

# Easymatch- An Eye Localization Method for Frontal Face Images Using Facial Landmarks

Muhammed Emre COLAK, Asaf VAROL

**Abstract:** Eye detection algorithms are being used in many fields such as camera applications for entertainment and commercial purposes, gaze detection applications, computer-human interaction applications, and eye recognition applications for security, etc. Successful and fast eye detection is an essential step for all these applications in order to achieve good results. There are many eye detection methods in the literature, and most of them rely on the Viola-Jones method to detect the face before localizing eyes. In this paper, a straightforward approach to detect eyes from images which contain a frontal face is proposed. The approach can be used for real-time eye detection using cheap web cameras or other cameras. First, face landmarks are detected from the image, and by utilizing these landmarks; the eye region is determined. The eye radius is estimated by utilizing eye corners. Then, reduced input images are tested with a tailored matching algorithm which does not need image reduction to determine where the eye is.

**Keywords:** eye localization; face detection; image matching; image processing

## 1 INTRODUCTION

Eye detection is a fundamental step for many applications including security, marketing, psychology, human-computer interaction and so on. After detecting a face in an image, the next logical step is to locate the elements of the face such as eyes, mouth, chin, etc. The necessity of locating eyes comes from the importance of eyes because as much as eyes are windows to the outside world, they are also windows to the inside. For example, the eyes have definitive signs to express emotions or mental state. The eye pupil may get bigger when in distress, or it may look upward while thinking and this kind of patterns can be used in creating a psychological profile which can then be used for marketing or treating psychological diseases, etc. One can deduce where one is looking from the positions of his/her eyes. After locating the eye positions, it is also possible to detect gaze. Gaze detection can be used in computer-human interaction or finding answers to questions like where people tend to look first when entering a market or mall or school building, where people tend to look most etc. Also, just like fingerprints, eyes also have a unique print, which is used in identification [1, 2]. By counting the number of blinks in a minute, it is possible to determine if a person is tired or sleepy, which is used in detecting whether the driver is tired or not [3]. All of these applications and many more like these rely heavily on locating eyes, and success of these methods is only as good as the success of the eye detection methods.

Current technologies can be classified into three categories. The first is invasive methods such as electro-oculography methods which need to place electrodes on the skin near the eyes [4] or using contact-lens based on eye coil systems [5, 6], etc. The second are semi-invasive methods which use some type of special light to illuminate the eyes. The last category is non-invasive and uses only images to determine where the eye is [7-17]. The invasive methods generally have a good detection rate, however, in today's world it is not possible to use these methods in many applications because they tend to have a high cost and low mobility. Semi-invasive methods have general usage and good detection rate; however, they also have a high cost. Non-invasive methods generally have a low cost;

however, they have a lower detection rate because they can be affected by many factors such as light condition, makeup, eyeglasses, etc.

Non-invasive methods can be divided into two main categories. The first category contains methods which use A.I. (Artificial Intelligence) methods to determine eye position. A.I. based methods are generally composed of three main steps: creating an eye pattern, using a classifier or statistical model and finally by utilizing these, precise eye positions can be found [8, 16, 17]. The second category uses the general characteristic of an eye by utilizing pixel distribution, pixel values, histograms, etc. [7, 10, 12]. A detailed survey on eye detection and tracking techniques is presented in [18].

Usually, the first step in locating eye location has been to detect the face within the image. In order to detect a face within images, many of the applications use the Viola-Jones method. Viola-Jones method is a well-accepted method which works fast and has a satisfactory success rate. However, to determine the eye region, it has to be used two times; one for detecting the face and a second time to detect the eye region. However, unlike images of the static objects, the image of the eye changes dramatically depending on where the gaze is, this causes the creation of larger eye regions which even contains eyebrows by the Viola-Jones method. After detecting the face, it is possible to use an educated guess or other methods to determine the eye region. However, using educated guess also causes large eye regions, other methods may work, but they are also extra workload for the algorithm.

This study proposes an easy way to localize the eye position in frontal face images by utilizing a template matching based approach and uses facial landmarks to determine a region of interest.

## 2 LOCATING EYE

In the proposed method the most simplistic approach to locate the eye is used. A few steps are usually the same in many of the eye locating methods. First, detecting a face with Viola-Jones method [19], determining general eye areas with an educated guess or using an eye cascade to detect eye region, finally using a novel approach to locate where an eye is as shown in Fig. 1.

## 2.1 Finding Face

The purpose of detecting the face is to ensure that the search area has limited characteristics and appearances. There should not be objects or images like the eye in the search area. Also by reducing the area, the workload of the algorithm can be reduced.

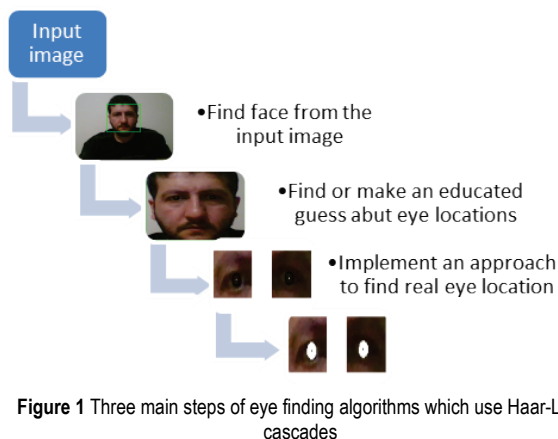


Figure 1 Three main steps of eye finding algorithms which use Haar-Like cascades

Face detection is a decade-old phenomenon, and there are many methods to solve this [20]. In literature, many choose the Viola-Jones method to detect the faces by using Haar-Like cascades. According to the tests carried out on the BioID database, this method manages to successfully detect faces in 1365 images out of 1521 images which means over 89% success rate. There are only eleven false positives out of 1521 images. These false positive results are usually a smaller region within a face so it can be easily cast off by choosing the biggest face from detected faces. For TalkingFace database success rate is 99.1%.

OpenCV (Open Computer Vision) library contains necessary cascade files for this kind of detection purposes, and in this study, frontal face file (haarcascade\_frontalface\_default.xml) is used to detect faces in images. Necessary files and more information about OpenCV could be found on their official web page [21].

Facial landmarks are a relatively new method to determine interest points in a person's face, and this method manages to not only locate the face with precision but also necessary interest points for determining many aspects of a human face [22]. In this study, a detector with 68 markups had been used as shown in Fig. 2. For detecting the facial landmarks, dlib library [23] had been used. Dlib uses an implementation of [24] to create facial landmarks, and it was trained on the iBUG 300-W face landmark dataset [25]. There are different detectors with different markup setups. However; since most of the markups are irrelevant to this study, it is not much important how many markups there are. The markups and numeric coding of markup for landmark points which are used in this study are shown in Fig. 3. In this study, BioID and TalkingFace databases are used for evaluation purposes. Web page of the BioID database could be reached from [26]. Web page of the TalkingFace database could be reached from [27].

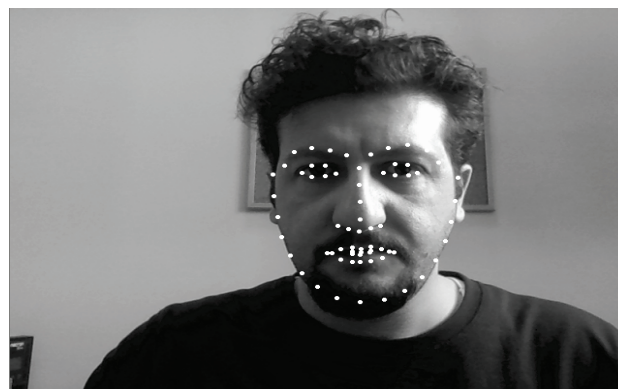


Figure 2 68 landmark points



Figure 3 The numeric markup for landmark points

According to the tests carried out on the BioID database, the facial landmarks successfully detect the faces in the 1509 images from 1521 images, which means more than 99% success rate. According to the test carried out on the TalkingFace database, the facial landmarks successfully detect all faces. There are five false positives in TalkingFace database. However, like the Viola-Jones method, these false positive results are usually a smaller region within a detected face, thus it can be easily discarded by choosing the biggest face from detected faces. The ColorFeret database success rate is also 100%. As seen in the results; using facial landmarks has a much better success rate than the Viola-Jones method.

## 2.2 Determining the Eye Region and Reducing the Image

In order to determine a general eye region, taking an educated guess like breaking image to four square parts and supposing up left half contains left eye and up right half contains right eye can be used [11]. However, this approach would cause a much bigger region of interest than what was needed but ensures that the eye is within that area. Another option is using a Haar-Like cascade or similar detection method to determine where the general eye region is. However, the usage of Haar-like cascades gives poor results. Because the eye itself is moving within the eye region, it makes it very hard for cascades to detect the eye region with high accuracy. Also, using a secondary detection method would increase the computation cost, but may give a smaller eye region.



Figure 4 Created eye region using landmarks and the natural rate of eye and distance between two eye corners

In this study, facial landmarks are used to determine the region of interest. At first, the eye region was tried to be determined using all points as seen in Fig. 3. However, these landmarks are not as accurate as desired, and usually they do not contain the entire eye region. For this reason, these landmarks cannot be used to detect an eye region successfully. However, there is a natural rate between two eye corners and the eye radius. Since facial landmarks include eye corners as landmarks, these corners can be used to determine a small eye region as seen in Fig. 4. In this study, it is seen that errors in eye corner landmarks are not causing many problems in detecting the eye region. First, the distance between eye corners is computed using the Eqs. (1, 2)

$$l_{\text{right}} = \sqrt{(p_1x - p_{10}x)^2 + (p_1y - p_{10}y)^2} \quad (1)$$

$$l_{\text{left}} = \sqrt{(p_9x - p_6x)^2 + (p_9y - p_6y)^2} \quad (2)$$

where  $l_{\text{right}}$  and  $l_{\text{left}}$  is the length between eye corners for right and left eyes,  $p_i$  is the  $i^{\text{th}}$  landmark as shown in Fig. 3,  $p_ix$  and  $p_iy$  are  $x$  and  $y$  coordinates of the landmark point  $p_i$  respectively. In order to determine the eye region of the right eye, top left of the eye region as  $(p_1x, p_1y - 1.2X l_{\text{right}})$ , right bottom of the eye region chosen as  $(p_{10}x, p_{10}y + l_{\text{right}})$ . As for left eye,  $(p_9x, p_9y - 1.2X l_{\text{left}})$  chosen as top left and  $(p_6x, p_6y + l_{\text{left}})$  for bottom right.

In this study, a smaller and more efficient region of interest for the eyes is determined by considering the natural rate between eye corners and the eye itself as seen in Fig. 4. Such a small eye region helps to avoid dark appearances, such as eyebrows and eyeglasses. Thus, it should help to raise the success rate of almost all of the eye detection methods because one of the main problems for eye detection methods is dark appearances like eyebrows or eyeglasses.

The ratio between eye radius and the distance between eye corners is estimated at about 0.22. The radius of the left and right eye can be found using the Eqs. (3) and (4).

$$r_{\text{right}} = l_{\text{right}} \times 2 / 9 \quad (3)$$

$$r_{\text{left}} = l_{\text{left}} \times 2 / 9 \quad (4)$$

### 2.3 Template Matching

The template matching is a process where each pixel in one input image is compared to another image to find if there is a matching part in it. This is done in a windowed manner, and for each window, a similarity value is calculated. Best similarity value determines which of the windows is matched to the input image. There are many methods to calculate similarity value like square difference (5), the normalized square difference (6), correlation (7), normalized correlation (8), etc. In Eqs. (5) to (9)  $R$  is the similarity value,  $I$  input image,  $S$  source image which a match tried to find in it.  $x_s, y_s$  are starting positions of the window and  $x', y'$  pixel positions of the input image.

The circular characteristic of the eyes makes it easy to use in a matching algorithm. In this study, images are reduced to black and white, and then a circle is used as a template for matching. Even though eyes are not fully circle, but an ellipse with a ratio of 1.13 usually this difference is negligible because cameras do not have the necessary resolution to catch that difference. However, in a state where the camera has a good resolution to catch that difference, an ellipse must be used as a template instead of a circle.

$$R(x_s, y_s) = \sum_{x', y'} (I(x', y') - S(x_s + x', y_s + y'))^2 \quad (5)$$

$$R(x_s, y_s) = \frac{\sum_{x', y'} (I(x', y') - S(x_s + x', y_s + y'))^2}{\sqrt{\sum_{x', y'} (I(x', y'))^2 * \sum_{x', y'} S(x_s + x', y_s + y')^2}} \quad (6)$$

$$R(x_s, y_s) = \sum_{x', y'} (I(x', y') * S(x_s + x', y_s + y')) \quad (7)$$

$$R(x_s, y_s) = \frac{\sum_{x', y'} (I(x', y') * S(x_s + x', y_s + y'))^2}{\sqrt{\sum_{x', y'} (I(x', y'))^2 * \sum_{x', y'} S(x_s + x', y_s + y')^2}} \quad (8)$$

A simple input image which will be used in template matching is created using an estimated radius value as seen in Fig. 5. However, this approach does not provide satisfactory results.



Figure 5 Example for input images for template matching

In order to improve the success rate and reduce the workload a modified approach based on template matching is tried. There are two pre-knowledge about the eye; the color of the eye is darker than is its vicinity and the eye has a circular shape. Since the eye has a circular shape if the sum of the points in a circular shape is calculated in a windowed manner, then the position of the window with the lowest sum can be accepted as the position of the eye. For this purpose a list is created which contains the  $x$  and  $y$  coordinates of each white point of the image as seen in the

right image in Fig. 5. Then, the list is used in sliding windows. In each window, the sum of the values of the list points is calculated using the Eq. (9). Then it is determined where the biggest sum is achieved.

$$R(x_s, y_s) = \sum_{i=0}^n S(x_s + list_i x, y_s + list_i y) \quad (9)$$

Using all points within the circle has a good success rate. However, in order to further reduce the workload of the method instead of using all points within a circle reduced points are used for experiment as seen in Fig. 6.

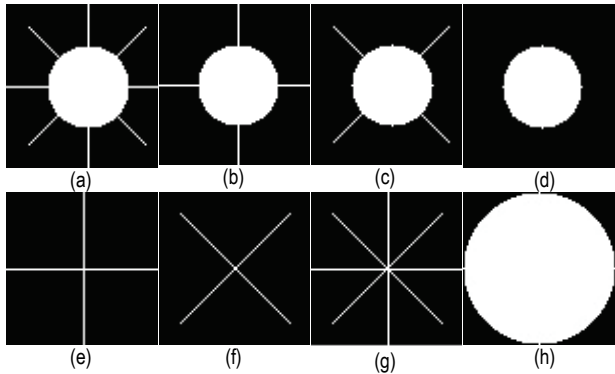


Figure 6 Reduced images

Considering characteristics of the eye, detecting eye position in a windowed manner within an eye region may be unnecessarily time-consuming. Also, as the resolution of the eye region grows, the time needed to compute every window would also grow. In order to further reduce the time consumption, instead of checking all of the windows, the two-stage window movement is tried as seen in Fig. 7. For the starting position, the middle of the  $y$ -axis of the eye region is chosen as a center of the  $y$ -axis of the window. After finding the best  $x$ -axis position for given  $y$ -axis; using best  $x$ -axis position the window is moved in  $y$ -axis to determine best overall position.

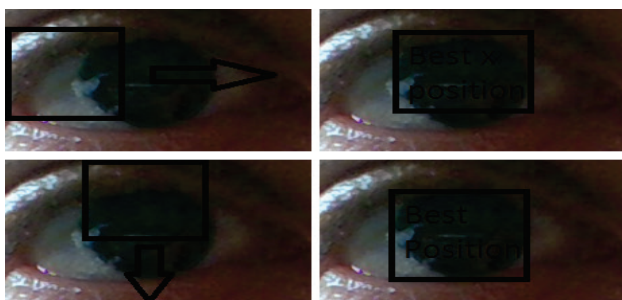


Figure 7 Reduced window movement

## 2.4 Matter of Blink

Blink detection is also a necessity for many applications which use eye detection. If it is known when a person blinks, then there is no need to try to detect eye positions since it is known that there is no eye in the image. Also by calculating eye blink frequency it is possible to determine the tiredness of an individual.

Eyeblink could be easily detected by utilizing landmark points ( $p_2, p_3, p_{11}, p_{12}, p_4, p_5, p_7, p_8$  in Fig. 3) and eye radius which is calculated by the Eq. (2). Eyeblink can be determined by calculating the distance between the

upper and bottom eyelids. If this distance is close to one or two pixels, then it can be assumed that the eye is shut. However, there is also a matter of deciding in which condition it should be deemed there is a blink. Should a moment where the eye is completely shut be chosen or should moments where eyes are partially shut also be added? As seen in Fig. 8, the act of blinking is composed of a few stages. When looking from the perspective of getting information out of images, in a state where eyes are partially shut and in a state where eyes are completely shut, they are the same because both have the same meaning that the eyes are in a state where they do not concentrate on looking. So, if the distance between the eyelids is smaller than half of the eye radius, it can be assumed that the eye is closed. Even though this is a practical approach there are still some negligible issues. When a person laughs or looks at a bright sight, people tend to partially close their eyes which may be interpreted as a blink. It is not easy to detect the difference between a blink and a laugh or a bright light reflex. It is possible to use other landmark points which correspond to mouth to detect the laugh or it may be possible to check the brightness by using a histogram. However, it would be too much effort to detect the difference, and it is an acceptable error. Also, it will not make much difference in practice.

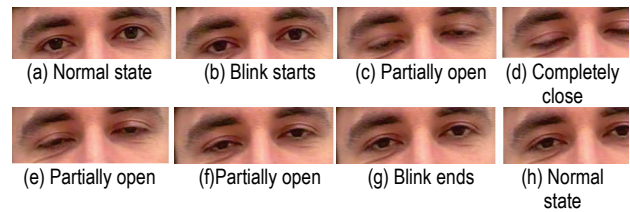


Figure 8 States of the eyes when blinking

## 3 EXPERIMENTAL RESULTS

In the experiments, a computer with Intel Core i7 2.8 GHz CPU and 32 GB ram has been used. The method is coded using python version 2.7. As for the computer vision library, OpenCV version 2.4 has been chosen.

The code takes 4.5 ms (millisecond) to detect both eye centers on images from the BioID database, not including the face detection time.

### 3.1 Databases

BioID database, TalkingFace database, and ColorFeret database are used to evaluate the success rate of the proposed method.

The BioID database consists of 1520 frontal face images with files which contain eye position annotations of the left and right eye. There is a single frontal face at each image, and these images belong to twenty-three different individuals. BioID database is a challenging database because images are gray level and have low resolution ( $384 \times 286$ ). Most of the images belong to the persons who use glasses, and there is an intense reflection in their glasses. There are images taken in areas that are not sufficiently illuminated or over-illuminated. There are images where an individual's eyes are fully closed or partially closed which makes it impossible to determine where eyes are since there is no eye in view. However, to determine the



success of an algorithm a challenging database is necessary. All of the images on the BioID database are used in the experiments.

The TalkingFace database consists of 5000 frontal face images of one individual which are taken while the individual is talking in a sufficiently illuminated area. The images are in color and have  $720 \times 526$  resolution. There is a single frontal face at each image and a file for each image that contains coordinates of 68 interest points of the face which include eye positions. All of the images of TalkingFace are used in the experiments.

ColorFerret database is another face database which contains color images to develop, test and evaluate face detection algorithms. Colorferet database contains 11338 facial images of 994 subjects from various angles. The images in the Color FERET Database are 512 by 768 pixels. Regular front images and alternative frontal images are used in this study. However, some of the images did not have eye position information, and usually, eye positions are not accurate.

Image resolution itself may not be adequate in understanding the database. Image resolution and resolution of the faces within images are given in Tab. 1. Mean resolution of the faces for the BioID database is very low. Resolution of the TalkingFace database is about 3.4 times that of the BioID database. However, resolution of the face area of the TalkingFace database is about 5.4 times that of the BioID database.

**Table 1** Image Resolution and The Face Resolution of Databases

Database	Image Resolution	Maximum Face Resolution	Mean Face Resolution
BioID	384 x 286	223 x 248	125 x 125
TalkingFace	720 x 526	321 x 321	291 x 291

### 3.2 Evaluation

Error rate  $e$  is calculated by using Eq. (10) where  $e$  is an error rate,  $L, R$  are actual position of the left and the right pupil whereas  $L', R'$  are calculated positions by the method and difference of them is the Euclidean distance between them.

$$e = \frac{\max(\|L - L'\|, \|R - R'\|)}{L - R} \quad (10)$$

If error rate  $e$  is less than 0.25, the algorithm is considered good for locating the eye, however, for application such as gaze detection,  $e$  should be less than 0.05.

As seen in the images, this approach manages to create a minimal eye region. Also, these eye regions do not contain eyebrows. As long as eye corners are detected at an acceptable error rate, this eye region creation approach is successful in creating a small region which contains only eye area.

In Fig. 9 and Fig. 10, successful results for BioID and TalkingFace databases are given. As seen in the results this method manages to draw a white circle to the eyes and the center of the circle is where eye pupil is.



**Figure 9** Example of the successful results for BioID database ( $e < 0.025$ )



**Figure 10** Example of the successful result for TalkingFace database ( $e < 0.025$ )



**Figure 10** Example of the unsuccessful result for TalkingFace database ( $e < 0.025$ )



**Figure 11** Example of the unsuccessful result of BioID database ( $e < 0.025$ )

In Fig. 11 and Fig. 12, unsuccessful results for BioID and Talking Face databases are given. In some cases as seen in Fig. 11 and Fig. 12, eye corners are not accurate. In such a case calculated eye radius may be smaller than the

real value. Even though the result will be within the eye, it may not be perfectly aligned with the eye. In some other cases, the small partition of the eyes is visible as seen in the top left image in Fig. 12. In such cases; the eye may not be accurately localized.

In Fig. 13 and Fig. 14 unsuccessful results for BioID and TalkingFace databases are given. However, as seen in the images these results were successful. It has been noticed that some of the eye positions are not annotated accurately.



Figure 12 In some cases eye positions of the BioID database are not accurate. These images are marked as unsuccessful for  $e < 0.05$



Figure 13 Successful localization, which is marked as unsuccessful for Talking Face database ( $e < 0.025$ )

Our attempt to correct the inaccurate eye positions showed that even the same person might mark different locations as an eye in different trials, which is why it has been decided to use given eye positions. However, it should be noted that for ( $e < 0.025$ ) some of the successful results will be marked as unsuccessful.



Figure 14 Example of the successful result of ColorFeret database ( $e < 0.025$ )

ColorFeret database has the best resolution for faces

out of the three databases. However, it has also the poorest eye position information. Usually, they merely did mark somewhere within an eye. They did not mark eye centers. In Fig. 15, some successful results from ColorFeret database are given. It can be seen that as the resolution grows the method can cope with it. Fig. 16 shows yet another example of failed eye corner detection for ColorFeret database. As seen in images, eye corners are not accurate which causes unsuccessful results. However, since eye positions are not accurate, ColorFeret database has not been used for success comparison.



Figure 15 Example of the unsuccessful result of ColorFeret database ( $e < 0.025$ )

Experimental results are given in Tab. 2 and Tab. 3. For  $e < 0.1$  at worst 91.91% success rate is achieved in both databases.

As seen in Fig. 6, (h) is the input list where all points within the eye circle are used. Thus the best result is expected from this list and as seen in Tab. 2 for both databases (h) has the best results. (b), (c), and (d) have the lowest score in both databases, and all of these input lists have the points from a circle which has a half radius of the eye. Usually, reflection occurs within this circle and, we believe, that is the reason for the low success rates.

Table 2 Experimental Results of Reduced Points

Database	Template	$e < 0.025$	$e < 0.05$	$e < 0.1$	$e < 0.25$	Mean Time / ms
BioID	(a)	38.79%	72.71%	93.35%	99.21%	1.8
	(b)	24.98%	55.95%	93.35%	99.21%	1.4
	(c)	25.64%	71.2%	94.01%	99.21%	1.5
	(d)	10.65%	44.37%	92.96%	99.27%	1
	(e)	30.11%	60.15%	93.09%	99.21%	1.1
	(f)	27.67%	74.22%	93.42%	99.21%	0.9
	(g)	41.74%	75.27%	93.16%	99.21%	1.6
	(h)	46.67%	79.75%	91.91%	99.21%	4.3
TalkingFace	(a)	22.1%	62.02%	94.72%	99.6%	25
	(b)	03.52%	34.6%	94.74%	99.6%	20
	(c)	09.3%	51.22%	95.16%	99.58%	22
	(d)	01.48%	32.64%	95.1%	99.58%	17
	(e)	20.24%	67.74%	93.42%	99.6%	9.8
	(f)	68.48%	87.94%	95.06%	99.58%	8.6
	(g)	63.18%	85.56%	94.12%	99.6%	16
	(h)	68.16%	88.92%	93.32%	99.58%	74

(e), (f), and (g) are the lists which have a low point count. In descending order, they are sorted (g), (e), and (f) according to the number of points they contain. The results of the (f) are surprisingly good for TalkingFace database,



and it is better than the (h) for  $e < 0.025$ . However, for BioID databases result of the (f) is the lowest among them. In a state where illumination is sufficient, resolution is adequate as well; (f) is an excellent choice for locating the eye. (e) was not very successful in both databases in contrast to (g) which manages to achieve good results in both databases. It seems (g) is an acceptable choice for an application which contains faces with different resolutions.

**Table 3** Experimental Results For Two Stage Window Movements

Database	Template	$e < 0.025$	$e < 0.05$	$e < 0.1$	$e < 0.25$	Mean Time / ms
BioID	(a)	35.76%	71.59%	93.68%	99.27%	0.5
	(b)	21.69%	54.24%	93.81%	99.27%	0.4
	(c)	24.72%	69.23%	94.08%	99.27%	0.4
	(d)	11.11%	49.57%	94.74%	99.27%	0.3
	(e)	25.04%	57.65%	93.68%	99.27%	0.3
	(f)	27.81%	72.18%	93.09%	99.21%	0.3
	(g)	39.77%	73.57%	93.49%	99.27%	0.5
	(h)	43.52%	78.89%	92.17%	99.21%	0.9
TalkingFace	(a)	23.2%	65.58%	96.44%	99.6%	2.6
	(b)	03.88%	37.28%	96.7%	99.6%	2.3
	(c)	10.46%	55.56%	96.82%	99.6%	2.2
	(d)	01.78%	36.14%	96.72%	99.6%	1.8
	(e)	23.42%	73.48%	96.62%	99.6%	1
	(f)	64.24%	88.72%	96.68%	99.6%	0.9
	(g)	61.32%	87.16%	96.46%	99.6%	1.6
	(h)	68.26%	89.44%	95.3%	99.6%	7.8

### 3.3 Comparison with the State of the Art

This study is compared with state of the art methods which use the BioID database and Talking Face database to determine the success rate. The results are given in Tab. 4.

**Table 4** Comparison With The State Of The Art

Method	$e < 0.025$	$e < 0.05$	$e < 0.1$	$e < 0.25$
BioID				
(h)	46.67%	79.75%	91.91%	99.21%
(g)	41.74%	75.27%	93.16%	99.21%
Fabian and Erdarth [6]	-	82.5%	93.4%	98.0%
Asadifard and Shanbezadeh[10]	-	47.0%	86.0%	96.0%
Valenti and Gevers[8]	55%*	86.1%	91.7%	97.9%
Asteriadis et al.[11]	22.5%*	42%*	81.7%*	97.4%*
Timm and Barth[6]	38%*	82.5%	93.4%	98%
Ahuja et al. [13]	NA	92.06%	97.96%	100%
Markus et al. [17]	61%*	89.9%	97.1%	99.7%
Levinshstein et al.[16]	68.13%	95.07%	99.59%	100%
TalkingFace				
Ahuja et al.	NA	94.78%	99%	99.42%
Levinshstein et al.	65.78%	95.68%	99.88%	99.88%
(h)	68.16%	88.92%	93.32%	99.58%
(g)	63.18%	85.56%	94.12%	99.6%
(f)	68.48%	87.94%	95.06%	99.58%

Easymatch method seems to be average against state of the art for the BioID database. However, it should be noted even though all of the methods use BioID database some of them use the database partially such as removing images with people who use eyeglasses or removing images with closed eyes. Also, some of the studies [6, 8, 10, 11] had used the Viola-Jones face detector to detect the face and according to our tests; successful face detection

rate of the Viola-Jones is 89%, which means the success rate of their method shouldn't be more than 89% if they used the Viola-Jones method.

Easymatch method has the best result in ( $e < 0.025$ ) against state of the art for TalkingFace database. This approach yields better results than low resolutions at decent resolutions. However, this difference is also due to poor environmental conditions of the BioID database.

This approach is faster than other methods. The time consumption is close to or less than the time consumption of image smoothing operations.

Methods which use AI-based approaches achieve better success rates; in fact, they achieve the best success rates overall. However, they also have some shortcomings. For AI-based approaches to work, images must be resized to a certain size, which means additional workload and additional quantization error. If an image has a better resolution, the approaches' success rates tend to drop. For the BioID database, 4 ms average time consumption is calculated in [16] while the approach presented in this paper manages to get less than 1 ms.

### 4 CONCLUSION

Eye detection is a fundamental and essential step for many applications. This paper presents a simple matching method to localize eye positions in frontal face images. Three characteristics of the eyes have been used to determine eye location. Those characteristics are: eyes are darker than the surface around of it, there is a ratio between the eye radius and the distance between two eye corners, and finally, eyes are round. Using facial landmarks and a tailored matching method a novel way to detect the eyes is presented. A face detector with 68 points has been used in experiments. However, out of 68 points, only the 12 points which are around the eye have been used in this study. We believe it would provide better results if an eye detector with 12 points were to be created for eye detection. The approach which is shown in this study provides successful results in a fast and easy way. Different reduced input images have been used to detect eye. Each input image has different computation speed and different success rates, which makes it possible to choose between them according to the expectations of the application.

Overall, the method yields promising results, considering it is an easy approach which does not involve any learning or model scheme. Also, the method does not need any complex algorithm to determine eye location; it is easy to understand and easy to implement.

### 5 REFERENCES

- [1] Daugman, J. (2009). How Iris Recognition Works. In *The Essential Guide to Image Processing*, 715-739. Academic Press. <https://doi.org/10.1016/b978-0-12-374457-9.00025-1>
- [2] Wildes, R. P. (1997). Iris recognition: An emerging biometric technology. *Proceedings of the IEEE*, 85, 1348-1363. <https://doi.org/10.1109/5.628669>
- [3] Nair, et al. (2016). A Survey on Driver Fatigue-Drowsiness Detection System. *International Journal of Engineering and Computer Science*, 5(11), 19237-19240. <https://doi.org/10.18535/ijecs/v5i11.92>
- [4] Bulling, A., Ward, J. A., Gellersen, H., & Troster, G. (2011). Eye Movement Analysis for Activity Recognition Using

- Electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4), 741-753. <https://doi.org/10.1109/TPAMI.2010.86>
- [5] Kenyon, R. V. (1985). A soft contact lens search coil for measuring eye movements. *Vision Research*, 25(11), 1629-1633. [https://doi.org/10.1016/0042-6989\(85\)90133-6](https://doi.org/10.1016/0042-6989(85)90133-6)
- [6] Rimmel, S. R. (1984). An Inexpensive Eye Movement Monitor Using the Scleral Search Coil Technique. *EEE Transactions on Biomedical Engineering, BME*, 31(4), 388-390. <https://doi.org/10.1109/TBME.1984.325352>
- [7] Timm, F. & Erdhardt, B. (2011). Accurate Eye Centre Localisation by Means of Gradients. *Sixth International Conference on Computer Vision Theory and Applications*. Retrieved from <https://www.inb.uniluebeck.de/fileadmin/files/PUBPDFS/TiBa11b.pdf>
- [8] Türkan, M., Pardas, M., & Çetin, A. E. (2007). Human eye localization using edge projections. Proceedings: *Second International Conference on Computer Vision Theory and Applications*, 410-415.
- [9] Valenti, R. & Gevers, T. (2012). Accurate eye center location through invariant isocentric patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9), 1785-1798. <https://doi.org/10.1109/TPAMI.2011.251>
- [10] Kroon, B., Hanjalic, A., & Maas, M. S. (2008). Eye localization for face matching: Is it always useful and under what conditions? *Proceedings of the 7<sup>th</sup> ACM International Conference on Image and Video Retrieval*, 379-388. Ontario, Canada. <https://doi.org/10.1145/1386352.1386401>
- [11] Asadifard, M. & Shanbezadeh, J. (2010). Automatic Adaptive Center of Pupil Detection Using Face Detection and CDF Analysis. *International MultiConference of Engineers and Computer Scientists*. Hong Kong. Retrieved from [https://www.researchgate.net/publication/44260567\\_Automatic\\_Adaptive\\_Center\\_of\\_Pupil\\_Detection\\_Using\\_Face\\_Detection\\_and\\_CDF\\_Analysis](https://www.researchgate.net/publication/44260567_Automatic_Adaptive_Center_of_Pupil_Detection_Using_Face_Detection_and_CDF_Analysis)
- [12] Asteriadis, et al. (2006). An eye detection algorithm using pixel to edge information. *2<sup>nd</sup> International Symposium on Communications, Control, and Signal Processing*. Marrakech.
- [13] Ahuja, K., Banerjee, R., Nagar, S., Dey, K., & Barbhuiya, F. (2016). Eye center localization and detection using radial mapping. *IEEE International Conference on Image Processing*, 3121-3125. <https://doi.org/10.1109/ICIP.2016.7532934>
- [14] George, A. & Routray, A. (2016). Fast and accurate algorithm for eye localisation for gaze tracking in low-resolution images. *IET Computer Vision*, 10(7), 660-669. <https://doi.org/10.1049/iet-cvi.2015.0316>
- [15] Villanueva, A., Ponz, V., Sesma-Sanchez, L., Ariz, M., Porta, S., & Cabeza, R. (2013). Hybrid method based on topography for robust detection of iris center and eye corners. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 9(3), Article No: 25. <https://doi.org/10.1145/2501643.2501647>
- [16] Levinshtein, A., Phung, E., & Aarabi, P. (2018). Hybrid eye center localization using cascaded regression and hand-crafted model fitting. *Image and Vision Computing*, 71, 17-24. <https://doi.org/10.1016/j.imavis.2018.01.003>
- [17] Markuš, N., Frljak, M., Pandžić, I. S., Ahlberg, J., & Forchheimer, R. (2014). Eye pupil localization with an ensemble of randomized trees. *Pattern Recognition*, 47(2), 578-587. <https://doi.org/10.1016/j.patcog.2013.08.008>
- [18] Hansen, D. W. & Ji, Q. (2010). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3), 478-500. <https://doi.org/10.1109/TPAMI.2009.30>
- [19] Viola, P. & Jones, J. M. (2004). Robust Real-Time Face Detection. *International journal of Computer Vision*, 57(2), 137-154. Retrieved from <https://link.springer.com/article/10.1023/B:VISI.0000013087.49260.fb>
- [20] Zafeiriou, S., Zhang, C., & Zhang, Z. (2015). A survey on face detection in the wild: Past, present and future. *Computer Vision and Image Understanding*, 138, 1-24. <https://doi.org/10.1016/j.cviu.2015.03.015>
- [21] <http://www.opencv.org>
- [22] Ouanan, H., Ouanan, M., & Aksasse, B. (2016). Facial landmark localization: Past, present and future. *4th IEEE International Colloquium on Information Science and Technology*. <https://doi.org/10.1109/CIST.2016.7805097>
- [23] [www.dlib.net](http://www.dlib.net)
- [24] Kazemi, V. & Sullivan, J. (2014). One millisecond Face Alignment with an Ensemble of Regression Trees. *Conference on Computer Vision and Pattern Recognition 2014 IEEE*. Columbus, OH, USA. <https://doi.org/10.1109/CVPR.2014.241>
- [25] <https://ibug.doc.ic.ac.uk/resources/facial-point-annotations>
- [26] <https://www.bioid.com/About/BioID-Face-Database>
- [27] [http://www.prima.inrialpes.fr/FGnet/data/01TalkingFace/talking\\_face.html](http://www.prima.inrialpes.fr/FGnet/data/01TalkingFace/talking_face.html)

**Contact information:****Muhammed Emre ÇOLAK**

(Corresponding author)  
Firat University, Faculty of Engineering,  
Department of Software Engineering,  
23119 Elazig/Turkey  
E-mail: memrecolak@firat.edu.tr

**Asaf VAROL**

Firat University, Faculty of Technology,  
Department of Software Engineering,  
23119 Elazig/Turkey  
E-mail: avarol@firat.edu.tr