Electronic Letters on Computer Vision and Image Analysis 16(1):15-32; 2017

Robust Real-Time Gradient-based Eye Detection Using Transform Domain and PSO-Based Feature Selection

Nasrin Salehi, Maryam Keyvanara and Amirhassan Monadjemi

Department of Artificial Intelligence, University of Isfahan ,Isfahan, Iran

Received 28th Sep 2015; Accepted 2nd Mar 2017

Abstract

Despite numerous research on eye detection, this field of study remains challenging due to the individuality of eyes, occlusion, and variability in scale, location, and light conditions. This paper combines a technique of feature extraction and a feature selection method to achieve a significant increase in eye recognition. Subspace methods may improve detection efficiency and accuracy of eye centers detection using dimensionality reduction. In this study, HoG (histogram of Oriented Gradient) descriptor is used to lay the ground for Binary Particle Swarm Optimization (BPSO) based feature selection. HoG features are used for efficient extraction of pose, translation and illumination invariant features. HoG descriptors uses the fact that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The method upholds invariance to geometric and photometric transformations. The performance of presented method is evaluated using several benchmark datasets, namely, BioID and RS-DMV. Experimental results obtained by applying the proposed algorithm on BioID dataset show that the proposed system outperforms other eye recognition systems. A significant increase in the recognition rate is achieved when using the combination of HoG descriptor, BPSO, and SVM for feature extraction, feature selection and training phase respectively. The Recognition rate for BioID dataset was 99.6% and the detection time was 15.24 msec for every single frame.

Key Words: Eye detection, HoG descriptor, BPSO feature selection, SVM classifier.

1. Introduction

Eyes are one of the most important elements of human face and can be considered a factor of discrimination between individuals. Eyes can express a person's identity, personality, emotional states and even interpersonal relations. In many face detection methods, eyes play a crucial role. The unique geometric, photometric, and motional characteristics of the eyes provide important visual signs for face detection, face recognition, and understanding facial expressions. For example, one of the primary stages in the Viola and Jones face detector

Correspondence to: maryam.k1ara@gmail.com

Recommended for acceptance by João Manuel Tavares
http://dx.doi.org/10.5565/rev/elcvia.811
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona,
Barcelona, Spain

is a Haar feature corresponding to the eye region [1]. There are several criteria for discrimination of the human eye such as the color of the iris, size of the pupil and shape of the eye [2].

Also, vision based recognition and tracking systems have gained a lot of attention in the recent years. In these systems, face detection and eye tracking has emerged as a remarkable research area with a diverse set of applications including human computer interaction, mobile interfaces, psychological disorder diagnosis, neurology and ophthalmology, assistive systems for drivers or disabled people, and biometrics [3-11].

A significant amount of work has been done on eye detection and tracking. Yang et al. used Gabor wavelet transform along with Hidden Markov Model for eye statement recognition [3]. Also Daud et al. used a neural network for recognition of eye patterns [2]. Moreover, other pattern recognition techniques are used for gaze tracking. In [12] a fuzzy template matching has been proposed which is constructed based on the piecewise boundary to judge the similarity between the input image and the eye template. In the template, the eyelid is constructed based on adjacent segments along the piecewise boundary regions. Also five feature parameters which express the geometry of the eye to measure similarity of the input image to eye template are determined.

Particle swarm optimization (PSO) algorithm can be used for object detection and specifically for eye detection and tracking, because of its effectiveness in heuristic search and relatively small computational time required [13]. This algorithm was originally introduced by Eberhart and Kennedy [13], though many adjustments have been made to the basic PSO algorithm over the past decade. Some have resulted in improved general performance, and some have improved performance on particular kinds of problems. Al-Mamun et al. proposed an approach for eye detection which exploits the flexibility of deformable template and uses genetic algorithm to match the template for eye detection by estimating the coordinates of the template location, scaling factor and rotation angle [14]. It utilizes Particle Swarm Optimization to select efficient features to obtain an efficient processing time. PSOAdaBoost algorithm is used for face and eye detection to select the best features and achieve better performance in terms of a small training time and better classification accuracy [15]. Cheng-Jian Lin et al. [16] used Gabor wavelet extracting local features of face feature and PCA to obtain facial texture and surface features vector from the gray images. But employ a hybrid Taguchi particle swarm optimization (HTPSO) algorithm for face recognition based on multilayer neural networks. Sharma and Singh [17] introduced an algorithm for object detection using PSO. Also, surface points can be used as a basis for classification according to a set of classification rules, which are discovered by an Ant Colony Optimization (ACO)/ PSO-based rule discovery algorithm. For 3D human body pose estimation, PSO can be used so that each particle position vector in the search space is a plausible skeleton configuration [18]. Likewise, a binary PSO algorithm with mutation operator for feature selection using decision tree is applied in a spam detection research [19]. Binary PSO was firstly introduced and used by Kennedy and Eberhart [20]. They described a very simple alteration of the canonical algorithm that operates on bit-strings rather than real numbers.

Several other researchers have made some changes to the basic particle swarm algorithm to allow it to operate on binary spaces. The location of the particles in each generation can be used as probabilities for selecting features in a pattern-matching task [21]. In this approach, each feature is assigned a slice of a roulette wheel based on its floating-point value, which was then discretized to one of the values is the set {0, 1}, indicating whether the feature was selected or not. In another work, Mohan and Al-Kazemi [22] suggested several ways that the particle swarm could be implemented on binary spaces. For example, “regulated discrete particle swarm,” method performed very well on a suite of test problems. Also, Pamparä et al [23] developed a method in which each particle stored the small number of coefficients of a trigonometric model instead of directly encoding bitstrings in the particles.

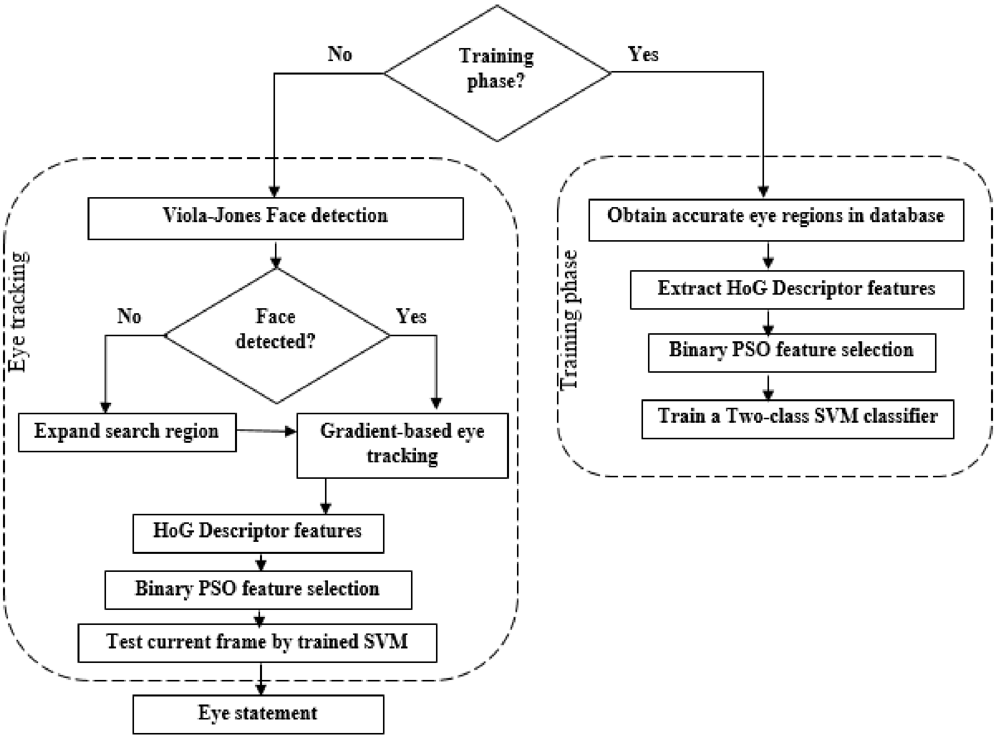

Training phase?
No
Yes
Eye tracking
Viola-Jones Face detection
Face detected?
No
Yes
Expand search region
Gradient-based eye tracking
HoG Descriptor features
Binary PSO feature selection
Test current frame by trained SVM
Eye statement
Training phase
Obtain accurate eye regions in database
Extract HoG Descriptor features
Binary PSO feature selection
Train a Two-class SVM classifier

Fig. 1. The workflow of the proposed method for eye tracking and driver fatigue system in this paper.

In the recent years, histogram of oriented gradient (HoG) has been successfully applied in many research fields especially in object detection algorithms for feature extraction [24-28]. For instance, in the recognition framework, dynamic dense grid-based HoG features are used to extract the appearance features by accumulating the gradient magnitudes for a set of orientations in 1D histograms defined over a size-adaptive dense grid [24]. HoG descriptor extraction is also used as a fast and accurate method for pedestrian recognition [29]. In [30], once the HoG features are extracted, PCA is applied to the feature vectors. Then a linear SVM (support vector machine) classifier is used to determine pedestrians.

Nonetheless, HoG descriptor has also been used for eye detection. In [25], at first the eye locations are preselected by using a couple of AdaBoost classifiers trained with Haar-like features and then, HoG descriptors are used for feature selection in a SVM classifier to obtain the best pair of eyes among all possible combinations of preselected eyes. In [28], fusion of HoG descriptors at different scales, are used to capture important structure for face recognition. In this work, in order to avoid errors in facial feature detection due to illumination, occlusions and pose changes, HOG descriptors were extracted from a regular grid. Fusion of HOG descriptors at different scales allows to capture important structure for face recognition. In addition, they offered a method for dimensionality reduction to remove noise and make the classification process less prone to overfitting.

In this work, we present a fully novel scheme for a robust real-time eye detection system which can be used for driver fatigue detection. In this method we combined methods: 1) HoG feature extraction, 2) Binary PSO for feature selection and SVM for training the most significant selected features to achieve an improvement in detection accuracy and detection time. The eye detection method is evaluated using BioID dataset. Also we evaluated the performance of our algorithm as a driver fatigue detection system by using RobeSafe Driver Monitoring video dataset.

2. Methodology

In this study, we present a novel method for real time eye detection and eye statement recognition. We divide the proposed algorithm to training and detection phases.

In training stage the algorithm tries to learn the most significant features of the eye regions by extracting HoG features followed by binary Particle Swarm Optimization (PSO) feature selection to select the most effective subset of features extracted in previous step. Selected HoG features are used as Support Vector Machine input. We use a two-class SVM classifier to learn the selected features. The SVM determines whether each region is an open eye or not. In the first step of the detection algorithm the face region is detected using Viola and Jones's algorithm. The algorithm then estimates eye centers by using image gradients. In the next step we rescale the detected face region to a fixed size. Then the eye pair candidates are extracted by a fixed size window centered at the detected eye centers. Finally HoG features are extracted for eye pairs regions, and the most significant features are selected and sent to the SVM classifier to determine eye statement. In some cases it is possible that the face detection fails in the previous step. For example, rotated or occluded face region may cause the Viola-Jones algorithm to fail in searching for the face. In these frames, we straightly detect candidate eye regions with image gradiants and then detect the true eye regions with HoG and support vector machine by expanding the search region of the consecutive frame. The complete workflow is presented in Fig. 1.

2.1. *Face Detection*

This study employs the Viola-Jones algorithm [31] to detect possible face regions. The Viola-Jones algorithm uses the scalar product of the input image and some Haar-like templates, which is known as Haar-like features. So if we denote *I* and *P* (both of the same size N×N) to be an image and an arbitrary pattern respectively, the features associated with pattern *P* of image *I* is calculated as Eq. (1) (an example is given in Fig. 2):

$$\sum_{1\le i\le N}\sum_{1\le j\le N} \mathrm{I(i,j)}1_{P(i,j) is\ white} - \sum_{1\le i\le N}\sum_{1\le j\le N} \mathrm{I(i,j)}1_{P(i,j) is\ black} \tag{1}$$

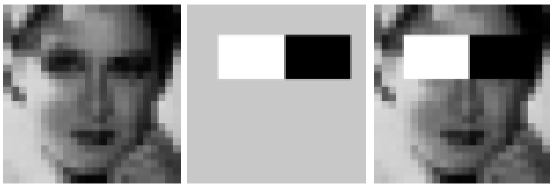


Fig. 2. Haar-like features. (a) original face region image (b) haar-like template (c) The result haar-like features as a product of haar-like templates and input face image [31].

Different lighting conditions can cause the extracted features to be unreliable. So, the mean and variance of all the images are normalized first. Moreover those images having little information of interest with a variance lower than 1, are left out of consideration. [31].

The Viola-Jones algorithm [32] works on the basis of integral images, cascade detectors and also the adaboost algorithm. The Adaboost is used for selecting the extracted features. By having the features, an observation is mapped to a label by the classifier. The label has the value in a finite set of $\{-1,1\}$, where 1 denotes the detection of a face and on the other hand -1 is when there is no face. The number of Haar-like features extracted from an image is determined by the parameter d described in reference [31].

A classifier needs to be able to map an observation to a label valued in a finite set. For example, for face detection, we can consider the function $f\colon \mathbb{R}^d \to \{-1, 1\}$, where 1 means that there is a face and -1 the contrary. Here, parameter d is the number of Haar-like features extracted from each given image. The Adaboost classifier tries to reduce an upper bound of the empirical loss iteratively by assigning the probabilistic weights $\omega \in \mathbb{R}_+$ to a training set which is made up of n observation label pairs referred as (p_{1i}, p_{2i}). The empirical loss is defined as bellow [31, 32]:

$$\sum_{i=1}^{n} \omega_i \, 1_{p_{2i} \neq f(p_{1i})} \tag{2}$$

2.2. *Eye Localization Using Image Gradients*

For eye detection, it is assumed that face region is detected in the previous step. If face detection has failed, the search region for eye detection expands to a larger region according to image size and the detected face size in previous frame. This speeds up the algorithm and decreases the computation time.

In this stage, first we detect candidate eye regions using image gradients. Then HOG features are extracted for each region, the most effective features are selected using particle swarm optimization and then, the SVM classifier determines whether the selected regions are open eyes or not.

So, in the first step, a gradient based method for eye localization would be useful for finding possible eye regions in each input frame. This method seems useful since it is invariant to resolution, contrast, scale and lighting conditions [33].

In this method, after the face has been detected, a rough region for each eye is found and used for eye region estimation. For each eye, an objective mathematical function is used. This function reaches its maximum when the center of the eye is localized. This is the intersection point of most gradient vectors. In fact, the square dot product between the displacement vector of a center candidate and the image gradient is calculated for every single pixel. Among these calculations, the position of the maximum is the one corresponding to the position where most image gradients intersecting. This method is efficient in time complexity and accurate in localizing the position of the eyes.

In cases where the maximum of the objective function is not found properly, that is to say that the reached maxima leads to incorrect centre estimates, prior knowledge about the eye should be used. In the detected eye region, the eye pupil is darker that other areas. Hence, a weight for every possible centre is applied so that dark centres have a bigger weight compared to lighter centres and so detected as the maxima. if we are given the pixel positions $x_i, i \epsilon \{1, \ldots, N\}$, the optimal centre of the eye c^* in an image can be obtained by as the following equations:

$$c^* = argmax_c \frac{1}{N}\sum_{i=1}^{N} w_c(d_i^T g_i)^2 \tag{3}$$

$$d_i = \frac{x_i - c}{\|x_i - c\|} \qquad , \forall\, i : \ \|g_i\|_2 = 1 \tag{4}$$

In Eq. (3) d_i are the displacement vectors, and are scaled to unit length so that all the pixel positions will be of an equal weight. Vector g_i is gradiant vector which should have the same orientation as the displacement vector d_i. The vectors d_i and g_i alongwith the points x_i and c are illustrated for more clarity in Fig. 3. The parameter, w_c=I*(c_x c_y) is the grey value at point (c_x, c_y) of the input image I* which is smoothed by a Gaussian filter and also inverted. The function $\sum_{i=1}^{N} w_c(d_i^T g_i)^2$ in this equation can be considered as the objective mathematical function. By using prior information in finding the maxima, more accurate results are reached and eye centres are localized precisely [33].

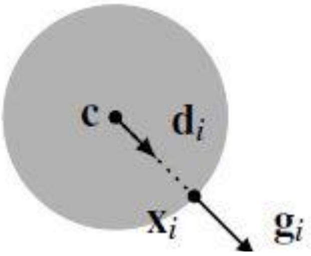


Fig. 3. The illustration of the vectors $\boldsymbol{d_i}$ and $\boldsymbol{g_i}$ alongwith the points $\boldsymbol{x_i}$ and $\boldsymbol{c}$. Note that the the displaceent vector di and the gradient vector gi have the same direction.

2.3. *Histogram of Oriented Gradients*

Histogram of Oriented Gradient (HoG) is an effective feature extraction method. It is in fact a greyscale image feature formed by a set of normalized gradient histograms [29]. The HoG descriptor has performed well in many object recognition problems including pedestrian recognition, human detection applications, body part detection and smile recognition [29, 34-37].

The main idea in HoG descriptors is that the shape and appearance of an object in the image can be defined using edge directions or distribution of intensity gradients. The implementation of these descriptors can be accomplished by dividing the image into small joined parts. These parts are called cells and can be either rectangle or radial. For each cell, a histogram of gradient directions or edge orientations for the pixels inside the cell should be accumulated [37, 38]. The combination of these histograms represents the descriptor. We can then normalize the contrast of these local histograms through the calculation of the intensity across a wider area of the image. This larger region is called a block. This value is then used to normalize all cells within the block. This will increase the accuracy of the operations as the normalization results in improved invariance to changes in illumination and have robustness for changes in appearance and position [30, 37]. HoG descriptor operates on localized cells and hence it upholds invariance to geometric transformations, excluding object orientation [37].

The first step of HoG feature calculation, is the computation of the gradient values. Before that, it should be ensured that grayscale values are normalized. A one dimensional discrete derivative mask is calculated in both horizontal and vertical directions. These masks are applied to calculate the gradient values. Specifically, this method filters the image with the filter kernels $[-1\ 0\ 1]$ and $[-1\ 0\ 1]^T$.

The next step is creating the cell histograms with channels that are evenly spread in the range of [0, 180]. Cell pixels are a weighted vote histogram obtained from the values found in the gradient computation. The histogram channels can also be in the range of [0, 360], if the gradient is signed. In this paper we used the range of [0, 180] to create cell histograms.

As noted before, to avoid changes in illumination and contrast, calculated gradient values are locally normalized. To do this, the cells are grouped together into larger, spatially connected blocks and then each block is normalized by a normalization factor [37].

2.4. *PSO Based Feature Selection*

For feature selection we use binary Particle Swarm Optimization (PSO) to select the most effective subset of features extracted in previous step.

In the recent years, many different combinations of the PSO algorithm have been suggested for feature selection. These versions of PSO include Binary PSO, multi-objective PSO and distributed PSO-SVM hybrid algorithm [39-42]. In [42] a Binary PSO is used for feature subset selection with a fuzzy fitness function. Chuang et al use binary PSO for feature selection using gene expression data [39].

In the cooperative optimization approaches, PSO is a population-based evolutionary computational algorithm [43]. In PSO algorithm there is only one swarm that contains a number of particles. Each particle in the swarm has two properties: position and velocity. These parameters are dynamically adjusted according to their experience, local best known position, and global best position in the swarm. For discrete problems Binary PSO (BPSO) is applied. In binary PSO, each position takes only values of 0 and 1.

As each particle i moves in the search space, its current position is represented by a vector $q_i = (q_{i1}\ q_{i2}\ \dots\ q_{id})$, where d is the dimension of the solution space. Another vector $v_i = (v_{i1}\ v_{i2}\ \dots\ v_{id})$ is used to determine particle velocity. The velocity should be limited in the range $[-v_{max}, v_{max}]$. For each particle, the best position thus far is set as *pbest* and the best position of the whole population which is the global best is called *gbest*. By having the local and global best positions for each particle, the search for optimal solution is continued [43].

Each particle's position and velocity in each iteration is updated according to Eq. (10) and (11) as below:

$$q_{id}^{t+1} = q_{id}^{t} + v_{id}^{t+1} \tag{10}$$

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{1i} * (p_{id} - q_{id}^{t}) + c_2 * r_{2i} * (p_{gd} - q_{id}^{t}) \tag{11}$$

Where, parameter t is the iteration step, d is the dimension of the search space and w is inertia weight which is employed to control the impact of the previous velocities on the current velocity. Constants c_1 and c_2 are acceleration rate parameters. The local and global best parameters are referred as p_{id} and p_{gd} accordingly. Since we are using binary PSO, the values of q_{id}, p_{id} and p_{gd} are limited to 0 and 1. The parameter r_{1i} and r_{2i} are the magnitude of the forces in the direction of local best and global best and are often called acceleration coefficients. According to the definition of the basic particle swarm optimization algorithm [13], these forces are selected to be random. By changing parameters r_{1i} and r_{2i} one can make the PSO more or less "responsive" and possibly even unstable, with particle speeds increasing without control. So these parameters need to be controlled by limiting their value in a small interval. In this paper, the values of r_{1i} and r_{2i} are randomly selected in the uniform distribution of [0,1]. The velocity specifies the equivalent component in the position vector having a value of 1. In order to transform v_{id} in the [0,1] range, a sigmoid function is used [20, 42]. In this method, the position of each particle is then updated as bellow:

$$q_{id} = \begin{cases} 1 & if\ rand < s(v_{id}) \\ 0 & otherwise \end{cases} \tag{12}$$

$$s(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \tag{13}$$

As stated before, for the position vector, a random number should be selected from a uniform distribution in the range $[0, 1]$. This is defined in Eq. (12) as the *rand* parameter.

In the implemented BPSO in this study, the best half of the population are considered in each iteration to be selected for the next generation. The acceleration coefficients, c_1 and c_2, were set to value 1.87. The value of w, the inertia weight, is selected between 0.45 to 0.8. The maximum velocity, v_{max} is set to 4. The w parameter plays an important role in the BPSO algorithm. The values selected for w have been precisely set so that it properly balances its local and global search. The swarm size was selected 60. The algorithm was run for $T = 500$ iterations.

2.5. *Support Vector Machine Classifier*

Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression. SVM is a machine learning technique based on statistical theory which can convert any local solution to a global optimum [44].

Consider a two-class classification linear problem of the form

$$f(z) = w^T \varphi(z) + b \tag{14}$$

Where $\varphi(z)$ denotes a fixed feature-space transformation known as kernel function. Also, parameter w denotes for the weight vector associated with each input vector z. The bias parameter b appeared in Eq. (14) is explicit. The training data set for this problem consists of N input vectors $z_1, \ldots, z_N$, which are assigned to target values $k_1, \ldots, k_{\mathrm{N}}$ where k_i can be assigned to one of the values -1 and 1. Each new input vector z can be classified according to the sign of $f(z)$ [44]. In this context, each vector z_i can be an input image that we are going to determine whether it is eye or not, and the function $f(z)$ can make a decision for the class of input vector.

For linearly separable in feature spaces, at least one choice of the parameters w and b exists such that a function of the form Eq. (14) satisfies $f(z_n) > 0$ for the input vectors with k_n = +1. Also there exists at least one choice of these parameters such that $f(z_n) < 0$ for input vectors with k_n = -1. We can conclude that the term $k_n \mathrm{f}(z_n)$ is none-zero and positive for all training data points. In this case the SVM method solves for

$$\underset{w,b}{\arg\min} \parallel w \parallel^2 \tag{15}$$

$$\text{With } \mathrm{k}(w^T \varphi(z) + b) \geq 1 \qquad n = 1, \ldots, N. \tag{16}$$

Using the Lagrangian function described in Eq. (17).

$$L(w, b, a) = \frac{1}{2}(\parallel w \parallel^2) - \sum_{n=1}^{N} a_n \{k_n (w^T \varphi(z) + b) - 1\}, \tag{17}$$

In order to solve this constrained optimization problem, we use Lagrange multipliers $a_n > 0$, with one multiplier $a_n$for each of the constraints in Eq. (16). Where a = $(a_1, \ldots, a_n)^T$. Note the minus sign in front of the Lagrange multiplier term, because we are minimizing with respect to w and b, and maximizing with respect to a [44].

Having the trained model, in order to classify new input vectors, the sign of $f(z)$ defined in Eq. (14) is determined and the input vector is assigned to one of the existing classes.

Among other classifiers like Neural Network, Decision Trees, Naïve Bayes and k-Nearest Neighbor, the Support Vector Machine has been known as a powerful classifier. If we consider the model with the highest classification accuracy as the best model, SVM has shown a desirable result in detecting objects compared to other classifiers [45].

3. Experimental Results

Several experiments has been done on two publically available datasets BIOID [41] and RS-DMV [47] to evaluate the performance of proposed eye detection algorithm.

The first experiment has done on BIOID dataset. This dataset features a large variety of illumination, background, and scale. BIOID dataset consists of 1521 gray level images with a resolution of 384×286 pixel. Each image is a frontal view of one person out of 23 different test persons [46]. The set also contains manually determined eye positions which allows us to compare our results to manually determined eye centers.

The second experiment has done on RS-DMV dataset. The RS-DMV dataset is a set of video sequences of drivers, recorded with cameras installed over the dashboard. The dataset contains 10 video sequences recorded indoors (3 videos) and outdoors (7 videos). The sequences contain occlusions, illumination changes and other elements that are recognized as computer vision difficulties in face tracking and driver monitoring systems. Size of outdoor video frames is 960x480 pixels, and 1390x480 for indoor videos. Frames are captured at 30 frames per second, and in grayscale format for both indoor and outdoor videos. Also faces in the videos have been marked with 20 points with the same pattern as the BioID database [47]. In this paper we use only 7 outdoor video sequences to evaluate the performance of our algorithm.

To reduce processing time, before eye detection step, face region is detected using Viola and Jones face detector and Haar features. If face detection is successful, we continue to detect eye centers inside possible eye region which is extracted according to anthropometric relations and the size of detected face. Otherwise the search region for eye detection expands to a larger region according to image size and the size of detected face in previous frame. Fig. 4 shows a sample detected face region and the corresponding left and right eye regions.

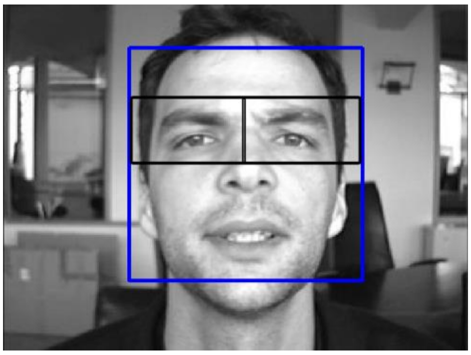


Fig. 4. Extracted regions for applying eye center detection, using detected face region and anthropometric relations.

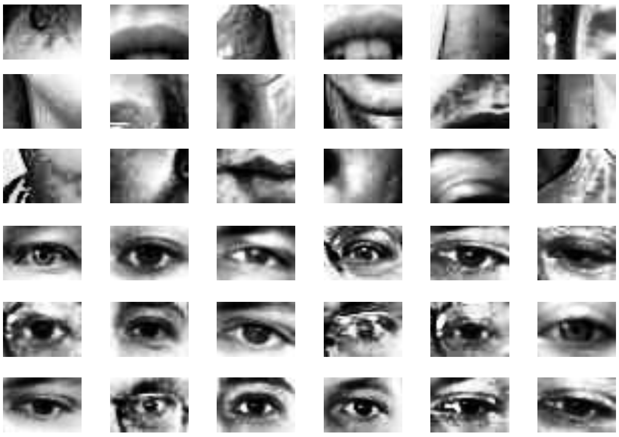


Fig. 5. Illustration of some negative (top three rows) and positive (bottom three rows) samples used for training.

Then gradient based eye detection is applied in each eye region to detect left and right eye centers individually.

After the eye centers localized accurately, we used a trained SVM classifier to determine whether the detected point is really an eye center or not. Extracted HoG features are too complex to be linearly-separatable. So, in this paper, a non-linear polynomial kernel function with a degree of 2 is used for SVM classifier. We applied several degrees to the SVM kernel function to see which degree of the kernel can describe the available data more accurately. In training procedure, the number of selected features which are passed to BPSO algorithm after applying PCA can significantly affect the detection results. In this paper we select 270 features for being tested by SVM classifier. This number of features is selected experimentally by testing numerous numbers and selecting the one which makes the algorithm perform better. The BioID dataset is used to train the SVM classifier in this paper. We manually eliminated closed eyes from extracted eye regions. This gave us a total of 2969 left and right open eye. Also, 1472 regions were randomly selected as none eye regions to fulfill our need to train the classifier. In Fig. 5 some positive (top three rows) and negative (bottom three rows) samples used for training is illustrated.

Table. 1. Comparative performance of various feature selection and feature extraction methods on BioID database.

Method	Recognition Rate (%)	Time/ Image (m sec)
PCA+SVM	91.4	12.68
HoG+SVM	99.8	18.43
HoG+PSO+SVM	99.6	15.24

3.1. BioID Database Results

To evaluate performance of proposed scheme used for eye detection in this paper, we compare it with two different combination of feature extraction and selection methods using BioId database. Table 1 gives the comparative performance using various feature extraction and selection combination schemes. The recognition rate is calculated using the percentage of true positive detections over BioId dataset. Since HOG features characterize different aspects of eye patches including scale, local texture, global shape, and local shape, it would be a promising alternative for extracting essential eye features in comparison to PCA features. Also, this table shows that using PSO as a feature selection method reduces the accuracy of the detection procedure, while it speeds up the detection since the number of the features is significantly reduced. Fig. 6 shows eye detection and eye center localization results. Also Fig. 7 illustrates the cases when eyes are closed and eye detection algorithm should lose the track of eye centers since it is a prerequisite for a driver fatigue system.

To compare eye detection algorithm with state of the art methods that have evaluated using BioID dataset, we need to use an accuracy measure. In this article we use normalized error [48] defined as:

$$e \leq \frac{1}{d}\max(e_1\,, e_2) \tag{18}$$

Here, d is the distance between detected eye centers and e_1 and e_2 are Euclidean distances between reference and detected left and right eye centers respectively.

Table 2 shows the comparison between the method used in this article for eye detection and state of the art methods. The comparison is done using two values of normalized error. Analyzes for evaluation of detection performance is done based on these values of normalized error. In Table 3, three values for e and the corresponding meaning of each value in human eye distances is stated.

From Table 3, we can conclude that a reasonable eye detection method must acquire high results for $e \leq 0.25$. Also an accurate eye detection approach should have high performance for $e \leq 0.05$ [33]. So, results in table 2 shows that the eye detection approach used in this paper achieves very reasonable performance for $e \leq 0.25$ and $e \leq 0.10$ (Iris detection).

When accounting time performance, our algorithm can perform more efficiently in real time compared to state of the art algorithms. For example, [53] is more precise in $e \leq 0.25$ and also $e \leq 0.10$ but the proposal achieves similar results in real time. Also [54] has better results in $e \leq 0.25$ but the executing time of 3 seconds for a single frame is not reasonable for a real time eye detection and tracking algorithm. Proposed method in [56] has similar executing time, however it cannot detect eyes as accurate as the proposed algorithm in this paper.

Table 2. Comparison of eye detection methods based on normalized error, evaluated on BioID database.

	Method	$e \leq 0.10$	$e \leq 0.25$	Time (s)
BioID dataset	Nanni et al. [49]	**99.3%**	**-**	**-**
	Li et al. [50]	-	95.2%	-
	Yi et al. [51]	99.0%	**100.0%**	-
	Asadifard and Shanbezadeh [52]	86.0%	96.0%	-
	Kroon et al. [53]	97.9%	99.9%	0.95
	Campadelli et al. [54]	93.2%	99.3%	3
	Kroon et al. [55]	87.0%	98.8%	-
	Valenti and Gevers [56]	90.9%	98.5%	0.01
	Turkan et al. [57]	73.7%	99.6%	**-**
	Niu et al. [58]	93.0%	97.0%	-
	Chen et al. [59]	89.7%	95.7%	-
	Cristinacce et al. [60]	96.0%	97.1%	-
	Behnke et al. [61]	86.0%	98.0%	-
	Used method	**93.4%**	**98.0%**	**0.01524**
RS-DMV dataset	**Used method**	**92.1%**	**96.9%**	**-**

3.2. *Video Sequences Results*

Fig. 8 illustrates eye detection results from RS-DMV video sequences. Each line in Fig. 8 shows eye detection results from one out of the seven outdoor video sequences. Fig. 9 demonstrates cases when eye center localization algorithm fails to detect eye centers. These cases usually occur when face is rotated or eye are closed, which can be considered as driver drowsiness in driving video sequences.

Table 2 shows the normalized errors for RS-SMV dataset. The results show that the proposed algorithm can accurately track eye centers in this dataset.

It worth mentioning that all the experiments were performed on a 1.7 GHz processor with 6 GB of RAM

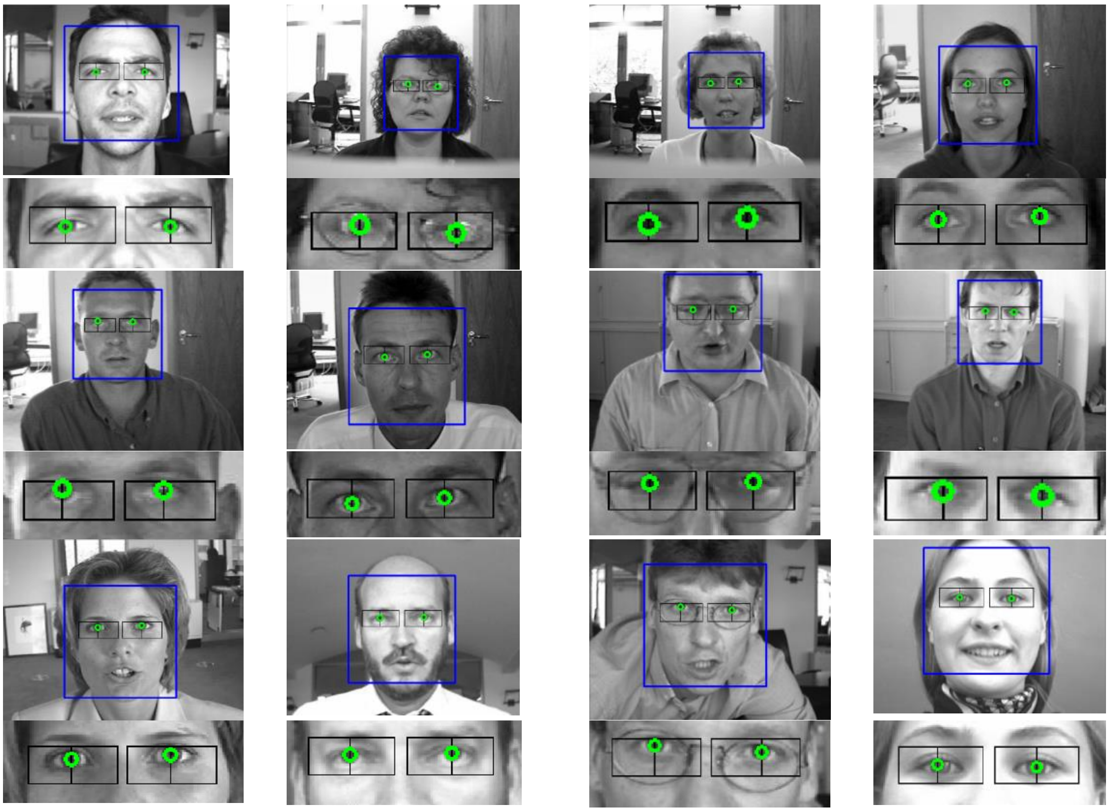


Fig. 6. Illustration of eye detection and eye center localization results on BioID dataset.

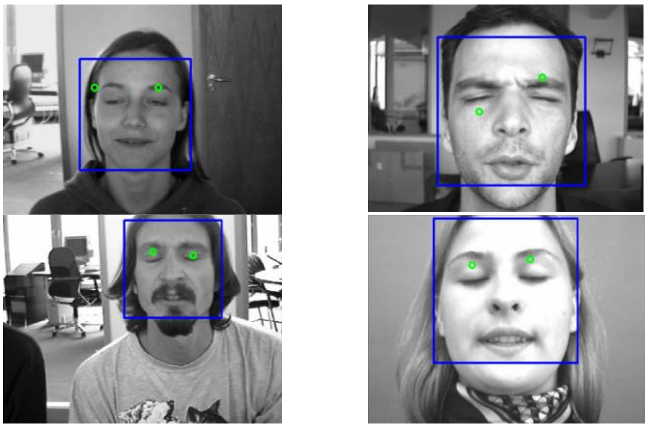
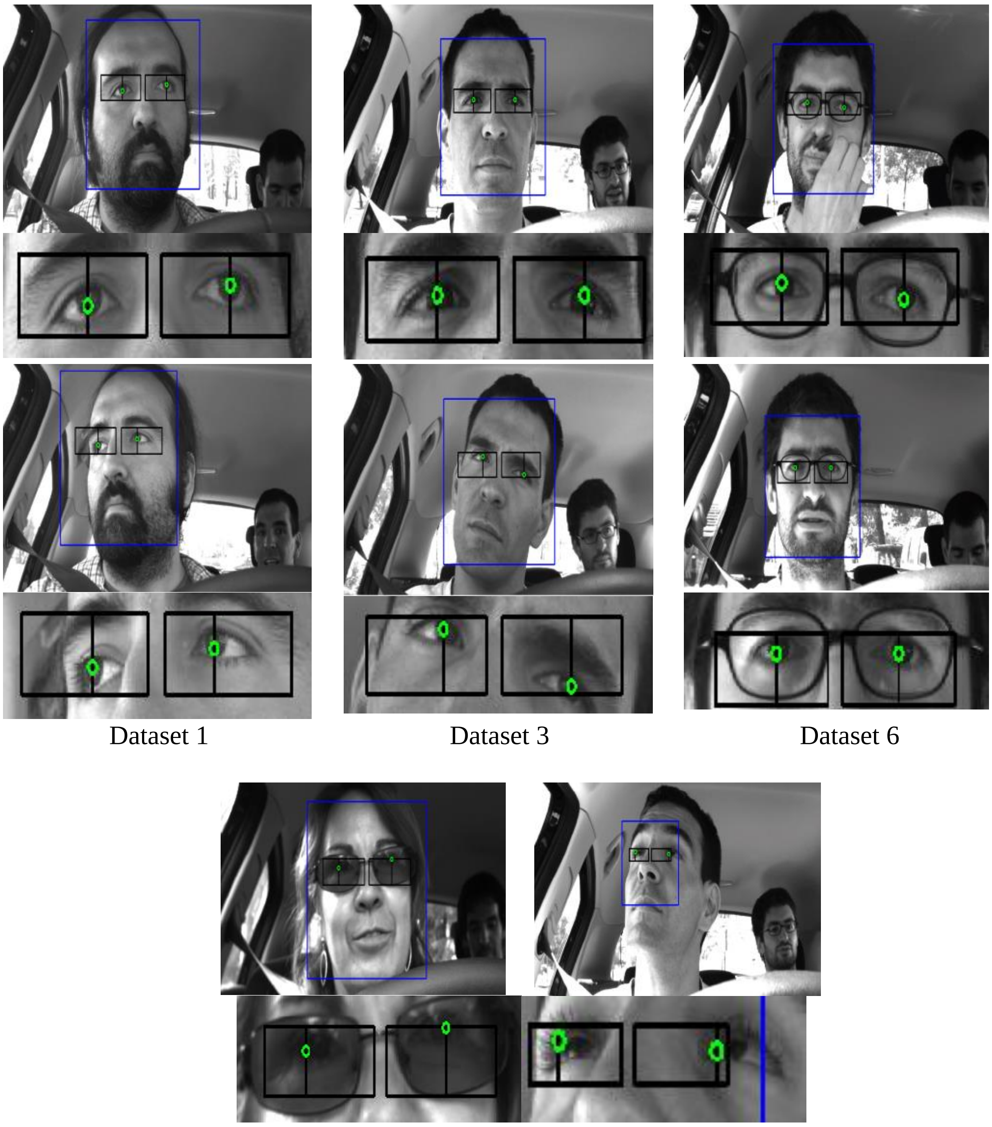


Fig. 7. Illustration of in accurate eye detection and eye center localization results when eye status is closed, on BioID dataset.

Table 3. Characteristics of normalized error for human eye distances

Normalized error	**Definition**
$e \leq 0.25$	Distance between the eye center and the eye corner
$e \leq 0.10$	Diameter of the iris
$e \leq 0.05$	Diameter of the pupil



Dataset 1

Dataset 3

Dataset 6

Fig. 8. Illustration of eye detection and eye center localization results on RS-SMV dataset.

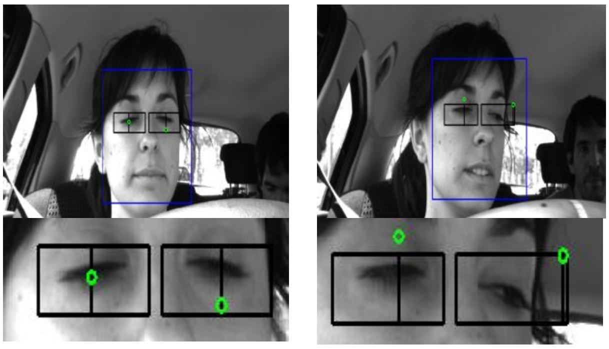


Fig. 9. Illustration of in accurate eye detection and eye center localization results when eye status is closed, on BioID dataset.

4. Conclusion

In this paper, a novel approach to a robust eye detection system is proposed, which uses the combination of HoG descriptors for feature extraction, binary PSO for feature selection, and SVM classifier for training the most promising features. HoG not only helps the detection process in enhancing the recognition rate, but also aids binary PSO in achieving a significant reduction in the number of selected features. The binary PSO reduces the feature set to the least number of optimum features for maximum recognition performance. We evaluated our algorithm using two well-known databases which can equally handle all image variations such as pose, scale, occlusion, illumination, facial expressions, and background. The detection rate for BioID dataset was 99.6% when combining HoG, binary PSO and SVM for feature extraction, feature selection and training respectively. The normalized error results for $e \leq 0.10$ and $e \leq 0.25$ were 93.4% and 98.0% respectively. We also evaluated the performance of our algorithm using RS-DMV dataset, for eye detection and normalized error results for $e \leq 0.10$ and $e \leq 0.25$ were 92.1% and 96.9% respectively. Proposed method in this paper can also be applied to several real-time applications such as driver fatigue detection, gaze estimation and eye prediction.

References

[1] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, pp. 137-154, 2004. DOI: 10.1023/B:VISI.0000013087.49260.fb

[2] W. Daud, W. M. Bukhari, M. F. Sulaima, M. Nasir, M. Naim, and M. Yahaya, "Comparison of Eye Imaging Pattern Recognition Using Neural Network," in *International Conference on Mathematics, Engineering & Industrial Applications*, 2014, pp. 117-117. DOI: 10.1063/1.4915761

[3] H. Y. Yang, X. H. Jiang, L. Wang, and Y. H. Zhang, "Eye Statement Recognition for Driver Fatigue Detection Based on Gabor Wavelet and HMM," in *Applied Mechanics and Materials*, 2012, pp. 123-129. DOI: 10.4028/www.scientific.net/AMM.128-129.123

[4] B. Cyganek and S. Gruszczyński, "Hybrid computer vision system for drivers' eye recognition and fatigue monitoring," *Neurocomputing*, vol. 126, pp. 78-94, 2014.DOI: 10.1016/j.neucom.2013.01.048

[5] H. Drewes, "Eye gaze tracking for human computer interaction," lmu, 2010.

[6] V. Rantanen, T. Vanhala, O. Tuisku, P.-H. Niemenlehto, J. Verho, V. Surakka, *et al.*, "A Wearable, Wireless Gaze Tracker with Integrated Selection Command Source for Human-Computer Interaction," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 15, pp. 795-801, 2011. DOI:10.1109/TITB.2011.2158321

[7] T. Nagamatsu, M. Yamamoto, and H. Sato, "MobiGaze: Development of a gaze interface for handheld mobile devices," in *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, 2010, pp. 3349-3354. DOI:

[8] G. AKINCI, E. Polat, and O. M. Kocak, "A video-based eye pupil detection system for diagnosing bipolar disorder," *Turkish Journal of Electrical Engineering and Computer Science*, vol. 21, pp. 2367-2377, 2013. DOI: 10.3906/elk-1204-63

[9] A. Lanatà, A. Armato, G. Valenza, and E. P. Scilingo, "Eye tracking and pupil size variation as response to affective stimuli: a preliminary study," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, 2011, pp. 78-84. DOI: 10.4108/icst.pervasivehealth.2011.246056

[10] A. Armanini and N. Conci, "Eye tracking as an accessible assistive tool," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, 2010, pp. 1-4.

[11] R. G. Lupu and F. Ungureanu, "A survey of eye tracking methods and applications," *Bul Inst Polit Iaşi,* pp. 71-86, 2013.

[12] Y. Li, X.-l. Qi, and Y.-j. Wang, "Eye detection by using fuzzy template matching and feature-parameter-based judgement," *Pattern Recognition Letters*, vol. 22, pp. 1111-1124, 2001. DOI: 10.1016/S0167-8655(01)00065-4

[13] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942–1948.

[14] H. A. Al-Mamun, N. Jahangir, M. S. Islam, and M. A. Islam, "Eye detection in facial image by genetic algorithm driven deformable template matching," *International Journal of Computer Science and Network Security,* vol. 9, pp. 287-294, 2009.

[15] A. W. Mohemmed, M. Zhang, and M. Johnston, "Particle swarm optimization based adaboost for face detection," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 2494-2501. DOI: 10.1109/CEC.2009.4983254

[16] C.-J. Lin, C.-H. Chu, C.-Y. Lee, and Y.-T. Huang, "2D/3D face recognition using neural networks based on hybrid taguchi-particle swarm optimization," in *Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on*, 2008, pp. 307-312. DOI: 10.1109/ISDA.2008.286

[17] A. Sharma and N. Singh, "Object detection in image using particle swarm optimization," *International Journal of Engineering and Technology,* vol. 2, pp. 419-426, 2010.

[18] Š. Ivekovič, E. Trucco, and Y. R. Petillot, "Human body pose estimation with particle swarm optimisation," *Evolutionary Computation,* vol. 16, pp. 509-528, 2008. DOI: 10.1162/evco.2008.16.4.509

[19] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems,* vol. 64, pp. 22-31, 2014. DOI: 10.1016/j.knosys.2014.03.015

[20] Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In Proceedings of the conference on systems, man, and cybernetics (pp. 4104–4109). Piscataway: IEEE. DOI: 10.1109/ICSMC.1997.637339

[21] Agrafiotis, D. K., & Cedeño, W. (2002). Feature selection for structure-activity correlation using binary particle swarms. Journal of Medicinal Chemistry, 45(5), 1098–1107. DOI: 10.1021/jm0104668

[22] Mohan, C. K., & Al-Kazemi, B. (2001). Discrete particle swarm optimization. In Proceedings of the workshop on particle swarm optimization, Indianapolis, IN, Purdue School of Engineering and Technology, IUPUI.

[23] Pamparä, G., Franken, N., & Engelbrecht, A. P. (2005). Combining particle swarm optimization with angle modulation to solve binary problems. In Proceedings of the IEEE congress on evolutionary computation (CEC) (pp. 225–239). Piscataway: IEEE. DOI: 10.1109/CEC.2005.1554671

[24] M. Dahmane and J. Meunier, "Emotion recognition using dynamic grid-based HoG features," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, 2011, pp. 884-888. DOI: 10.1109/FG.2011.5771368

[25] D. Monzo, A. Albiol, J. Sastre, and A. Albiol, "Precise eye localization using HOG descriptors," *Machine Vision and Applications,* vol. 22, pp. 471-480, 2011. DOI: 10.1007/s00138-010-0273-0

[26] C. Shu, X. Ding, and C. Fang, "Histogram of the oriented gradient for face recognition," *Tsinghua Science & Technology*, vol. 16, pp. 216-224, 2011. DOI: 10.1016/S1007-0214(11)70032-3

[27] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, "Face recognition using HOG–EBGM," *Pattern Recognition Letters*, vol. 29, pp. 1537-1543, 2008. DOI: 10.1016/j.patrec.2008.03.017

[28] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognition Letters,* vol. 32, pp. 1598-1603, 2011. DOI: 10.1016/j.patrec.2011.01.004

[29] S. E. Lee, K. Min, and T. Suh, "Accelerating Histograms of Oriented Gradients descriptor extraction for pedestrian recognition," *Computers & Electrical Engineering,* vol. 39, pp. 1043-1048, 2013. DOI: 10.1016/j.compeleceng.2013.04.001

[30] T. Kobayashi, A. Hidaka, and T. Kurita, "Selection of histograms of oriented gradients features for pedestrian detection," in *Neural Information Processing*, 2008, pp. 598-607. DOI: 10.1007/978-3-540-69162-4_62

[31] Y.-Q. Wang, "An Analysis of the Viola-Jones face detection algorithm," *Image Processing On Line,* vol. 4, pp. 128-148, 2014. DOI: 10.5201/ipol.2014.104

[32] P. Viola and M. J. Jones, Robust real-time face detection, International Journal of Computer Vision, 57 (2004), pp.137. DOI:10.1023/B:VISI.0000013087.49260.fb

[33] F. Timm and E. Barth, "Accurate Eye Centre Localisation by Means of Gradients," in *VISAPP,* 2011, pp. 125-130.

[34] Y. Bai, L. Guo, L. Jin, and Q. Huang, "A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 3305-3308. DOI: 10.1109/ICIP.2009.5413938

[35] E. Corvee and F. Bremond, "Body parts detection for people tracking using trees of histogram of oriented gradient descriptors," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, 2010, pp. 469-475. DOI: 10.1109/AVSS.2010.51

[36] G. Xu, X. Wu, L. Liu, and Z. Wu, "Real-time pedestrian detection based on edge factor and Histogram of Oriented Gradient," in *Information and Automation (ICIA), 2011 IEEE International Conference on*, 2011, pp. 384-389. DOI: 10.1109/ICINFA.2011.5949022

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 886-893. DOI:10.1109/CVPR.2005.177

[38] G. Tsai, "Histogram of oriented gradients," *University of Michigan,* 2010.

[39] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry,* vol. 32, pp. 29-38, 2008. DOI:10.1016/j.compbiolchem.2007.09.005

[40] B. Xue, M. Zhang, and W. N. Browne, "Multi-objective particle swarm optimisation (PSO) for feature selection," in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, 2012, pp. 81-88. DOI: 10.1145/2330163.2330175

[41] C.-L. Huang and J.-F. Dun, "A distributed PSO–SVM hybrid system with feature selection and parameter optimization," *Applied Soft Computing,* vol. 8, pp. 1381-1391, 2008. DOI: 10.1016/j.asoc.2007.10.007

[42] B. Chakraborty, "Feature subset selection by particle swarm optimization with fuzzy fitness function," in *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, 2008, pp. 1038-1042. DOI: 10.1109/ISKE.2008.4731082

[43] L. Cervante, B. Xue, M. Zhang, and L. Shang, "Binary particle swarm optimisation for feature selection: A filter based approach," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, 2012, pp. 1-8. DOI:10.1109/CEC.2012.6256452

[44] C. M. Bishop, *Pattern recognition and machine learning*: springer, 2006.

[45] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 161-168. DOI:10.1145/1143844.1143865

[46] (15 July 2015). *BioID Face Database*. Available: https://www.bioid.com/About/BioID-Face-Database

[47] J. Nuevo, L.M. Gergasa, P. Jimenez. RSMAT: Robust simultaneous modeling and tracking, Pattern Recognition Letters. 2010. DOI: 10.1016/j.patrec.2010.07.016

[48] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," in *Audio-and video-based biometric person authentication*, 2001, pp. 90-95. DOI: 10.1007/3-540-45344-X_14

[49] L. Nanni, A. Lumini, Combining face and eye detectors in a high-performance face-detection system, IEEE MultiMedia 19 (2012) 20–27.

[50] W. Li, Y. Wang, Y. Wang, Eye location via a novel integral projection function and radial symmetry transform, International Journal of Digital Content Technology and its Applications 5 (8) (2011) 70–80.

[51] D. Yi, Z. Lei, S.Z. Li, A robust eye localization method for low quality face images, in: Proceedings of International Joint Conference on Biometrics, 2011, pp. 1–6. DOI: 10.1109/IJCB.2011.6117499

[52] Asadifard, M. and Shanbezadeh, J. (2010). Automatic adaptive center of pupil detection using face detection and cdf analysis. In Proceedings of the IMECS, volume I, pages 130–133, Hong Kong. Newswood Limited.

[53] B. Kroon, S. Maas, S. Boughorbel, A. Hanjalic, Eye localization in low and standard definition content with application to face matching, Computer Vision and Image Understanding 113 (8) (2009) 921–933. DOI:10.1016/j.cviu.2009.03.013

[54] P. Campadelli, R. Lanzarotti, G. Lipori, Precise eye and mouth localization, International Journal of Pattern Recognition and Artificial Intelligence 23 (3) (2009) 359–377.DOI: 10.1142/S0218001409007259

[55] Kroon, B., Hanjalic, A., and Maas, S. (2008). Eye localization for face matching: is it always useful and under what conditions? In Proceedings of the 2008 CIVR, pages 379–388, Ontario, Canada. ACM. DOI:10.1145/1386352.1386401

[56] Valenti, R. and Gevers, T. (2008). Accurate eye center location and tracking using isophote curvature. In Proceedings of the CVPR, pages 1–8, Alaska. IEEE. DOI: 10.1109/CVPR.2008.4587529

[57] Turkan, M., Pardas, M., and Cetin, A. E. (2007). Human eye localization using edge projections. In Proceedings of the 2nd VISAPP, pages 410–415. INSTICC.

[58] Niu, Z., Shan, S., Yan, S., Chen, X., and Gao, W. (2006). 2d cascaded adaboost for eye localization. In Proceedings of the 18th IEEE ICPR, volume 2, pages 1216– 1219, Hong Kong. IEEE. DOI:10.1109/ICPR.2006.1194

[59] Chen, D., Tang, X., Ou, Z., and Xi, N. (2006). A hierarchical floatboost and mlp classifier for mobile phone embedded eye location system. In Proceedings of the 3rd ISNN, LNCS, pages 20–25, China. Springer. DOI: 10.1007/11760023_4

[60] Cristinacce, D., Cootes, T., and Scott, I. (2004). A multistage approach to facial feature detection. In Proceedings of the 15th BMVC, pages 277–286, England. DOI: 0.5244/C.18.30

[61] Behnke, S. (2002). Learning face localization using hierarchical recurrent networks. In Proceedings of the ICANN, LNCS, pages 135–135. Springer. DOI: 10.1007/3-540-46084-5_213