human skin. The experimental results show that skin pixels have lower $C_b$ and higher $C_r$ values. If $YC_bC_r$ colour space is used, many errors occur in skin pixels detection. The authors in [12] presented a non-linear transformation to overcome this problem. In their model, the chrominance components ($C_b$ and $C_r$) are expressed as functions of luminance component ($Y$). The obtained new colour space is $YC'_bC'_r$, where $C'_b$ and $C'_r$ values for skin pixels follow the elliptical model as [12].

$$\frac{(x - ec_x)^2}{a^2} + \frac{(y - ec_y)^2}{b^2} = 1$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} C'_b - c_x \\ C'_r - c_y \end{bmatrix}$$

where $c_x = 109.38$, $c_y = 152.02$, $\theta = 2.53$ (in radian), $ec_x = 1.6$, $ec_y = 2.41$, $a = 25.39$ and $b = 14.03$ which are computed from the skin cluster in $C'_bC'_r$ space. More details are presented in [12]. The results presented in [12] indicate that in this case, skin pixels are better separated from background pixels. We use the above model to obtain a binary image from the input image in which, the white pixels represent the skin region. This binary image is denoted as FaceMask. However, sometimes other regions in the image have the same colour specifics as the skin region, consequently they are appeared as skin region in FaceMask. To overcome this problem, we consider an area around the biggest connected component in the binarised image (i.e. FaceMask) as the search area. This area is shown in Fig. 2b (rectangle 1). Then, we propose to create a rectangle with a predefined length and width. We shift this rectangle in the search area and compute the number of white pixels inside each shifted rectangle. Whenever the number of white pixels inside a specific shifted rectangle is maximum, that rectangle will be considered as face area. This rectangle is demonstrated in Fig. 2b (rectangle 2).

As sometimes face size is smaller than the rectangle size; hence, we perform morphological operations such as closing and erosion [24]. However, to reduce the computational complexity, we perform morphological operations only on the search area. In doing so, the unnecessary regions in the rectangle are removed and the holes appeared in eye regions of the FaceMask are removed. Next, we create another rectangle in order to
bound the biggest connected component. However, sometimes face rotation may cause that eyes be placed in the border of face area, which may result in problem in eye regions detection. Hence, we release some extra area in the right and left of bounding rectangle. This rectangle is shown in Fig. 2c (red rectangle). Finally, we select the area inside the rectangle in the input coloured image to obtain face area (Fig. 2d).

3 Proposed eye region detection method

After finding face area, we try to find eye regions. Eye detection is a very important step in eye state analysis. An accurate eye detection algorithm provides enough information for fatigue detection, that is, opening or closing of eye. Experimental results show that eyes are located in the upper half part of face region. So, in order to increase the speed of detection, we remove the lower 40% of the face area, and remain the upper 60%; then we look for eyes in this area. Under low illumination, the ability to discriminate eyes from skin degrades. Thus, the image enhancement method proposed in [25] is used to improve the separability. This method uses fuzzy set to enhance the image; it is simple and achieves better results than the colour histogram equalisation. It also preserves the hue component well. We briefly describe the steps of the method of [25] below:

1. Transform the original RGB space into HSI space.
2. Enhance intensity component using fuzzy rules.
3. Enhance saturation component.
4. Transform the HSI data to RGB data.

Fig. 3 demonstrates the result of enhancement method on the upper 60% of face area for hybrid image. We observe that the enhanced image has uniform illumination.

After enhancement, we detect eye regions. The block diagram of the proposed eye regions detection method is shown in Fig. 4.

In [12], the authors presented EyeMap algorithm to determine eye regions in different light conditions. They used YCbCr colour space to detect eyes. This method first builds two EyeMaps, one from chrominance components (EyeMapC) and the other from luminance component (EyeMapL). These two maps are then combined into a single map. The EyeMapC is based on the observation that high $C_b$ and low $C_r$ values are found around eyes. EyeMapC is obtained as

$$\text{EyeMapC} = \left( C_b^2 + C_r^2 + (C_b/C_r) \right)/3 \quad (5)$$

where $C_b$ and $C_r$ are the chrominance components of YCbCr colour space and $\hat{C}_r = 255 - C_r$. The values of $C_b^2$, $C_r^2$ and $C_b/C_r$ are normalised to the range [0, 255]. The 1/3 scaling factor is used to ensure that the resulted EyeMapC remains within the range of [0, 255].

Since eyes usually contain both dark and bright values in the luminance component, so grey-scale dilation and erosion (non-flat) with ball structuring element are used to construct the EyeMapL from the luminance component as

$$\text{EyeMapL} = \frac{Y(x, y) \oplus g_b(x, y)}{Y(x, y) \Theta g_b(x, y) + 1} \quad (6)$$

where $g_b$ represents the ball structuring element and $\oplus$ and $\Theta$ denote the grey scale dilation and erosion operations, respectively. The value ‘1’ is added to the denominator of (6) to avoid division by zero. Then, EyeMapL is smoothed by Gaussian lowpass filter. Finally, EyeMapL is multiplied by EyeMapC to produce the final EyeMap.

$$\text{EyeMap} = (\text{EyeMapC}) \times (\text{EyeMapL}) \quad (7)$$

This EyeMap algorithm is not time consuming. Also, it can determine eye regions very well. In order to enhance the separation of eyes from other facial components, we apply morphological top-hat operation with ball structure element. After determining eye regions by EyeMap, the optimum threshold value is obtained by Otsu’s method [24]. However, for better seperation, we propose to multiply the optimum threshold by a constant value and then convert the grey-scale image into the binary one (EyeBin). Next, we apply some operations on the connected components to obtain eye regions, which are presented in the following:

1. Remove the connected components touching the border.
2. Remove the connected components having low or large number of white pixels. The reason is that noise can produce some unwanted connected components. Therefore this approach removes some of these unwanted connected components.

![Fig. 3 Image enhancement before eye regions detection](image_url)

*a* Original upper 60% of face area

*b* Enhanced image
3. Apply some shape and geometric rules such as aspect ratio and solidity to remove the other undesired connected components. In this way, the connected components with low solidity or connected components with very low or very high aspect ratios are removed. A measure of solidity of object can be obtained as the ratio of the object area to the area of the object convex hull.

4. Remove eyebrows if they are still present in the image. For this purpose, we start with one connected component and compute the horizontal and vertical distances among the centre of mass of the mentioned connected component and those of other connected components. If the computed distances satisfy the predefined conditions, that connected component is removed. We apply this approach for all connected components.

5. To increase the accuracy of detection, we calculate the horizontal and vertical distances between the centres of masses of all possible pairs in the connected components. Those pairs that have high vertical distance or low or high horizontal distance are ignored from the possible eye regions.

6. If only one pair of connected components satisfies the conditions of step 5, we accept that pair as eye regions. If more than one pair of connected components satisfy the conditions of step 5, we apply step 7. If no pair satisfies the conditions, we reduce the constant value that is multiplied by the optimum threshold and start from step 1. We reduce the constant value five times. If after the last try, no pairs of connected components satisfy distance conditions, our system fails in eye detection (this case rarely happened in the experiments).

7. We define the following parameter to detect eyes

$$K_i = \tilde{C}_r(1 + \text{EyeMap}) + C_b \quad i = 1:n$$

where \( n \) is the number of connected component pairs that satisfy step 5 conditions. For noise reduction, we apply an averaging filter on image. Then, for any pair of connected components \( i \) that satisfies step 5 conditions, we sort the \( \{K_i\} \) in descending order (denoted as \( K_{s,i} \)), and calculate the sum of top 30 values of \( \{K_{s,i}\} \) as

$$\text{SUM}_i = \sum_{j=1}^{30} K_{s,i}(j) \quad i = 1:n$$

8. Finally, the pair with the largest \( \text{SUM}_i \) is considered as eye region.

Our results show that in many cases, eye regions are obtained before the step 7. Hence, this step is rarely used. A typical image and the result of applying the new method is shown in Fig. 5. It is observed that the proposed method can find eye regions accurately.

4 Proposed eye state detection method

In this section, we describe different steps of our eye state detection algorithm. They are the last three stages of the proposed algorithm in Fig. 1.

4.1 Estimation of pupil centre

Pupil centre is considered to have the lowest \( C_r \) value in the eye region. Hence, finding pixels with the minimum \( C_r \) seems to be a good choice for determining pupil centre. However, our experiments showed that this method does not always yield correct results. In [1], the authors consider that the pupil centre is the centre of mass of the eye region obtained from EyeMap. Although this method achieves better results than the minimum \( C_r \) method, but it does not accurately determine the pupil centre. Hence, we need to modify it. In the following, we explain the modifications.

4.2 Refining pupil centre and iris circle detection

To refine the pupil centre and iris circle detection, it is necessary to extract pupil and iris from skin. In [13], the authors have used RGB colour space to separate iris pixels from skin pixels. In RGB colour space, the illumination component is not separated from the other components and as a result a little variation in light condition causes large variations in RGB components. Therefore we propose to use \( YC_bC_r \) colour space in which the illumination component is separated from the chrominance components.
The proposed IrisMask is based on $C_b$ and $\tilde{C}_r$ components and is created as follows

$$\text{IrisMask} = \sum_{k \in \{C_b, \tilde{C}_r\}} (I_k - m_k)^2$$

where $I_k$ refers to the values of $C_b$ and $\tilde{C}_r$ components and $m_k$ indicates the mean of each component. As a result, pixels with high intensities in the IrisMask are concentrated in the pupil and iris region, as shown in Fig. 6a. Next, IrisMask is normalised to $[0, 1]$. Here, we propose some modifications for better separation of pupil and iris from the skin region. We first apply top-hat filter with ball structuring element. Then, we apply median filter to eliminate the noise of pupil. The resulted image is depicted in Fig. 6b.

Now, with this IrisMask, we refine pupil centre and then detect iris circle. Approximated pupil centre has been already obtained using the centre of mass of eye region. We note that the shape of iris is circle and the experiments [1] showed that the iris circle radius ($R_1$) is proportional to $1/10$th of the distance of between the centres of two eyes.

To determine the exact pupil centre, we create three spatial circular averaging filters with radius sizes of $R_1 - 1$, $R_1$ and $R_1 + 1$. A circular averaging filter with radius $R$ is of size $(2R + 1) \times (2R + 1)$ pixels and its centre is located at the pixel $(R + 1, R + 1)$. It is expressed as

$$\text{CirFil}(R) = \begin{cases} \frac{1}{S_0} & \text{if } (x - (R + 1))^2 + (y - (R + 1))^2 \leq R^2 \\ 0 & \text{otherwise} \end{cases}$$

where $S_0$ is the normalisation constant and is chosen in a way that satisfies $\sum_{x,y} \text{CirFil}(R) = 1$, $x$, $y$ denote the locations of each pixel in the filter, and

$$1 \leq x \leq (2R + 1); \quad 1 \leq y \leq (2R + 1)$$

Fig. 7 demonstrates a spatial circular averaging filter. We apply the three filters to the IrisMask. For each filter, we find the average intensity of IrisMask pixels in the circle, where the centre of circle is the pixel with the maximum value in the filtered image. Then, we calculate the mean of the obtained average values. The radius whose average value has the minimum difference with the mean value is selected as the iris radius.

Fig. 5 Different steps of our eye region detection method

a Original image
b 60% upper of face image with enhancement
c EyeMap
d EyeBin
e Eye regions candidates after applying steps 1 and 2
f Eye regions candidates after apply steps 3 and 4
g Pairs satisfying distance conditions
h Selected eye regions in the corresponding original image
Figs. 8a and b. show the estimated pupil centre and refined pupil centres along with iris boundary in the original eye region image and its corresponding IrisMask.

It should be noted that the circular averaging filters we created are like [1]. However, the authors in [1] shift the circle in the area around the estimated pupil centre and the circle in which the sum of grey levels, has the largest value, is selected as the iris circle. A disadvantage of the method of [1] is that when the estimated pupil centre is far from the exact pupil centre, the iris circle is not detected correctly. As another disadvantage, when the circle radius becomes larger, the sum of grey levels inside the circle also becomes larger and the probability that larger circle is selected as the iris circle, is more than that of small circles. Although in our work, to overcome these problems we search for the pupil centre in the whole of connected components that are detected as eye regions. Further, we calculate the average of pixels in the circle.

4.3 Determination of eye state

We note that when eye is closed, the pupil centre is probably located under the skin region around the eye, so it cannot be seen and iris circle contains more skin pixels. Therefore after detecting iris circle, to discriminate open eye from closed one, we must extract iris from skin.

The IrisMask, obtained previously (Fig. 6), shows that iris pixels have higher grey level values than the skin pixels, that is, they are brighter. Thus, we can separate iris pixels from the skin pixels by selecting a proper threshold value. To find the optimum threshold value (Th), we use Otsu’s method [18]. Then, we convert IrisMask to binary image to obtain IrisBin as

\[
\text{IrisBin} = \begin{cases} 
1 & \text{IrisMask} \geq \text{Th} \\
0 & \text{otherwise}
\end{cases}
\]

To remove noise from binary image, we apply binary morphological closing operation. However, sometimes eyelids and other parts of skin are also visible in the IrisBin as shown in Fig. 8c. To remove them, we first apply erosion and then dilation operations with two different structure elements. The result is demonstrated in Fig. 8d. The eye is detected as open if the number of white pixels inside the iris circle is more than the number of black pixels. That is, the ratio of white pixels to the total pixels of iris circle is greater than 0.5. Otherwise, eye is detected as closed.

5 Experiments and results

5.1 Database

The performance of the proposed system was evaluated on four different databases. Subset of FERET [14], IMM [15], Aberdeen [16] and CVL [17] databases were used in our experiments.

In the FERET database, images are from different descents, face positions and rotations. Also lighting conditions and face size vary too much. We select 4517 images from this database randomly. The selected images have different face rotations represented by \(fa, fb, ql, qr, rb\) and \(rc\) indices, which stand for regular frontal image, alternative frontal image that taken shortly after the corresponding \(fa\) image, quarter left (head turned about 22.5° left), quarter right (head turned about 22.5° right), random image (head turned about 15° left) and random image (head turned about 15° right). All images have size of \(512 \times 768\) pixels and are in portable
pixel map format. Besides, the selected images are with and without glasses.

IMM database includes 40 persons with six different positions and light conditions. 222 images of this database are coloured and 18 ones are grey level. In this experiment, we used all 222 coloured images. The six images of each person are labelled by facial expressions, lighting conditions and face rotations as normal, happy, 30° rotation to the person’s right, 30° rotation to the person’s left, spot light added at the person’s left side and arbitrary expression. All images are of size 640 × 480 pixels and have JPEG format.

Aberdeen database consists of 687 images with face rotations. Many changes in this database are related to lighting conditions, but no descriptions and no categories are reported for these variations. To evaluate the algorithm, we choose all images of this database except those (22 images) in which the face rotations are 45°, 67° or 90°, where the two eye regions are not completely appeared. Therefore we used 665 images that have different sizes and JPEG format. In this database, images have different sizes. The width and height of images vary in the ranges [336–624] and [464–560] pixels, respectively.

CVL database consists of images from 114 persons. There are seven images for each person. These images are labelled by side view that includes far left, angle 45°, angle 135° and far right, frontal view that includes serious expression, smile (showing no teeth) and smile (showing teeth). These images were taken under uniform illumination and no flash and projection screen are in the background. All images are of size 640 × 480 pixels and have JPEG format. To evaluate the new method, we choose images that have frontal view, because our method is based on the presence of both eyes.

### 5.2 Face detection

In Table 1, the results of the proposed face detection algorithm evaluated on four databases are presented. If both eye regions appear in the upper 60% of detected face area.

<table>
<thead>
<tr>
<th>Database</th>
<th>Number of images</th>
<th>Accuracy, %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Proposed method</td>
<td>Viola Jones method [26]</td>
</tr>
<tr>
<td>FERET</td>
<td>4517</td>
<td>99.40</td>
</tr>
<tr>
<td>Aberdeen</td>
<td>665</td>
<td>100</td>
</tr>
<tr>
<td>IMM</td>
<td>222</td>
<td>99.55</td>
</tr>
<tr>
<td>CVL</td>
<td>359</td>
<td>98.89</td>
</tr>
<tr>
<td>Total</td>
<td>5743</td>
<td>99.36</td>
</tr>
</tbody>
</table>

---

Fig. 8  Illustration of pupil centre, iris circle and iris mask detection steps

a Original eye region image, dot marker: estimated pupil centre, plus marker: refined pupil centre, white circle: iris boundary
b Pupil centre and iris boundary shown on IrisMask
c IrisMask after binarisation (IrisBin)
d IrisBin after applying morphological operations
image, we accept that the face area is detected correctly. The experimental results demonstrate that our algorithm with fixed size rectangle can find face area very well in colour images.

As mentioned previously, in the selected databases, the size of acquired image, the size of face area, head pose and lighting conditions vary too much. Since the proposed face detection algorithm is applied to the whole captured image, the search area for face detection must be selected in a way that it is not too small or too large. We choose a fixed size rectangle and its position is determined by the input image. The size of this rectangle should be large enough to cover the entire face region, but not too large to include non-face regions. In our experiments, we choose a rectangle of size 120x120 pixels, which is large enough to cover the entire face region in most cases.

Fig. 9  Steps of the proposed face detection algorithm in the case that there are more than one face in the captured image

a Original image
b FaceMask (binarised image), rectangle (1): search area, rectangle (2): region that has maximum number of white pixels
c Search area, rectangle: face region
d Face region cropped from the original image considering part c

Fig. 10  Steps of the proposed face detection algorithm in the case that there are more than one face in the captured image

a Original image
b FaceMask (binarised image), rectangle (1): search area, rectangle (2): region that has maximum number of white pixels
c Search area, rectangle: face region
d Face region cropped from the original image considering part c
detector is based on skin colour, it can also detect frontal faces with rotation. When face has rotation from 0 up to 90° to the right or left, our method can detect face area. Moreover, if the pose of frontal face changes to up or down, it can detect face area.

As mentioned in Section 1, the proposed algorithm was designed based on the assumption that there is one face in the image. Therefore to evaluate the performance of the

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Performance of the proposed eye detection method and the methods evaluated on FERET database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Total images</td>
</tr>
<tr>
<td>fa</td>
<td>1096</td>
</tr>
<tr>
<td>fb</td>
<td>1093</td>
</tr>
<tr>
<td>ql</td>
<td>702</td>
</tr>
<tr>
<td>qr</td>
<td>698</td>
</tr>
<tr>
<td>rb</td>
<td>320</td>
</tr>
<tr>
<td>rc</td>
<td>608</td>
</tr>
<tr>
<td>total</td>
<td>4517</td>
</tr>
</tbody>
</table>

Table 3 | Accuracy of the new eye detection method and method in [30] evaluated on CVL database |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Total images</td>
</tr>
<tr>
<td>serious expression</td>
<td>114</td>
</tr>
<tr>
<td>smile – showing no teeth</td>
<td>114</td>
</tr>
<tr>
<td>smile – showing teeth</td>
<td>111</td>
</tr>
<tr>
<td>total</td>
<td>339</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of images selected from CVL database</th>
<th>Accuracy, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guan [30]</td>
<td>342</td>
<td>96.2</td>
</tr>
<tr>
<td>our method</td>
<td>339</td>
<td>95.32</td>
</tr>
</tbody>
</table>

The proposed face detection algorithm, we used the databases in which there is a single person in the input image. Noting Section 2.2, we consider search area around the biggest connected component in the FaceMask and shift the rectangle in the search area. In the cases that there are more than one face in the image, with high probability the biggest connected component belongs to the closest face to the camera, hence it is detected as the face area. In the

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Performance comparison of the proposed eye detection method and the other methods examined on IMM database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Accuracy, %</td>
</tr>
<tr>
<td>Our method</td>
<td>Tabrizi et al. [1]</td>
</tr>
<tr>
<td>normal</td>
<td>100</td>
</tr>
<tr>
<td>happy</td>
<td>100</td>
</tr>
<tr>
<td>30° to person’s right</td>
<td>100</td>
</tr>
<tr>
<td>30° to person’s left</td>
<td>97.3</td>
</tr>
<tr>
<td>spot light added</td>
<td>100</td>
</tr>
<tr>
<td>arbitrary</td>
<td>96.4</td>
</tr>
<tr>
<td>total</td>
<td>98.65</td>
</tr>
</tbody>
</table>

Table 5 | Accuracy of the proposed eye detection method evaluated on Aberdeen database |
| Database | Total images | Accuracy, % |
| Aberdeen | 665 | 94.29 |

Table 6 | Performance comparison of the proposed eye detection algorithm on four databases |
| Database | Total images | Accuracy, % |
| FERET | 4517 | 96.98 |
| Aberdeen | 665 | 94.29 |
| IMM | 222 | 98.65 |
| CVL | 339 | 95.32 |
| Total | 5743 | 96.3 |
images captured from the driver, in most cases the driver’s face is the closest one to the camera, and there is rarely another face than the face of driver closer to the camera. However, in the case that the faces have the same distances from camera, the face with more number of white pixels in the FaceBin is detected as the face area.

To verify the above statement, Fig. 9 demonstrates an image in which there are two faces where their distances from camera are different. The steps of face detection are the same as shown in Fig. 2. It is observed that the closer face is detected as the face area. In Fig. 10, the two faces have the same distances from camera, but the face that has

<table>
<thead>
<tr>
<th>Database</th>
<th>Total images</th>
<th>Accuracy, %</th>
<th>Accuracy, %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Face detection</td>
<td>Eye regions detection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With rectangle</td>
<td>Without rectangle</td>
</tr>
<tr>
<td>FERET</td>
<td>4517</td>
<td>99.40</td>
<td>97.37</td>
</tr>
<tr>
<td>Aberdeen</td>
<td>665</td>
<td>100</td>
<td>98.80</td>
</tr>
<tr>
<td>IMM</td>
<td>222</td>
<td>99.55</td>
<td>96.65</td>
</tr>
<tr>
<td>CVL</td>
<td>339</td>
<td>98.80</td>
<td>97.64</td>
</tr>
<tr>
<td>total</td>
<td>5743</td>
<td>99.96</td>
<td>97.60</td>
</tr>
</tbody>
</table>

Fig. 12. Effect of rectangle in face and eye regions detection for a face image with extra regions

a Original image
b FaceBin
c 60% upper of detected face with eye regions area using rectangle
d 60% upper of detected face area without using rectangle
greater connected component is selected as the face area. It is worth mentioning that the proposed method was designed for driver fatigue detection; hence detection of one face is the main issue.

To show the efficiency of the proposed face detection algorithm, we compare the obtained result with the well-known Viola Jones [26] face detector. The Viola Jones face detector is a boosting-based face detector that involves the calculation of Haar-like features and weak recognition by using AdaBoost classifiers. This method is computationally efficient. However, its performance depends on the amount of training data, and it may not give a precise position of face. Further, Viola Jones is a training-based method and need data for training, while our method does not need any training data. In order to use Viola Jones face detector, we used OpenCV 2.1 provided in [27]. If several objects are detected as face area in the image, the biggest object is considered as face area. The results are shown in Table 1. We observe that the proposed method has higher accuracy. Our experiments show that most errors occurred in the Viola Jones face detector are because of the rotation of face area to left or right or different poses of face area. In Fig. 11, some images from different databases are shown that demonstrate the success of proposed method in finding face, while Viola Jones method fails.

5.3 Eye region detection

The proposed method for eye regions detection is based on the assumption that both eye regions are appeared in the face area. Consequently, face rotation has serious effect on eye regions detection. Therefore in the selected databases, we use images in which both eye regions are present. In this section, for more clarification, the results of eye region detection are presented against face rotation position and facial expression in more details.

In Table 2, we have shown the results of applying the new eye detection algorithm on FERET database. It is observed that the new algorithm achieves rather the same accuracy rate for different face rotations. For comparison we have also presented the results of previous works. It is seen that our method has better performance than the methods of [13, 28, 29]. The binary tree approach used in [29] is time consuming and its accuracy rate depends on the initial position of template. The recently proposed method [28] is based on applying fast independent component analysis (ICA), which needs training data. However, our method does not need training data and is not time consuming. In [13, 28, 29], the authors have used 1000, 1500 and 311 images from FERET database, respectively. Since they have used some images from FERET database not all of them, it is probable that the selected images were different in different references. However, in our work, we used a large number of images (4517 images) which is much more than the above works. Consequently, our selected images cover all kinds of variations of images. Therefore our results are general and validate the robustness of proposed method in comparison with other methods.

In Table 3, the results of eye detection method evaluated on CVL database are presented. Most of the errors occurred in this database are because of images with glasses which cause bad reflection. However, the results show the efficiency of our method. Comparing the accuracy rates of the new method with that of [30], we observe that our

![Fig. 13. Effect of rectangle in face detection for a face image with some separated connected regions](image-url)

- a Original image
- b FaceBin
- c 60% upper of detected face area using rectangle
- d 60% upper of detected face area without using rectangle
method has about 0.9% less accuracy rate. The method of [30] is based on multi-cue facial information and makes use of a combination of features such as colour, edge, intensity and some prior knowledge to extract eyes from image. It is worth mentioning that the method of [30] achieves 93.65% accuracy for AR database [31] which contains 3276 images from 114 subjects and 94.82% for their own database which has 336 images.

In Table 4, we have compared our eye detection algorithm with other methods evaluated on IMM database. We observe that the new method outperforms the methods of [1, 32].

In Table 5, we have demonstrated the results on Aberdeen database which has different lighting conditions. It is seen that the new method yields high accuracy rate.

Table 6 shows the performance of the new eye detection algorithm on the four mentioned databases. We observe that the average accuracy of the proposed method is 96.3% for 5743 images. In [13], authors have obtained 87.95% detection rate on 1353 images. In [30], the average rate for three databases with 3974 images is obtained as 94.97%. Also, comparing the results with those of mentioned in [28] which are in the range of [93.3–98.4%], we conclude that the new method has higher performance.

In the face detection step, we have used rectangle to avoid false skin detection and overcome bad lighting conditions. If false skin detection occurs, the other areas of image are considered as the face area. In this case the size of detected face area increases and other regions appear in the face area which degrade the performance. Hence, we use rectangle for precise detection of face area which also results in high accuracy rate in the eye detection step. If rectangle is not used and the biggest connected component is selected as the face area, we involve two problems; (1) the first one is wrong eye regions detection because other regions appear in the face area, (2) due to the bad lighting conditions, the whole of face area probably does not appear in the FaceBin and the biggest connected component does not cover all of face area. In this case it is probable that eye regions do not appear in the upper 60% of the face area. However, by employing rectangle, face detection accuracy increases in bad and non-uniform lighting conditions, and eye regions appear in the upper 60% of detected face area. Also, by removing the unnecessary regions from the face area, eye regions detection accuracy increases.

<table>
<thead>
<tr>
<th>Database</th>
<th>Total images</th>
<th>With top-hat and multiplying optimum threshold value by constant value</th>
<th>Without top-hat and multiplying optimum threshold value by constant value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FERET</td>
<td>4517</td>
<td>96.98</td>
<td>93.80</td>
</tr>
<tr>
<td>Aberdeen</td>
<td>665</td>
<td>94.29</td>
<td>90.69</td>
</tr>
<tr>
<td>IMM</td>
<td>222</td>
<td>98.65</td>
<td>93.69</td>
</tr>
<tr>
<td>CVL</td>
<td>339</td>
<td>95.32</td>
<td>92.63</td>
</tr>
<tr>
<td>Total</td>
<td>5743</td>
<td>96.3</td>
<td>93.21</td>
</tr>
</tbody>
</table>

Table 8: Effect of top-hat and multiplication of optimum threshold by constant value in eye regions detection step.

<table>
<thead>
<tr>
<th>Database</th>
<th>Total images</th>
<th>Proposed method</th>
<th>Method of [13]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FERET</td>
<td>4517</td>
<td>96.03</td>
<td>92.80</td>
</tr>
<tr>
<td>Aberdeen</td>
<td>665</td>
<td>93.83</td>
<td>91.15</td>
</tr>
<tr>
<td>IMM</td>
<td>222</td>
<td>95.94</td>
<td>90.54</td>
</tr>
<tr>
<td>CVL</td>
<td>339</td>
<td>94.03</td>
<td>91.74</td>
</tr>
<tr>
<td>Total</td>
<td>5743</td>
<td>95.59</td>
<td>92.30</td>
</tr>
</tbody>
</table>

Table 9: Performance of the proposed eye state detection algorithm and the method of [13].

*Fig. 14* Performance comparison of the proposed method and the method of [13] in separating iris pixels from other parts of eye window:

(a) Original eye window image
(b) IrisMask obtained by the method of [13]
(c) Binary of IrisMask obtained by the method of [13]
(d) IrisMask obtained by the proposed method
(e) Binary of IrisMask obtained by the proposed method.
To show the effect of rectangle, we have compared the performance of face and eye regions detection steps with and without rectangle. Table 7 depicts the results. It is obvious that using rectangle yields more accuracy in face and eye detection steps and its effect is more visible in the eye detection stage. Further, as examples, Figs. 12 and 13 show the face and eye regions detection performances with and without using rectangle. In Fig. 12, the first problem is demonstrated and in Fig. 13, the second issue is shown.

Table 7 and the above mentioned figures verify the importance of using rectangle.

To provide better separation of eye regions from other facial components, in Section 3 we proposed the scheme that converts EyeMap to EyeBin. In brief, at first top-hat is applied to the EyeMap and then the optimum threshold value obtained by Otsu’s method is multiplied by a constant value. To demonstrate the effectiveness of this scheme, we compared the results of eye regions detection with and without rectangle. Fig. 15 shows the eye regions detection performances with and without using rectangle. In Fig. 15, the first problem is demonstrated and in Fig. 13, the second issue is shown.

**Fig. 15**  Performance of the new method on some images selected from four different databases with different descents and positions

- a Original images with detected eye regions
- b Eye images with pupil centre and iris circle
- c Binarised IrisMask with iris circle

To show the effect of rectangle, we have compared the performance of face and eye regions detection steps with and without rectangle. Table 7 depicts the results. It is obvious that using rectangle yields more accuracy in face and eye detection steps and its effect is more visible in the eye detection stage. Further, as examples, Figs. 12 and 13 show the face and eye regions detection performances with and without using rectangle. In Fig. 12, the first problem is demonstrated and in Fig. 13, the second issue is shown.

**Fig. 16**  Some images from different databases with wrong detected eye regions
Fig. 17  Performance of the proposed method evaluated on several images with different illumination conditions and poses in indoor and outdoor environments

Face area and eye regions are demonstrated by large and small rectangles, respectively. Eye image (with pupil centre and iris circle) along with the binarized IrisMask (with iris circle) are shown at the corners of the original images.
without considering the scheme. The results presented in Table 8 indicate that applying top-hat and changing the optimum threshold value increase the accuracy of eye regions detection (3–5% improvement).

We have used some hard-coded thresholds in eye regions detection algorithm (in steps 2, 3–5 of Section 3). These thresholds are obtained by different experiments on several images from different databases. The obtained thresholds then remain fixed in the experiments on other images. In the selected databases, face area has different sizes and rotations. Hence, eye regions characteristics such as size, solidity, aspect ratio, horizontal and vertical distance between eyes, and distance between eyebrows and eyes vary greatly from one image to another. Considering the results, it is concluded that the obtained thresholds are robust against variations and achieve good performance.

5.4 Eye state detection

Eye state analysis is the final step of the proposed algorithm. In this part we present the results of eye state analysis. As mentioned before, in order to detect eye state, we consider the pixels inside the iris circle. Then, based on the percentage of white pixels, the eye state is determined. To judge whether the result is true or false, one way is by observing of eye in the image, deciding about its state (open or closed), and comparing with the state obtained from the algorithm. For half open eyes for more accuracy, an alternative way to determine the actual percentage of opening of eye and consequently its state is that, we first find the iris circle manually. Then, the pixels inside the iris circle are converted to the binary iris in a way that if the pixel belongs to iris, it is set to 1 otherwise it is set to 0. By this manner, the actual percentage of eye opening (ratio of white pixels to all pixels in the iris circle) is determined and then the decision about the state is compared with the result of the algorithm. In Table 9, we have presented the results of eye state detection method. It is observed that the proposed algorithm can detect eye state with high accuracy.

To show the efficiency of the proposed eye state detection algorithm that uses YCbCr colour space, we have compared its performance with the method of [13] that uses RGB colour space. The weakness of RGB colour space in separating iris pixels from skin pixels was explained in Section 4.2 and to overcome this weakness, we proposed YCbCr colour space. The obtained results for both methods are presented in Table 9. It is observed that the proposed method has 3–4% better efficiency in separating iris from other regions of eye window. In Fig. 14, the efficiency of the proposed algorithm in comparison with the method [13] is demonstrated with a typical eye window image.

In Fig. 15, we have presented some images from different databases with different eye states which demonstrate the performance of the new algorithm. Fig. 16 shows some images that our system fails in finding eye regions. It is observed that most of errors are because of the people with glasses.

We have examined the performance of the proposed method on several images which differ from the images of the databases used in this study. The captured images have various conditions such as very low or high illuminations, different poses and indoor and outdoor environments. The results are presented in Fig. 17. It is observed that the proposed algorithm achieve high performance.

In this work, we proposed to employ rectangle to overcome lighting conditions which results in higher accuracy rate in detection of face area and eye regions. We also applied several modifications in the eye detection step to achieve higher performance. Moreover, we used YCbCr colour space in eye state detection instead of RGB colour space that reduces the effect of lighting conditions. As the results show, the proposed schemes in different stages achieve higher accuracies on more than 5700 images of four different databases when compared with the previous works. Further, our work does not need training data, hence it is computationally efficient.

6 Conclusion

In this paper, we proposed an efficient algorithm for determining eye state. To find eye state, we first extracted face region from the input image. Then, we found eye regions based on EyeMap algorithm and finally obtained pupil centre, iris circle and eye state. In each step, we proposed new algorithm and some modifications to achieve better results. The new method does not require training data and is computationally efficient. Our experiments were performed on 5743 images of FERET, CVL, IMM and Aberdeen databases. In the selected databases, images size, lighting conditions, descent of people, face area size and eye state are variable. In face detection step, we achieved 99.36% accuracy, which shows the robustness of the proposed face detection algorithm. In eye regions detection stage, the proposed algorithm yields the average accuracy of 96.3%. Moreover, the new scheme achieves better results than the other methods evaluated on the mentioned databases. In the eye state detection step, we proposed a new algorithm which uses YCbCr colour space and achieves accuracy of 95.59%.

7 Acknowledgment

This work was supported by Iran National Science Foundation (INSF) under contract no. 90001153.

8 References

3 Huang, W., Marianni, R.: ‘Face detection and precise eyes location’. Proc. Int. Conf. on Pattern Recognition, 2000, pp. 722–727
14 FERET database available at http://face.nist.gov/colorferet/
15 Aberdeen database available at http://pics.psych.stir.ac.uk/
16 IMM database available at http://www.imm.dtu.dk/~aam/
17 Computer Vision laboratory, Faculty of Computer and Information Science, University of Ljubljana, Slovenia, http://www.lrv.fri.unilj.si/facedb.html/
27 OpenCV 2.1 available at http://codeproject.com