ISSN: 0067-2904

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

Abdul Mohssen Jaber Hossen*, Raheem Abdul Sahib 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 proposes a new improved approach for face detection based on coding eyes by using Open CV's Viola-Jones algorithm which removes the falsely detected faces depending on coding eyes. The Haar training module in Open CV 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, Open CV, frontal faces.

كشف الوجه باستخدام مكتبة برمجية مفتوحة للرؤية الحاسوبية في خوارزمية فيولا جونز على اساس شفرة العيون

عبد المحسن جابر حسين*، رحيم عبد الصاحب عكلة، ميثم محمود علي
قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق.

الخلاصة
تحديد الوجه هو احد الطرق البيومترية التي تنفذ لتحديد أي صورة للوجه مع استخدام الخصائص الأساسية لهذا الوجه. تقترح هذه الورقة نهج جديد ومحسن لكشف الوجه بناءا على شفرة العينين باستخدام خوارزمية فيولا جونز وبالاعتماد على مكتبة برمجية مفتوحة للرؤية الحاسوبية التي تزيل اخطاء الكشف عن الوجوه المعتمدة على شفرة العينين ، ويتم تنفيذ قالب فيولا جونز باستخدام وحدة التدريب حيث ان خوارزمية التدريب تعتبر كمدخل لمجموعة التدريب التي تضم الصور الايجابية والسلبية ويتم انتاج خصائص صورة مميزة على شكل ملف من نوع (XML) والتي ممكن استخدامها لاحقا للكشف عن العينين والوجه المطلوب في الصورة ، ويستخدم تكامل الصور لغرض تسريع حساب خصائص الهار لكل صورة من قاعدة البيانات (MIT & FERET) ويتم جمع المصنفات الضعيفة لغرض انتاج مصنف قوي عن طريق تنفيذ خوارزمية الادبوست . كما ويتم زيادة الدقة والسرعة باستخدام شلال المصنفات حيث اثبتت الطريقة المقترحة ان لديها دقة حوالي 98.97٪ لتحديد الوجوه.

*Email: AbdulMoohsen53@yahoo.com

Introduction

Face detection in images and video sequences has become an important part of the recognition. For face recognition, detection is 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 researcher based on Haar cascade classifiers [1]. Open CV 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 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].

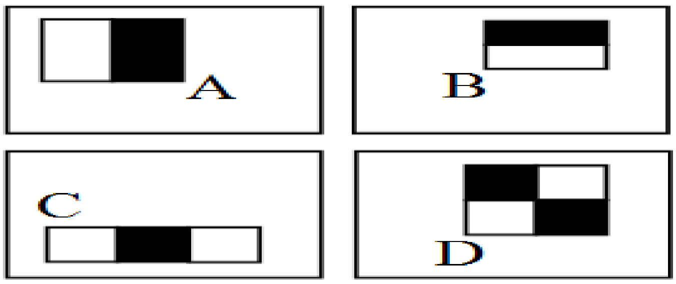

Figure 1-Haar cascade classifier**.**

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 × G [since the value of G=24] the feature connected with pattern Pn of image Img is characterized by

$$\sum_{1\le i\le G}\sum_{1\le j\le G} Img(i.j)1Pn(i,j) is\ white - \sum_{1\le i\le G}\sum_{1\le j\le G} Img(i,j)1Pn(i,j) is\ black \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

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:

$$\mathrm{IImg(i.j)} \coloneqq \begin{cases} \sum 1 \le s \le i\ Imag(es,t), 1 \le i \le G\ and\ 1 \le j \le G \\ 0, \quad otherwise \quad , \end{cases} \ldots\ldots\ldots\ldots (2)$$

Is so defined as that equation below:-

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

Holds for all G1 ≤ G2 and G3 ≤ G4.

s 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:

Img (G1, G3) = IImg (G1, G3) − IImg (G1, G3 − 1) − IImg (G1 − one, G3) + IImg (G1 − one, G3 − 1)……………………………………………………………………………….. (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 × 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 × 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, wg ϵ R_+ relegated to a training set mentioned up of n observation-label pairs (xi; yi).

$$\sum_{I=0}^{n} \mathrm{wg1yi} \neq \mathrm{f(xi)} \quad \ldots \quad (5)$$

The building block of the Viola-Jones face detector is a decision stump, or a depth one decision tree, parameterized by a feature f ϵ {1; …... d}, a threshold t ϵ R and a toggle to ϵ {−1; 1}.
Given an observation β ϵ R^{dn}, a decision stump C predicts its label using the following rule:-

$$C(\beta) = (1\ \pi_{f\ \beta} \geq t - 1\ \pi_{f\ \beta} < t)\ to = (1\ \pi_{f\ \beta} \geq t - 1\ \pi_{f\ \beta} < t)1_{\ to=1} + (1\ \pi_{f\ \beta} < t - 1\ \pi_{f\ \beta} \geq t)1_{\ to=-1} \in \{-1; 1\}; \quad \ldots \quad (6)$$

Where πf β is the feature vector's f-the coordinate. Envision a threshold put some place the genuine line, if the toggle is set to 1, the resulting rule will pronounce an example β positive if πf β is higher than the threshold and negative otherwise. This permits us to evaluate the state's experimental error thereby selecting the toggle that fits the dataset better. Since margin:-

$$\min_{i:y_i=-1} |\pi f \beta_i - t| + \min_{i:y_i=1} |\pi f \beta_i - t|, \quad \ldots \quad (7)$$

Classifier Cascade

The strong classifier classifies each sub-window. But we have a large number of robust classifiers. This makes the time of image processing long. To improve the detection time with efficient computation, a cascade of robust classifiers is used. The idea is to construct smaller and efficient classifiers based on the sub-windows within the input image. Here the strong classifier is assorted from the best classifier to the worst classifier. The best classifiers that have excellent features will reject the negative sub-windows, rotation and noise. Additional negative sub-windows, rotation and noise are eliminated by subsequent layers called "stages" while requiring additional computation. Each stage contains classifiers. In this project, 20 stages of the cascade that are produced by Open CV and (MIT, FERET) dataset have been used, started from the stronger classifiers and ended with weaker classifiers. For this reason, the number of sub-windows supposed to be reduced rapidly after several stages of processing and this gives a good accuracy and less processing time.

In each stage, Haar-like features of the sub-window are calculated, then these functions are compared to the threshold to see if it is a face or not.

The threshold value is taken from the strong classifier. Also, it is possible to adjust this threshold till obtaining the face. If features satisfied the threshold condition, the sub-window goes to the next stage of the cascade to do the same process. If features do not deal with the threshed condition in any stage, the sub-window is rejected. If sub-window passes all cascade stages, the sub-window is indicated as the face image.

The classifier cannot be trained for every tilt faces. It is observed that eyes are always present whether the face is frontal or tilted.

Viola-Jones is capable of detecting the eyes through both eyes classifier produced by OpenCV. First, the average eye width and average eye distance are calculated with the help of already detected actual faces. After detecting the tilted faces classifier produced by Open CV, the following parameters are used for detection of eyes classifier produced by OpenCV and forming the region of interest for faces as shown in Figure- 2 that show. For these distances, the following equations are used:

Hface =1.8 d eye……………………………………………………………… (8)
Heye =0.2 h face ……………………………………………………………… (9)
Weye =0.225h face …………………………………………………………… (10)

Where,
Hface = height of the face
Deye = distance between eyes h eye = height of eyes
weye = width of eyes

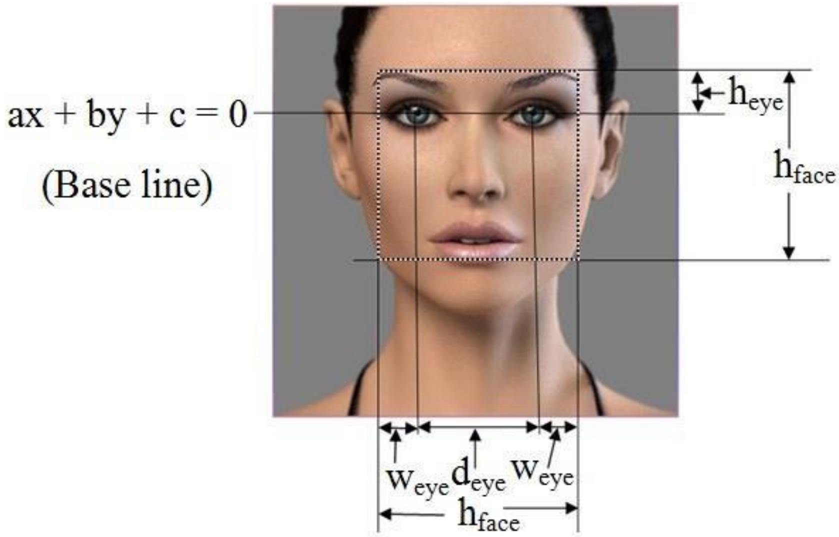

Figure 2- Variable Description

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 Avg and a distance between eyes as d Avg are calculated by using equation (11).

$$Average = \frac{\sum_{i=1}^{n} xi}{n} \quad \ldots\ldots (11)$$

Let the distance between eyes are deye, the average distance between eyes are davg and width of the eyes is weye. For both eyes in the image, if the respective value of deye and wavg are such that:
For double eye face detection:
davg / 2 < deye < 2 * davg ………………………………………………………………… (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 weye and deye. 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, y1) and (x_2, y2) is calculated as:

$$Dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad \ldots\ldots (13)$$

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.

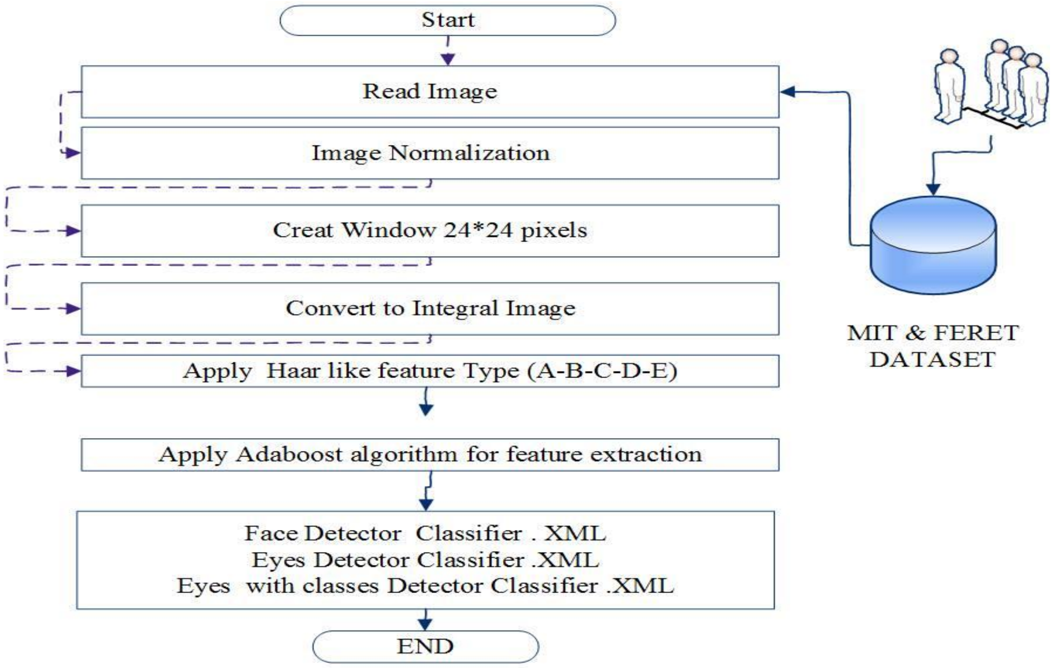

Figure 3- Flow chart the open CV classifier.

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.

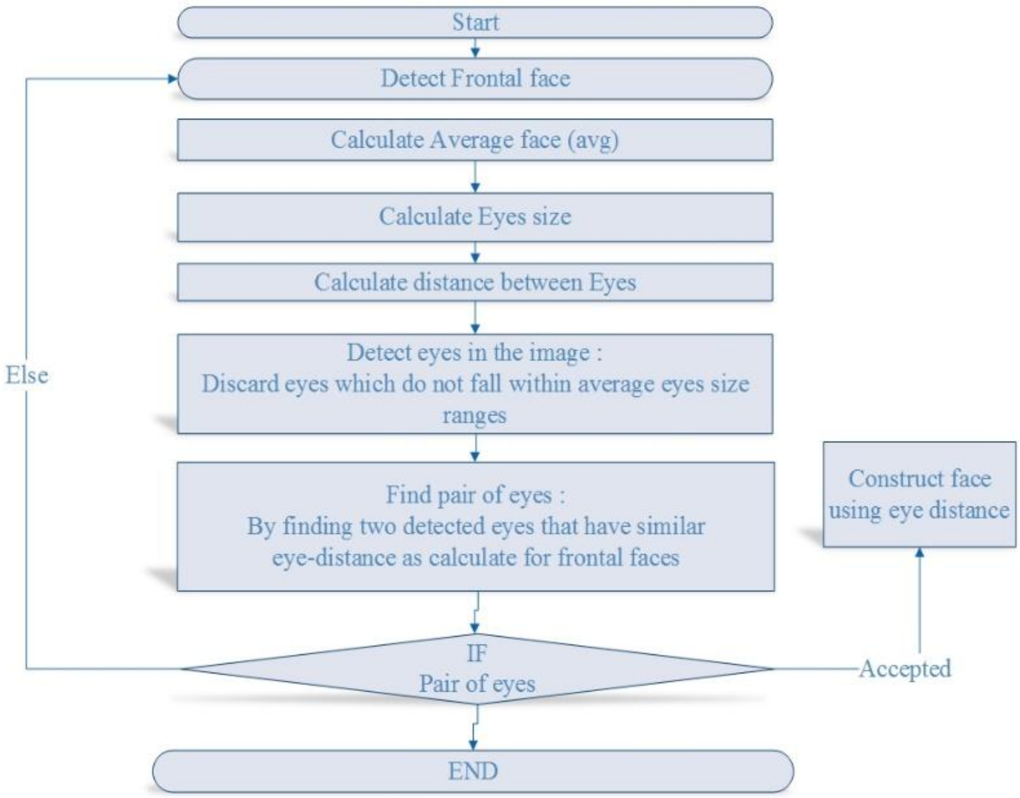

Figure 4- Eyes detected in already exposed faces.

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)
Then

p1< (r1 and r2 both) < p2 (14)

q1< (s1 and s2 both) < q2 (15)

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 x_1, x_2 & y_1, 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.

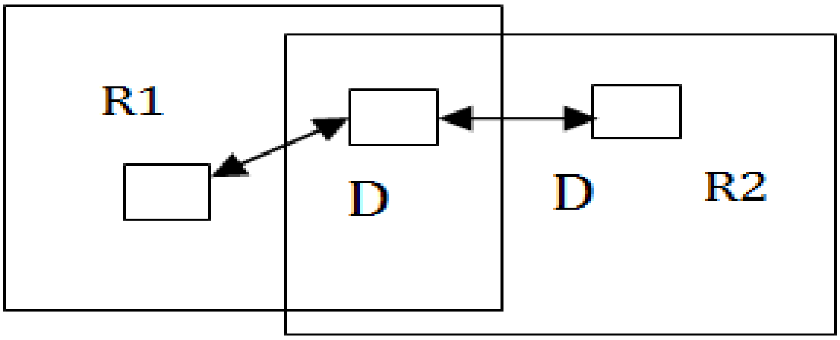

Figure 5- Make a pair of eyes for face detection.

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 Open CV

Here, we will manage with detection. Open CV 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- Haar-cascade detection in OpenCV.

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

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- 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.

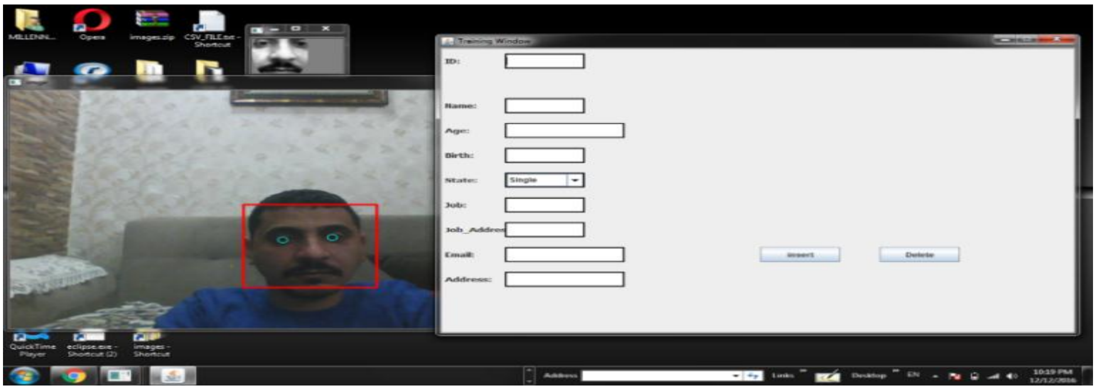

Figure 9-training database of information stage**.**

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.

Face detection system	**Accuracy**
Yang *et al.*—FA[11]	89.4%
Yang *et al.*—LDA[11]	91.5%
Roth *et al.*[12]	94.1%
Proposed algorithm	98.97%

Table-2 shows the comparison of face detection for viola-jones detection and proposed algorithm.

Table 2- Comparison of Face part Detection.

Criteria	Viola& Jones face detection	Proposed algorithm
False Positive Rate	30.09346	0.954836
False Negative Rate	8.09346	0.073577
Accuracy	61.81308	98.97158

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 3-Average time consumed and faces detection accuracy of test images.

Average accuracy in image auto size	**Viola Jones**	**Hybrid design**
Average time consumed per image	1.467s	**s0.9067**
Face detection accuracy	60.12%	**% 90.89**
Eyes detection accuracy	61.81%	**98.97 %**

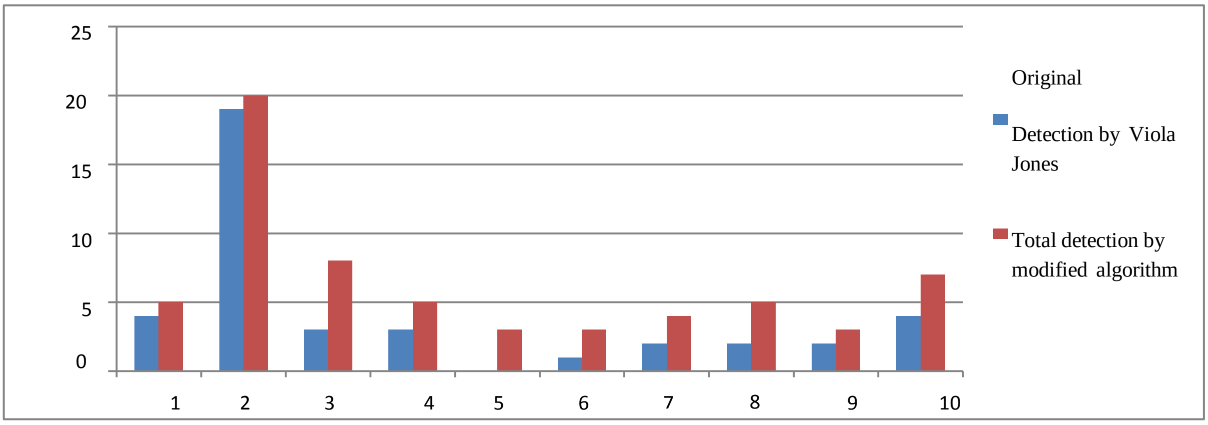

Figure 10- Graph depicting Faces detection based on eyes.

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 multi-scales. 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

1. Jones, M. and Viola, P. **2001**. Rapid Object Detection using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition. IEEE Computer Society Conference, **1**(1), pp: 511-518.
2. Himani, S. P., Darshak, G. T. and Udesang, K. Jal. **2014.** A Survey on Object Detection and Tracking Methods. *International Journal of Innovative. Research in Computer and Communication Engineering*, **2**(2): 6 -10.
3. Oualla, M., Sadiq, A. and Mbarki, S. **2015.** Comarative study of the methods using Haar-like feature. *International Journal of Engineering Sciences & Research Technology*, **4**(4): 1-9.
4. Kakumanu, P., Makrogiannis, S. and Bourbakis, N. **2007.** A survey of skin-color modeling and detection methods. *Pattern Recognition Journal*, **40**(3): 1106-1122.
5. Ming-Hsuan, Yang, David J. Kriegman and Narendra, Ahuja. **2002**. Detecting Faces in Images: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(1) : 34-58.
6. Alpika, Gupta, and Rajdev, Tiwari. **2015**. Face Detection Using Modified Viola Jones Algorithm**.** *International Journal of Recent Research in Mathematics Computer Science and Information Technology***,** **1**(2): 59-66.
7. Wang, S. Amr and Abdel-Dayem. **2012**. Improved Viola-Jones Face Detector. ICCIT-7th *International Conference on Computer Sciences and Convergence Information Technology*, **57**(2) : 124-126.
8. Lam, K. M., Siu, W.-C. and Kwok-Wai, Wong. **2004**. An efficient algorithm for human face detection and facial feature extraction under different conditions. *Pattem Recognition Society*, **34**(10): 1993-2004.
9. Di Huang, Caifeng Shan, Mohsen Ardebilian, Yunhong Wang, and Liming Chen. **2011**. Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. *IEEE Transctions On System*, **41**(6): 765-781.
10. Paul V., Michael J. Jones. **2004**. Robust Real-Time Face Detection. *International Journal of Computer Vision*, **57**(2): 137–154.
11. Yang, M.-H., Ahuja, N. and Kriegman, D. **2000**. Face detection using mixtures of linear subspaces. In Proceedings Fourth IEEE International Conference on Automatic, **24**(1), pp: 70-76.
12. Roth, D., Yang,M.-H. and Ahuja, N. **2000**. A Snow-based face detector Advances in Neural Information Processing Systems 12 (NIPS 12), **51**(1)، pp: 855-861.