Face Detection by Using OpenCV’s Viola-Jones Algorithm based on coding eyes

Abdul Mohsen Abdul Hossen*, Raheem Abd Alsaheb Ogla, Maitham Mahmood Ali
Department of Computer Science, University of Technology, Baghdad, Iraq.

Abstract
Facial identification is one of the biometrical approaches implemented for identifying any facial image with the use of the basic properties of that face. In this paper, we propose a new improved approach for face detection based on coding eyes by using OpenCV’s Viola-Jones algorithm which removes the falsely detected faces depending on coding eyes. The Haar training module in OpenCV is an implementation of the Viola-Jones framework, the training algorithm takes as input a training group of positive and negative images, and generates strong features in the format of an XML file which is capable of subsequently being utilized for detecting the wanted face and eyes in images, the integral image is used to speed up Haar-like features calculation for each image in (MIT, FERET) dataset and the adaboost algorithm is implemented to collect the weak classifiers and produce strong classifier. By using classifier cascade process, the speed and accuracy of face detection system is increased. The proposed method has accuracy is about 98.97% for detection faces.

Keywords: Face detection, Viola Jones, eye detection, OpenCV, frontal faces.

*Email: AbdulMoohsen53@yahoo.com
Introduction

Face detection in images and video sequences has become an important part of the recognition. For face recognition, detection must be done when multiple faces are present in an image. The correct suited face detection is still in demand since false identification leads to wrong face detection and wastage of computational power as well as time. Several face detection techniques are available like hidden markov models, neural networks, and optical flow to Haar cascade classifiers. The Viola-Jones algorithm used by most of the researchers based on Haar cascade classifiers [1]. OpenCV is the open source computer vision library by Intel. It has several implemented and optimized algorithms of image processing usage. One of the most popular algorithms is Viola-Jones algorithm which is capable of detecting face and objects. The Viola-Jones method was given by Viola and Jones in the year 2001 and has been improved several times. Faces can be detected by using the integral image to calculate feature extraction [2-6].

Related Work

Many authors have worked on various methods for face detection. One of them is the eigen faces method which is used by several authors. Neural networks and artificial neural networks are also extensively used. Improvements in Viola Jones have also been made [4]. For false positives filtering, Alpika et al. gives an insight about various colour spaces that can be used for filtering the wrong face detections and specify the skin colour ranges as well [6]. For detection of the faces through eyes, Wong et al. [7] gives information to evaluate the face length and breadth by various parameters and relationship between different parameters of the face.

The hypothesis after the evaluation of the detection object scheme introduced by Viola and Jones was improved by these authors [8, 9]; this hypothesis makes it possible to train the classifier for finding objects that are partially occluded. The explanation is possible to progress the hit rate of the classifier in the partial occlusions presence by training the classifiers with occluded examples.

Object detection framework

One of the first methods suggested by Viola and Jones in 2001 to provide object detection at very rapid rates [1]. It is the method for fast and correct object detection by the use of adaboost machine learning. The main features of Viola-Jones algorithm are:

1. Integral image: Is required to fast detection of objects needs computation of Haar features, and to compute them an integral image is obtained using few operations per pixel. After this calculation, in constant time Haar features of any type can be computed.
2. The adaboost Learning algorithm, it created efficient classifiers from set of extracting relevant visual features. Learning must exclude a majority of features available in the image for fast classification. Critical features extracted by this algorithm while discarding all other unimportant features.
3. The cascade classifier which is concentrated on objects such as parts and discards the background as shown in Figure-1. Cascade is a kind of mechanism that knows its region of interest and rejected areas are not likely to include any object. This is very fast in real-time detection [10].

![Figure 1-Haar cascade classifier.](image)

Algorithm

For the sake of studying the algorithm in all its details, the beginning will be with the image properties for the task of classification:-
Haar-like features

The Viola-Jones algorithm utilizes Haar-like features, that is, a scalar item between the image and some Haar-like templates. All the more absolutely, let Img and Pn mean an image and a pattern, both of similar size $G \times G$ [since the value of $G=24$] the feature connected with pattern Pn of image Img is characterized by

$$
\sum_{i=1}^{G} \sum_{j=1}^{G} Img(i, j)1Pn(i, j) \text{is white} - \sum_{i=1}^{G} \sum_{j=1}^{G} Img(i, j)1Pn(i, j) \text{is black} \quad \cdots \cdots \cdots (1)
$$

To compensate the effect of different lighting conditions, all the images should be mean, and variance normalized beforehand. Those images with difference lower than one, having little information of interest in the first place, are left out of consideration.

Integral image

Rather than summing up every one of the pixels inside a rectangular window, this strategy reflects the utilization of total distribution functions. The recognized image IImg of Img can the contact details of as:

$$
\text{IImg}(i, j) := \begin{cases} 
\sum_{1 \leq s \leq i} \sum_{1 \leq t \leq G} Img(es, ts), & 1 \leq i \leq G \text{ and } 1 \leq j \leq G, \\
0, & \text{otherwise} 
\end{cases} ............... (2)
$$

Is so defined as that equation below:-

$$
\sum_{G1 \leq G2} \sum_{G3 \leq G4} Img(i, j) \\
= IImg(G2, G4) - IImg(G2, G3 - 1) - IImg(G1 - 1, G4) + IImg(G1 - 1, G3 - 1) \cdots \cdots \cdots (3)
$$

Holds for all $G1 \leq G2 \text{ and } G3 \leq G4$.

As a result, computing an image’s rectangular local sum requires at most four simple operations given its integral image. Moreover, obtaining the integral image itself can be done in linear time:

Setting $G1 = G2$ and $G3 = G4$ in (eq.3), find that:

$$
\text{Img } (G1, G3) = IImg (G1, G3) - IImg (G1, G3 - 1) - IImg (G1 - one, G3) + IImg (G1 - one, G3 - 1) \cdots \cdots \cdots (4)
$$

Features scaling

Scaling for Haar-like features has been used rather than scaling image sub-windows to get a high-speed detection system. The feature (1x2) where the white rectangle is above the black rectangle for the 24x24 sub-window matrix (x1 to x23, y1 to y23), this feature starts at the position $x1,y1$, sum the pixel in white rectangle and subtracted from the pixels amount of in black rectangle, then the first feature is obtained. All calculations are based on this classifier. After the scan, the whole 24x24 sub-window, this feature is scaled and moved to the next feature size, which will be (1x4) in ($x1, y1$) position, the feature will keep increment in size. It will be (1x8), (1x16) and so on; it cannot exceed the size of sub-window. After a complete scanning of all the 24x24 sub-windows, another Haar-like feature is used in the same steps. For each sub-window 24 $\times$ 24 images, feature (1x2) there are 43200 features, and the feature (1x3) where the black rectangle is the two weight rectangle as shown in Figure-1 (b)and(d)) category. For each sub-window (24 $\times$ 24) image there are 27600 features while in feature (1x4) where there are two black rectangles and two Wight rectangles as shown in Figure -1(e)) category there are 20736 features, hence 162336 features in all.

Feature Selection with AdaBoost

For face detection, it assumes the type of f: $R^{dn} \rightarrow \{-1,1\}$, where 1 implies that there is a face and -1 the opposite and N is the number of Haar-like features removed from an image. Given the probabilistic weights, $w_{g} \in R$, relegated to a training set mentioned up of n observation-label pairs $(x_i; y_i)$.
\[
\sum_{l=0}^{n} w_l g_1 y_i = f(x_i) \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdOTS
All the correctly detected faces have almost similar length and breadth of eyes. Thus, to take the average value for both the width of eyes as $w_{\text{Avg}}$ and a distance between eyes as $d_{\text{Avg}}$ are calculated by using equation (11).

$$\text{Average} = \frac{\sum_{i=1}^{n} x_i}{n}$$  \hspace{1cm} \text{...(11)}

Let the distance between eyes are $d_{\text{eye}}$, the average distance between eyes are $d_{\text{avg}}$ and width of the eyes is $w_{\text{eye}}$. For both eyes in the image, if the respective value of $d_{\text{eye}}$ and $w_{\text{avg}}$ are such that:

For double eye face detection:

$$\frac{d_{\text{avg}}}{2} < d_{\text{eye}} < 2 \times d_{\text{avg}}$$ \hspace{1cm} \text{...(12)}

In the images, it is seen that the eyes in the picture are of almost same sizes. Thus, for each eye the equations (8), (9) and (10) are used to calculate values of $w_{\text{eye}}$ and $d_{\text{eye}}$. All those eyes whose parameters do not satisfy equation (12) are considered outliers, which either have the very large size or are very small. Therefore, all the eyes, which were less than twice these averages and greater than half of these averages, were only selected. It removes ‘falsely detected eyes if their value does not lie between. Thus, the Euclidean distance between $(x_1, y_1)$ and $(x_2, y_2)$ is calculated as:

$$\text{Dist} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$ \hspace{1cm} \text{...(13)}

Proposed System

The proposed methodology is focused on the detection of faces based on coding eyes. First, there are two stages to the implementation. To accomplish this task, two steps the first stage is Open CV Haar training module: The implementation of the Viola-Jones framework was Haar training module in Open CV. The input is a training set of positive and negative images takes as in the training algorithm, and creates a strong features in the image of a XML file that can next be used to detect the goal object (face and eyes) in images since training preparation stage creates a strong classifier for face detection system this stage uses the Haar-like features that are extracted from each image in training database. To increase the speed of the scheme, the integral image representation is utilized to high speed Haar-like features calculation for each image in (MIT,FERET) dataset and the adaboost machine-learning algorithm is used to collect the weak classifiers and produce strong classifiers while the evaluation stage: uses the strong classifier to implement face detection system. To detect face and eyes, the Viola and Jones front face detection algorithm has been used, the Viola and Jones eyes detection and the Viola-Jones eyes with class's detection. By using classifier cascade process, the speed and accuracy of face detection system are increased as shown in Figure- 3 that show the open CV classifier. The database that used in this work was the MIT Media Laboratory’s database of more...
than 7000 images and The FERET database is a standard[citation needed] dataset used for facial recognition system evaluation.

**Implementation of Viola-Jones algorithm to detect face based on coding eyes**

Now through the eye detection following are the further steps: Step 1: Eyes detected in already detected faces and step 2: Detection of eyes as shown in Figure- 4.

**Figure 3- Flow chart the open CV classifier.**

**Figure 4- Eyes detected in already exposed faces.**
And the algorithm (1) represents the detection of eyes in already exposed faces as seen below:

**Algorithm (1): Eyes detected.**

**Input**: read image file  
**Output**: Eyes detected in already detected faces

**Step 1**: The coordinates for all the faces previously detected were stored.  
When these eyes were detected, they were first checked for their presence inside the coordinates of the face (see (14) & (15)). All the eyes of existing detected faces are considered for the algorithm. If (p1, q1) and (p2, q2) are coordinates of the face and (r1, s1) and (r2, s2) are coordinates of the eyes, (eyes are present inside the face)

\[
\begin{align*}
p_1 &< (r_1 \text{ and } r_2 \text{ both}) < p_2 \quad (14) \\
q_1 &< (s_1 \text{ and } s_2 \text{ both}) < q_2 \quad (15)
\end{align*}
\]

Otherwise, Eyes are Discarded (lie outside the face).

**Step 2**: Detection of eyes.  
For the remaining eyes, compute the distance between the eyes by using (eq.13), and If that distance satisfies (eq.13), Then distance signifies that eyes are paired eyes of the face or not. When one eye is detected then, it is checked with other eyes for being a pair of eyes by their intermediate distance and near about same sizes. If the two detected eyes are rectangles such that their topmost, leftmost points are (x1, y1) and (x2, y2), Compute the distance between these two points using (eq.13). For both eyes left If (there is an eye with Euclidean distance between the eyes less than 2 * davg) Then, Make these a pair. Remove all nearby eyes.

Then,  
**Condition-1: Pair of eyes is found.**

**Condition-1**:  
For each pair found, find the region of interest of face i.e. makes the Square, which represents the face?  
If more than, one pair of eyes is formed like.  
1. (x1, y1), (x2, y2)  
2. (X1, y1), (x3, y3) ... and so on

**For all the pair of eyes:**  
The Region of interest (ROI) is constructed for all the pair of eyes, shown in (Figure- 5) as R1 and R2. The ROI is formed with taken min of x1, x2 & y1, y2. The min(x) and min(y) is the starting point of the ROI; this rectangle is extended on both sides with the (eq.8)

**Step3**: End.
**Results**

A- **The training and detection stage:**

Ready training cascade of weak classifiers have been used to detect a human face based on coding eyes.

B- **Haar-cascade Detection in OpenCV**

Here, we will manage with detection. OpenCV already contains many pre-trained classifiers for face, eyes. Stored those .xml files in OpenCV/source/data/Haar cascades/folder. Then, the frontal face, eyes, and eyes with the class detector with OpenCV have been made.

First, the required .xml classifiers were loaded for detecting the human face and eyes in front of the camera, and usually more than one shot per person is stored and more than an environment to increase the efficiency of the system, stored these images in the custom folder for images and store the images information in CSV file text.

Presently the faces in the picture have been discovered. If faces are found, it gives back the places of detected faces as the rectangle \((x, y, w, h)\). When getting these areas, it is possible to make an ROI for the face and apply eyes detection on this ROI (since eyes are dependably on the face as a circle). As appeared in Figure- 6.

![Figure 6](image6.png)

**Figure 6-** Haar-cascade detection in OpenCV.

Then the input image is loaded in the grayscale mode as illustrated in Figure- 7.

![Figure 7](image7.png)

**Figure 7-** Shows input image in grayscale mode.
After the open CV_train cascade application has finished its work, the images are stored in folders; each folder has the name like the (id) of a person in the database. As shown in Figure- 8.

![Figure 8](image)

**Figure 8-** the database of person.

### C. Training Database stage:

Database content from at least one table is composed. The table comprises of record (database) or more, and the record contains a field (field) or more. For instance, it’s a specific individual record incorporates a few fields, for example, employee number - the employee - name of the employee - title - Birthday degree - and email - phone number, and other information data stored in the database in an organized way, where the data bank (database) to encourage the handling and display information in the discovery and identification phase, and empower the client to include and alter them. As appeared in Figure- 9.

![Figure 9](image)

**Figure 9-** training database of information stage.

Table-1 shows the comparison of face detection accuracy for proposed algorithm for face detection and other method face detection that using the same dataset (MIT).

**Table 1-** Comparison accuracy of face detection for proposed algorithm with other methods.

<table>
<thead>
<tr>
<th>Face detection system</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang et al.—FA[11]</td>
<td>89.4%</td>
</tr>
<tr>
<td>Yang et al.—LDA[11]</td>
<td>91.5%</td>
</tr>
<tr>
<td>Roth et al.[12]</td>
<td>94.1%</td>
</tr>
<tr>
<td>Proposed algorithm</td>
<td>98.97%</td>
</tr>
</tbody>
</table>
Table-2 shows the comparison of face detection for viola-jones detection and proposed algorithm.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Viola &amp; Jones face detection</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>False Positive Rate</td>
<td>30.09346</td>
<td>0.954836</td>
</tr>
<tr>
<td>False Negative Rate</td>
<td>8.09346</td>
<td>0.073577</td>
</tr>
<tr>
<td>Accuracy</td>
<td>61.81308</td>
<td>98.97158</td>
</tr>
</tbody>
</table>

The percentage is calculated by total faces detected by our algorithm/whole faces present. The algorithm achieved 90.89 % accuracy in face detection, detecting the faces depending on the specified distance from the camera.

Here, the correct percentage becomes 98.97 % if the images are clear and eyes are correctly detected. Eyes play an important role for detecting front and faces because it reduces the false positive detected faces. On the other hand, the proposed design consumes less time and detects eyes accurately as shown in Table-3 and Figure-10 shows average time consumed by techniques and their face, eyes detection accuracy.

<table>
<thead>
<tr>
<th>Average accuracy in image auto size</th>
<th>Viola Jones</th>
<th>Hybrid design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average time consumed per image</td>
<td>1.467s</td>
<td>0.9067s</td>
</tr>
<tr>
<td>Face detection accuracy</td>
<td>60.12%</td>
<td>% 90.89</td>
</tr>
<tr>
<td>Eyes detection accuracy</td>
<td>61.81%</td>
<td>98.97 %</td>
</tr>
</tbody>
</table>

![Figure 10](image-url)  
**Figure 10-** Graph depicting Faces detection based on eyes.

**Conclusions**

In this work, Viola and Jones face detection system based on coding eyes based Haar-like classifiers have been presented that reduce calculation time and achieving high rate for detection by using an integral image is to compute Haar-like features.

Fundamental picture is to register Haar-like elements. The image does not need to calculate multiscales. This is reducing the image processing for detecting the object. AdaBoost technique is used for feature selection. It is an efficient and aggressive technique for selection of Haar-like features. The system designer can use a very large set of Haar-like features for the learning process. Only a small set of Haar-like features are used to detect the face in real time. Constructing a cascade of classifiers technique is used to reduce computation time radically and improving detection rate. First cascade
stages are used to reject a plurality of the negative image and focus on face regions. The cascade presented is simple and easy in structure. Using open CV based on Haar cascade classifier (frontal face, eyes, eyes with class) for building face detection system will give very high speed because of executing the code in parallel.

References