AUTOMATIC THEFT SECURITY SYSTEM (SMART SURVEILLANCE CAMERA)

Veena G.S [1], Chandrika Prasad[2] and Khaleel K[3]

Department of Computer Science and Engineering, M.S.R.I.T,Bangalore, Karnataka
veenags@msrit.edu
chandrika@msrit.edu
khr.night@gmail.com

ABSTRACT

The proposed work aims to create a smart application camera, with the intention of eliminating the need for a human presence to detect any unwanted sinister activities, such as theft in this case. Spread among the campus, are certain valuable biometric identification systems at arbitrary locations. The application monitosr these systems (hereafter referred to as "object") using our smart camera system based on an OpenCV platform.

By using OpenCV Haar Training, employing the Viola-Jones algorithm implementation in OpenCV, we teach the machine to identify the object in environmental conditions. An added feature of face recognition is based on Principal Component Analysis (PCA) to generate Eigen Faces and the test images are verified by using distance based algorithm against the eigenfaces, like Euclidean distance algorithm or Mahalanobis Algorithm.

If the object is misplaced, or an unauthorized user is in the extreme vicinity of the object, an alarm signal is raised.

KEYWORDS

OpenCV, Haar Classifers, Viola-jones, PCA, Eigen Faces, Theft security, Surveillance.

1. INTRODUCTION

A scenario is considered where we are safeguarding a biometric device for authorized users which contains sensitive information. Device can be tampered or stolen by unauthorized users .Our smart camera system, eliminates the need for human surveillance and massive redundant storage of unnecessary data of mundane usual activity. It captures and retaliate only when certain subset of unwanted events occur such as tampering or theft.

The objective is to create a theft security system which can eliminate the redundancy involved with currently employed technology. We aim to empower the device to become a smart application with a basic sense of artificial intelligence. We attempt to store biometric identification of authorized users, maintain a constant monitoring environment for the object, identify the object and the authorized subject in question using machine learning principles. Our

Natarajan Meghanathan et al. (Eds) : ITCSE, ICDIP, ICAIT - 2013
pp. 75–87, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3908

system also identifies and simultaneously records and take appropriate action in retaliation to any mischievous conduct of unauthorized individuals like tampering and theft..

We make use of Open CV as the primary platform, on which said application is developed
With the advent of embedded systems and the increasing availability of technology based on computer vision, we can achieve this daunting task with ease. This is accomplished by training the system for simple patterns deduced to identify and isolate certain events in the environment

2. SYSTEM ARCHITECTURE DESIGN

The overall basic paradigm employed is as illustrated in the following steps:

- **Image acquisition** – A digital image is produced by one or several image sensors, in addition to
- various types of light-sensitive cameras.
- **Pre-processing** – Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is necessary to process the data in order to assure that it satisfies certain assumptions implied by the method.
- **Feature extraction** – Image features at various levels of complexity are extracted from the image data.
- **Detection/segmentation** – At some point in the processing, a decision is made about which image points or regions of the image are relevant for further processing.
- **High-level processing** – At this step the input is typically a small set of data, for example, a set of points or an image region which is assumed to contain a specific object.
- **Decision making -** Making the final decision required for the application, for example, pass/fail or match / no-match.

The following image depicts the flowchart representing the basic design that the system works under.

- **Monitoring**: as the system is put into working mode, its first instinct is to monitor its surroundings to analyze any motion occurring in the considered scenario. Motion analysis mainly aims at avoiding massive redundant storage of activity. As soon as the camera comes across some unknown entity approaching towards the object, the recording of the motion begins.

- **Motion Detection:** The surveillance system remains inactive, silently monitoring, till it notices an unknown entity approaching, soon after which the camera starts recording the scenario in consideration containing a clear view of the object. The training given to the system via Haar classifiers and Viola -Jones algorithm helps in keeping the system alert about the presence of the object that it has been trained to safeguard.

- **User recognition:** The next step, that is the most crucial, involves face recognition of the unknown user to recognize if it is an authorized user or an unauthorized one. Using Eigenfaces algorithm and Principal Component Analysis along with distance-based analysis, the system compares the test image (unknown user's image) with the Eigenfaces. using either Euclidean distance algorithm or Mahalanobis Algorithm and then if the face is from authorized user then face detection is done.

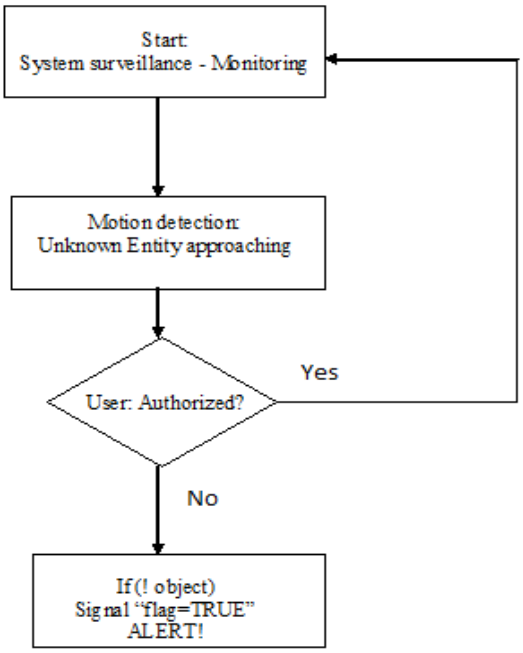

Fig 1: System Working Architecture

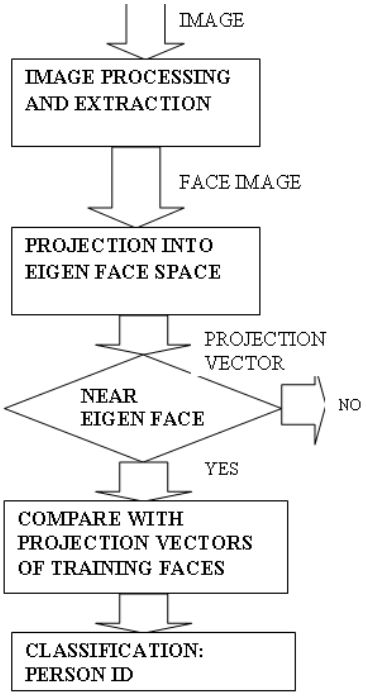

Fig 2: Face Recognition paradigm

The Fig 2 depicts the flowchart representing the face recognition paradigm:

- Grab a frame from the camera
- Convert the color frame to 78grayscale
- Detect a face within the 78grayscale camera frame
- Crop the frame to just extract the face region
- Preprocess the face image
- Recognise the person in the image

3.WORKING

OpenCV enables us to teach the machine using its machine learning algorithms to distinguish between various user use cases and unauthorized perpetrators' unusual unwanted activity in order to take appropriate action as per the environmental conditions (normal functions or retaliation). Using the image processing tactics and mathematical deductions, we are able to successfully implement logic to enact the artificial intelligence concept at hand to identify and classify the events that occur. In addition to that, the system is capable of taking action in accordance with the event taking place. All this is designed using a simple interface of C / C++ to make the most of the optimized libraries in OpenCV, thus combining prejudice with efficiency.Previously mentioned in the "Servo Magazine" 2007 edition, spread over a few months [2], we can find reasonable solutions to aforementioned objectives.

The **Viola–Jones object detection framework** [7][8] is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be used to identify a variety of different objects based upon its features, it primary focus was the problem of face detection. This algorithm is used in OpenCV as cvHaarDetectObjects(). Thus we can have an accurate working function based on this which will enable us to determine the presence of the object in the scenario as well as candidate faces.

The **Principal Component Analysis (PCA)** [10] is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. It basically enables us to determine possible candidate features in the human faces to assist the system in its daunting task of distinguishing one face from another. . The Eigenfaces approach is based on PCA, in which a small subset of characteristic pictures, are used to identify and ascertain the variation between face images.

Once trained to the features in question, we use the function to find Nearest Neighbour. The algorithm caches all of the known users' training samples, and predicts the response for a new input sample of an entity by analyzing a certain number of the nearest neighbours of the sample.

3.1 Viola-Jones Algorithm

The features employed by the detection framework universally involve the sums of image pixels within rectangular areas. However, since the features used by Viola and Jones all rely on more than one rectangular area, they are generally more complex. With the use of an image representation called the integral image (A summed area table, also known as an integral image, is a data structure and algorithm for quickly and efficiently generating the sum of values in a rectangular subset of a grid).

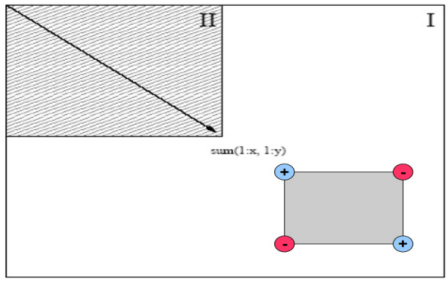

Fig 3: Integral Image Computation

Rectangular features can be evaluated in constant time, which gives them a considerable speed advantage over their more sophisticated relatives. The evaluation of the strong classifiers generated by the learning process isn't fast enough to run in real-time. A cascade ordering is used in order of their complexity, where each classifier stronger than its predecessor is trained only on the select candidates which have passed the classifiers prior to them. If at any stage in the cascade a classifier rejects the sub-window under inspection, no further processing is performed and continues on searching the next sub-window.

AdaBoost is a machine learning boosting algorithm used in Viola-Jones, capable of constructing a strong classifier through a weighted combination of weak classifiers. (A weak classifier classifies correctly in only a little bit more than half the cases.)

A weak classifier is mathematically described as:

$$h(x, f, p, \theta) = 1 \, , if \; pf(x) > p\theta \; ; 0 \, , Otherwise$$

Where x is a 24*24 pixel sub-window, f is the applied feature, p the polarity and θ the threshold that decides whether X should be classified as a positive (a face) or a negative (a non-face).

Viola-Jones' modified AdaBoost algorithm is presented in pseudo code as follows:

- Given examples images (x1,y1),...,(xn,yn) where y1=0,1 for negative and positive examples.
- Initialize weights $w_{1,i}$ = 1/2m , 1/2l , for y1=0, 1, where m and l are the numbers of positive and negative examples.

- For t=1 … T:

1) Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

2) Select the best weak classifier with respect to the weighted error:

$$\varepsilon_t = min_{f,p,\theta} \sum_i w_i \; | h(x_i, f, p, \theta) - y_i |$$

3) Define h_t (x) = h(x, f_t, p_t, θ_t) where f_t, p_t and θ_t are the minimizers of €$_t$.

4) Update the weights:

$$w_{t+1,i} = w_{t,i}\,\beta^{1-\varepsilon_i}$$

Where $e_i = 0$, if example x_i is classified correctly and $e_i = 1$ otherwise, and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$

- The final strong classifier is:

$$C(x) = 1, \quad if \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{T} \alpha_t \; ; \; 0, \quad otherwise$$

Where $\alpha_t = \log(1/\beta_t)$

3.2 Face Recognition – PCA and Eigen faces.

Principal component analysis (PCA), based on information theory concepts, seeks a computational model that best describes a face by extracting the most relevant information contained in that face. In the Eigen faces approach, which is based on PCA, we describe the variations between face images based upon a small set of characteristics. The goal is to find the eigenvectors (Eigen faces) of the covariance matrix of the distribution, spanned by training a set of face images. Later, every face image is represented by a linear combination of these eigenvectors. We project the new face image onto the Eigen face spanned subspace and then classifying it by comparing relative positions in the face space against those positions of known persons, thus performing recognition.

Principal Component Analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components, where, the number of principal cannot exceed the number of original variables. It is equal to or lesser than that. This transformation defines the first principal component to have the largest possible variance.

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. Equivalently, a matrix Q is orthogonal if its transpose is equal to its inverse where I is the identity matrix.

$$\mathbf{Q^T Q = Q Q^T = I}$$

A data matrix, $\mathbf{X}^T$; The singular value decomposition of $\mathbf{X}$ is $\mathbf{X = W\Sigma V^T}$, where the $m \times m$ matrix $\mathbf{W}$ is the matrix of eigenvectors of the covariance matrix $\mathbf{XX}^T$, the matrix $\mathbf{\Sigma}$ is an $m \times n$ rectangular diagonal matrix with nonnegative real numbers on the diagonal, and the $n \times n$ matrix $\mathbf{V}$ is the matrix of eigenvectors of $\mathbf{X}^T\mathbf{X}$. The PCA transformation that preserves dimensionality (that is, gives the same number of principal components as original variables) $\mathbf{Y}$ is then given by:

$$\begin{aligned} Y^T &= \mathbf{X^T W} \\ &= (\mathbf{W \sum V^T})^T \mathbf{W} \\ &= \mathbf{V \sum^T W^T W} \\ &= \mathbf{V \sum^T} \end{aligned}$$

The Pseudo code for PCA is as per the following steps:

PCA1: Perform PCA using covariance.

- Subtract off the mean for each dimension
- Calculate the covariance matrix
- Find the eigenvectors and Eigen values
- Extract diagonal of matrix as vector
- Sort the variances in decreasing order
- Project the original data set

We use PCA to convert all your hundreds of training images into a set of "Eigen faces" that represent the main differences between the training images. Initially, the mean value of each pixel is computed to create the average face image. Subsequent Eigen Faces are calculated and formed in comparison with respect to the average face formed prior to this. Where the first eigenface is the most dominant face differences and the second eigenface is the second most dominant face differences, and so on, until you have about 50 eigenfaces that represent most of the differences in all the training set images.

Fig 4: Average Face Image

Given face images for each of several people, plus an unknown face image to recognize,

1. Compute a "distance" between the new image and each of the example faces
2. Select the example image that's closest to the new one as the most likely known person
3. We set a threshold, according to which if the distance to that face image is greater than the threshold, we “recognize the image to a person ID; else identify that face as an unknown entity.

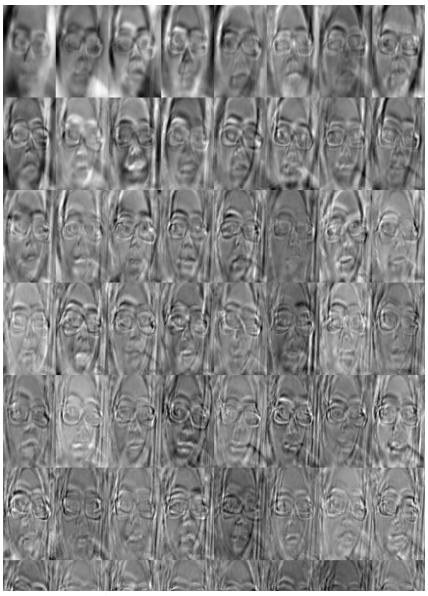

Fig 5: Sequence of Eigen Faces

Distance-based matching method in Eigenface algorithm can be:

- Euclidean distance based algorithm
- Mahalanobis distance based algorithm

Euclidean algorithm: When the distance, in the original Eigen face paper, is measured as the point-to-point distance it is also called Euclidean distance.

It is performed mathematically as “find nearest neighbour” using the Euclidean Distance function. It checks the similarity quotient between the input image and the respective training images, and finds the most resembling image by gauging the least distance in the Euclidean Space.

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given according to the Pythagoras Theorem. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space.

The Euclidean distance between points p and q is the length of the line segment connecting them (PQ).

In general, for an n-dimensional space, the distance is:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_i - q_i)^2 + \ldots + (p_n - q_n)^2}.$$

Mahalanobis algorithm: Mahalanobis distance is a distance measure which is based on correlations between variables by which different patterns can be identified and analyzed.It gauges *similarity* of an unknown sample set to a known one.It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant.

Formally, the Mahalanobis distance of a multivariate vector [x=(x1,x2…,xn)^T], from a group of values with mean [u=(u1,u2,….,un)^T] and covariance matrix S is defined as:

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}.$$

4. IMPLEMENTATION

Implementation is done in OpenCV using the simplest functions so as to keep efficiency and simplicity as our top priorities. We have an input sky cam feed which shows us the area as displayed. At this point we attempt to monitor the object of interest (in red). This is accomplished by by drawing a cvRect around the detected object, as determined by the previously trained Haar Classifier database created by us. This is focused upon on a particular region with minor allowances for variations in the setup. Since there is only one object in question, we can set the cv return biggest object flag on to increase accuracy. When the perpetrator lifts and steals the object, the system is aware that the event of theft has occurred.

The alarm is raised if the object of interest is missing for more than a threshold of frames, say let x=10. So if the object is missing from its place for 10 frames (approx 2 seconds at 5fps) an alarm is raised.

The pseudo code is as follows:

```
cvSetImageROI(img, cvRect());
cropped = cvCreateImage( cvSize(), img->depth, img->nChannels );
object = cvHaarDetectObjects( cropped_img, cascade_classifier, storage, scale factor, min
neighbour,  cvSize(),cvSize() );
   for( all objects )
      {
CvRect *r = ( CvRect* )cvGetSeqElem( object, i );
cvRectangle( img, cvPoint( r->x, r->y ), cvPoint( r->x + r->width, r->y + r->height), CV_RGB);
}
frameid++; //Next frame.
cvResetImageROI(img);
```

Followed by the preprocessing and focusing, we now execute our recognition algorithm as mentioned previously. For a single input image, it is compared against every training image, and against every eigenface of every training image (generated using PCA function during database creation) to be more precise. The datatypes store the number of eigenfaces, training images, names of entities, and other details.

After the comparisons, we flag and store the training image with the least square value generated as a result from the Euclidean / Mahalanobis distance generated. Using the function find Nearest Neighbour we find the closest matching neighbor for the given instance of the input image. The probability confidence of the authorized user is anywhere above 75%, and unauthorized user is much below that.

The pseudo code is as follows:

```
for ( all training images)
        {
        for(all eigenface images)
        {
        distSq = distance square using     Euclidean / Mahalanobis.
        }

        if(distSq < least_distance_Sq)
        {
        Least_distance_sq = distSq;
        nearest_img = train_img;
        }
        }
  if( confidence > 0.60 )
        print: " person_id"
  else
        print: "unauthorized entity. Alert. Save data."
cvSaveImage(img);
```

5. PERFORMANCE

Out of Approximately 57 valid images of human faces, users or otherwise, 47 were successfully detected and recognized. 10 candidates failed in detection itself. This gives us an 82% success rate.

Similarly we have 100% detection in phase 1 detection of the valuable object. With all instances of the object being detected and spotted while it is in its operational position.

It successfully passes three test cases to recognize certain events such as:

- Check for detection of object of interest. It should track / detect the object of interest in the input environment. If the camera detects the object irrespective of presence of other entities then PASS else FAIL.

- Check for detection of object of interest with the presence of an unknown, potentially unwanted entity. It should track / detect the object of interest in the input environment for as long as the object is in place. Else raise alarm if object is missing. If the camera detects the object is missing irrespective of presence of other entities then PASS else FAIL.

- Check for presence of authorized users in the vicinity of object. It should be able to differentiate between authorized and unauthorized users. No action taken for authorized users' operation. An alarm signal should be raised and store the surveillance data from different aspects. If the camera distinguishes between authorized and unauthorized users in their presence then PASS else FAIL.

The confidence value suggests the probabilistic estimate of the surety with which the system can ascertain and identify who the person in the input image is. It is basically a measure of the accuracy of recognition. Of all the training images in the database, we pick the one which has the least mean squared value (of the feature analysis performed earlier in the find Nearest Neighbor function based on the PCA). The confidence is directly proportional to the root mean square (basically the square root) of the same, which is used in the further computations. The database image with the closest match to the input image is found and correspondingly noted.

Table 1: Performance values of authorized user

Confidence	Least mean value squared	Root mean square	Nearest Image / Training Images
0.816357	89373218049.483398	777.513111	5 / 385
0.868392	98758815105.867645	817.319684	5 / 385
0.881817	101256386579.118423	827.590003	5 / 385
0.856998	96663601080.892471	808.603298	5 / 385
0.814468	89041272774.036575	776.067868	5 / 385
0.741031	76616554546.688766	719.888659	5 / 385
0.797728	86126957102.629959	763.261884	5 / 385
0.83599	92859365173.946091	792.532130	56 / 385
0.858796	96992688444.877335	809.978557	5 / 385
0.84961	95317124764.295609	802.951825	5 / 385
0.816556	89408266541.237198	777.665550	5 / 385
0.830606	91896742457.726105	788.413553	5 / 385
0.76748	80984006719.728073	740.122578	5 / 385

0.773221	81947945043.508011	744.514322	5 / 385
0.803165	87068233614.200577	767.421372	5 / 385
0.739743	76406956631.454453	718.903295	5 / 385
0.720257	73271158266.812729	703.996583	4 / 385
0.702008	70394065766.955856	690.036475	5 / 385

Based on further calculations, we can deduce the average confidence accuracy value of the authorized user "user1" determined to be = 0.8041235. The higher the rank of iNearest, the more accurate recognition is achieved with a high probability confidence in the user image face as authorized.

The unauthorized user is primarily distinguished by the seemingly far distance measurement from the accurate records of user images in the user database. As we know, the Eigen images after 25 consist mostly of noise.

Table 2: Performance values of unauthorized entity

Confidence	Least mean value squared	Root mean square	Nearest Image
0.426425	33950985323.859634	479.214791	30
0.504853	42984472232.067719	539.212340	30
0.459102	37585433288.927147	504.212727	30
0.426391	33947378096.328480	479.189333	30
0.420855	33349996057.181774	474.954401	4
0.368939	28005650092.943787	435.238046	4
0.40244	31400940759.230412	460.866725	4
0.405481	31718761766.491341	463.193158	30
0.40944	32134843782.965813	466.221309	4
0.443883	35869752386.690361	492.570289	4
0.446891	36205663023.469536	494.871309	4
0.475075	39429325361.242966	516.432656	30
0.48653	40778719879.825264	525.195283	172
0.458798	37550807119.145470	503.980417	4
0.500954	42510284302.949623	536.229902	5
0.46663	38448928075.062569	509.971776	30
0.493445	41604422277.093262	530.485809	30
0.534615	46691112116.129532	561.980352	30
0.532205	46385315590.337914	560.137026	30
0.529038	45984919976.211121	557.714249	30
0.544037	47896504418.848640	569.188258	30
0.56983	51274894472.403275	588.920145	30
0.553742	49154125206.362709	576.612441	30
0.513716	44072349539.750275	545.993063	30
0.489118	41086710886.496712	527.174884	30
0.454007	37006661947.937645	500.315525	205
0.435567	34949194299.763756	486.208567	205
0.399314	31075943752.018639	458.475557	205

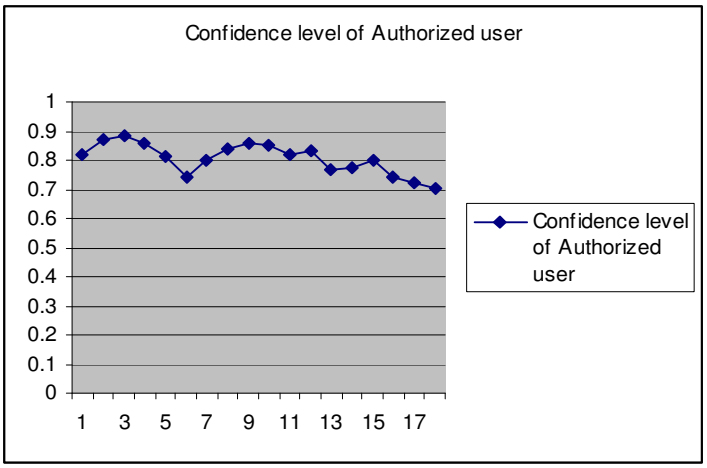

Fig 6:Confidence Level of Authorized users.

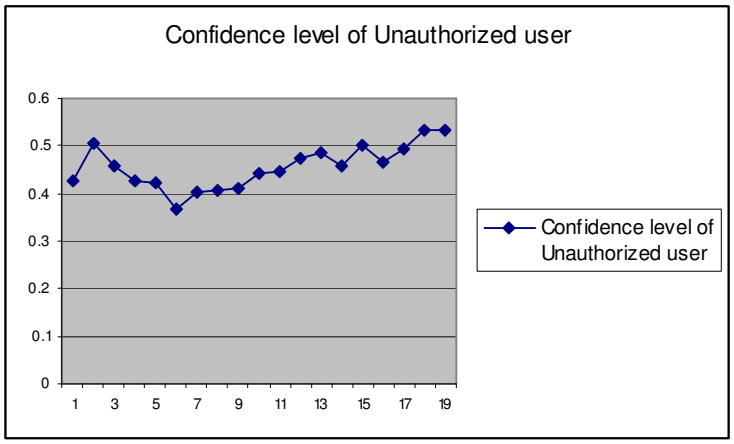

Fig 7: Confidence Level of Unauthorized users

6. CONCLUSION

As per our expectations we can deliver a security system to eliminate the redundancies of normal operations of obsolete systems around the world. We are able to achieve optimal levels of accuracy and efficiency with the system delivered as anticipated.

REFERENCES

[1] Learning OpenCV – O'Reilly
[2] http://www.cognotics.com/opencv/servo_2007_series/ A reference to an article in Servo Magazine
[3] http://opencv.willowgarage.com
[4] Wikipedia.org
[5] Coursera on CV - /www.coursera.org/course/vision
[6] CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision.
[7] http://www.docstoc.com/docs/113503439/Robust-Real-time-Object-Detection-by-Paul-Viola-and-Michael-Jones
[8] http://research.microsoft.com/en-us/um/people/viola/Pubs/Detect/violaJones_CVPR 2001.pdf

[9] Implementing the Viola-Jones Face Detection Algorithm, by Ole Helvig Jensen, Kongens Lyngby 2008 IMM-M.Sc.-2008-93. Technical University of Denmark.
[10] An Introduction to Principal Component Analysis, Greg Coombe.
[11] Face Localization via Shape Statistics By M.C. Burl, T.K. Leung, and, P. Perona, Face localization via shape statistics. Int. Workshop on Automatic Face and Gesture Recognition, Zurich, June 1995.
[12] Neural Network-based Face Detection By Rowley, Baluja and Kanade. PAMI, January 1998.