1. INTRODUCTION

1.1 OBJECTIVE

The main purpose of the system is to provide citizens with safe and secure environment and reduce the manual work done by providing an integrated security system capable of tracking any malicious or threatening activity.

1.2 PROBLEM STATEMENT

Creation of a smart, integrated surveillance system that will attempt to track locations of and uniquely identify users across the system. This will help in the prevention of threats or malicious elements as well as their live monitoring.

To create a smart surveillance system that takes CCTV footage as input and recognizes wanted criminals.

1.3 METHODOLOGY USED

Image processing using MATLAB.

2. LITERATURE SURVEY

A survey was carried out in order to explore features which would prove useful for tracking criminals. An extensive questionnaire was prepared which addressed the vital problems federal system and to an extent civilians faced. At the same time, feedbacks were taken on the solutions to these problems. The survey revealed that:

- An integrated security system will help the federal system to track various absconding criminals and quick actions can be taken to forestall any threat.
- An online database will help the police department to segregate the criminal records without any loss due to manual misplacement. It would also reduce tedious paperwork.
- Extra feature of number plate detection can be used to detect missing vehicles in a city.

Smart Surveillance

CCTV cameras have become ubiquitous, nowhere more so than in the United Kingdom. So far, the value of CCTV surveillance has failed to live up to the hype. A recent report by the Metropolitan Police in London revealed that the city's one million plus cameras have helped to solve only a handful of crimes. The criminologist Clive Norris points out that CCTV operators tend to act on their prejudices (for example, focusing cameras on people because of their skin colour) or merely on scenes which they find entertaining, to relieve the boredom of staring at mundane street scenes all day. So far, the main CCTV success stories are for forensic use and as a deterrent against some forms of petty or opportunistic crime. CCTV surveillance is often justified on the basis that it can be used to prevent terrorist attacks, but the reality is somewhat more banal. As points out, those committing acts of terrorism are usually unknown before the act takes place, and it is very difficult to conduct surveillance of someone whose identity is unknown. There are few examples of CCTV surveillance being used effectively for real-time detection or prevention of crime. It has been suggested that the effectiveness of CCTV surveillance can be improved by tracking known individuals as they move through a surveillance network. One possible application could be to follow known hooligans as they enter and move about a football stadium. This paper will consider the design of a video surveillance network which could track the movement of known individuals in real time.

Applications of Smart Surveillance

Law enforcement and justice solutions

• Today's law enforcement agencies are looking for innovative technologies to help them stay one step ahead of the world's ever-advancing criminals.

• As such, FRS is committed to developing technologies that can make the jobs of the law enforcement officer easier. This includes acclaimed CABS-computerized arrest and booking system and the child-base protection, a software solution for global law enforcement agencies to help protect and recover missing and sexually exploited children, particularly as it relates to child pornography.

Security

• Store all offence-related detain one easy-to-use system -- data is entered once and only once.

• Integrate with any database -- including other detachments and other applications (RMS, CAD, Jail Management systems, and "most-wanted" databases).

• Link victims to offenders -- to aid in criminal analysis and investigations

• Capture and store digital images of the offender -- encode all mug shots, marks, tattoos, and scars

• Perform rapid and accurate searches -- on all data and image fields for crime statistics and reporting

• Produce digital lineups -- using any stored image in minutes

• Identify previous offenders -- pre-integrated with advanced biometric face recognition software.

ChildBase Protection

ChildBase is an application that helps protect and recover missing and sexually-exploited children, particularly those children victimized through child abuse images.

Identification solutions

With regards to primary identification documents, (Passports, Driver's licenses, and ID Cards), the use of face recognition for identification programs has several advantages over other biometric technologies.

• Leverage your existing identification infrastructure. This includes, using existing photo databases and the existing enrolment technology (e.g. cameras and capture stations);

• Increase the public's cooperation by using a process (taking a picture of one's face) that is already accepted and expected;

• Integrate with terrorist watch lists, including regional, national, and international "most-wanted" databases.

Homeland Defense

• Since the terrorist events of September 11, 2001, the world has paid much more attention to the idea of Homeland Defense, and both governments and private industries alike are committed to the cause of national defense.

• This includes everything from preventing terrorists from boarding aircraft, to protecting critical infrastructure from attack or tampering (e.g. dams, bridges, water reservoirs, energy plants, etc.), to the identification of known terrorists.

Airport Security

• Airport and other transportation terminal security is not a new thing. People have long had to pass through metal detectors before they boarded a plane, been subject to questioning by security personnel, and restricted from entering "secure" areas. What has change, is the vigilance in which these security efforts are being applied.

• The use of biometric identification, can enhance security efforts already underway at most airports and other major transportation hubs (seaports, train stations, etc.).

• This includes the identification of known terrorists before they get onto an airplane or into a secure location.

Immigration

• Most countries do not want to be perceived as being a "weak link" when it comes to accepting immigrants and refugees, particularly if that individual uses the new country as a

staging ground for multi-national criminal and terrorist activities. Consequently, governments around the world are examining their immigration policies and procedures.

• Biometric technology, particularly face recognition software, can enhance the effectiveness of immigration and customs personnel. After all, to the human eye it is often difficult to determine a person's identity by looking at a photo, especially if the person has aged, is of a different ethnic background, has altered their hair style, shaved their beard, etc. FRS does not have this difficulty.

Access Control

• The use of biometric technology, particularly face recognition software (either independently or as one part of a multi-layered biometric solution), can enhance your security efforts considerably.

• Biometric identification ensures that a person is who they claim to be, eliminating any worry of someone using illicitly obtained keys or access cards.

Financial Services

• The financial services industry revolves around the concept of security. Yet for the most part, security within the industry is limited to a simple personal identification number (PIN) or password.

• Biometrics, particularly face recognition software, can improve the security of the financial services industry, saving the institution time and money both through a reduction of fraud cases and the administration expenses of dealing with forgotten passwords.

• Furthermore, biometric-based access control units can safeguard vaults, teller areas, and safety deposit boxes to protect against theft.

• The use of biometrics can also ensure that confidential information remains confidential while deterring identity theft, particularly as it relates to ATM terminals and card-not-present e-commerce transactions.

Scene analysis and surveillance solutions

• This includes the ability to extract, categorize, and search non-facial imagery. For example, within the law enforcement application it allows you to capture, archive, and retrieve such identifying characteristics as tattoos, marks, or scars.

• It can also analyse scenes from either streaming or archived video, "looking" for out-of-the-ordinary occurrences, the presence of certain vehicles, specific faces, etc.[6]

• This is beneficial and can save significant time and money to those individuals who spend hours, days, or weeks monitoring video streams (i.e. examining a bank's security in a criminal investigation).

Existing Systems

digiKam

digiKam is a free and open-source image organizer and tag editor written in C++ utilizing the KDE Platform. digiKam runs on most known desktop environments and window managers, as long as the required libraries are installed. It supports all major image file formats, and can organize collections of photographs in directory-based albums, or dynamic albums by date, timeline, or by tags. Users can also add captions and ratings to their images, search through them and save searches for later use. Using plug-ins, users can export albums to various online services including (among others) 23hq, Facebook, Flickr, Gallery2, Google Earth's KML files, Yandex.Fotki, MediaWiki, Rajce, SmugMug, Piwigo, Simpleviewer,Picasa Web Albums. Plug-ins are also available to enable burning photos to a CD and the creation of web galleries.

digiKam provides functions for organizing, previewing, downloading and/or deleting images from digital cameras. Basic auto-transformations can also be deployed on the fly during picture downloading. In addition, digiKam offers image enhancement tools through its KIPI (KDE Image Plugins Interface) framework and its own plugins, like red-eye removal, color management, image filters, or special effects. digiKam is the only free photo management application on Linux that can handle 16 bit/channel images. Digital Asset Management is the mainstay of digiKam.

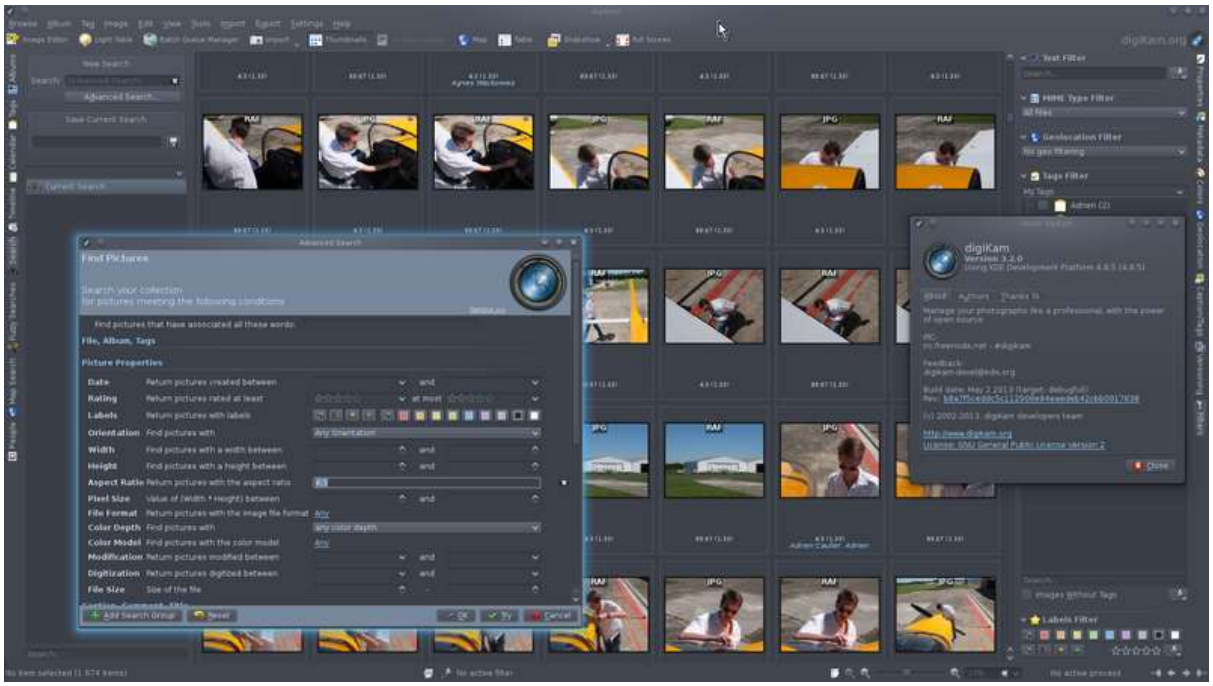

Figure 1: digiKam 3.2.0 on Linux.

iPhoto

iPhoto is a digital photograph manipulation software application developed by Apple Inc. It has been included with every Macintosh personal computer since 2002, originally as part of the iLife suite of digital media management applications. iPhoto can import, organize, edit, print and share digital photos.

iPhoto is often compared to Google's Picasa, CyberLink's MediaShow, Adobe's Photoshop Album, Phase One's Media Pro and Microsoft's Windows Photo Gallery. iPhoto '11 (9.5), the latest version of the software, was released as part of the iLife '11 suite on October 20, 2010.

On March 7, 2012, Apple CEO Tim Cook announced a new, iOS-native version of iPhoto alongside the third-generation iPad.

OS X

Version	iLife	Introduction	OS X	Binary
iPhoto 1	–	January 7, 2002[1]	10.1	PowerPC
iPhoto 2	iLife	January 3, 2003	10.1	PowerPC
iPhoto 4	iLife '04	January 6, 2004	10.2.6	PowerPC
iPhoto 5	iLife '05	January 11, 2005	10.3.4	PowerPC
iPhoto 6	iLife '06	January 10, 2006	10.4.3	Universal
iPhoto 7	iLife '08	August 7, 2007	10.4.9	Universal
iPhoto 8	iLife '09	January 7, 2009	10.5.6	Universal
iPhoto 9	iLife '11	October 20, 2010	10.6.3	Intel (32-bit)
iPhoto 9.5	–	October 22, 2013	10.9	Intel (64-bit)

Table 1: Versions of OS X

iOS

Version	Introduction	Ios
iPhoto for iOS 1.0	March 7, 2012	5.1
iPhoto for iOS 1.1	September 19, 2012	6.0

Table 2: Versions of iOS

Figure 2: iLife '11

OpenCV

OpenCV (*Open Source Computer Vision*) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel, and now supported by Willow Garage and Itseez. It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on real-time image processing. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself.

OpenCV runs on Windows, Android, Maemo, FreeBSD, OpenBSD, iOS, BlackBerry 10, Linux and OS X. The user can get official releases from SourceForge, or take the current snapshot under SVN from there. OpenCV uses CMake.

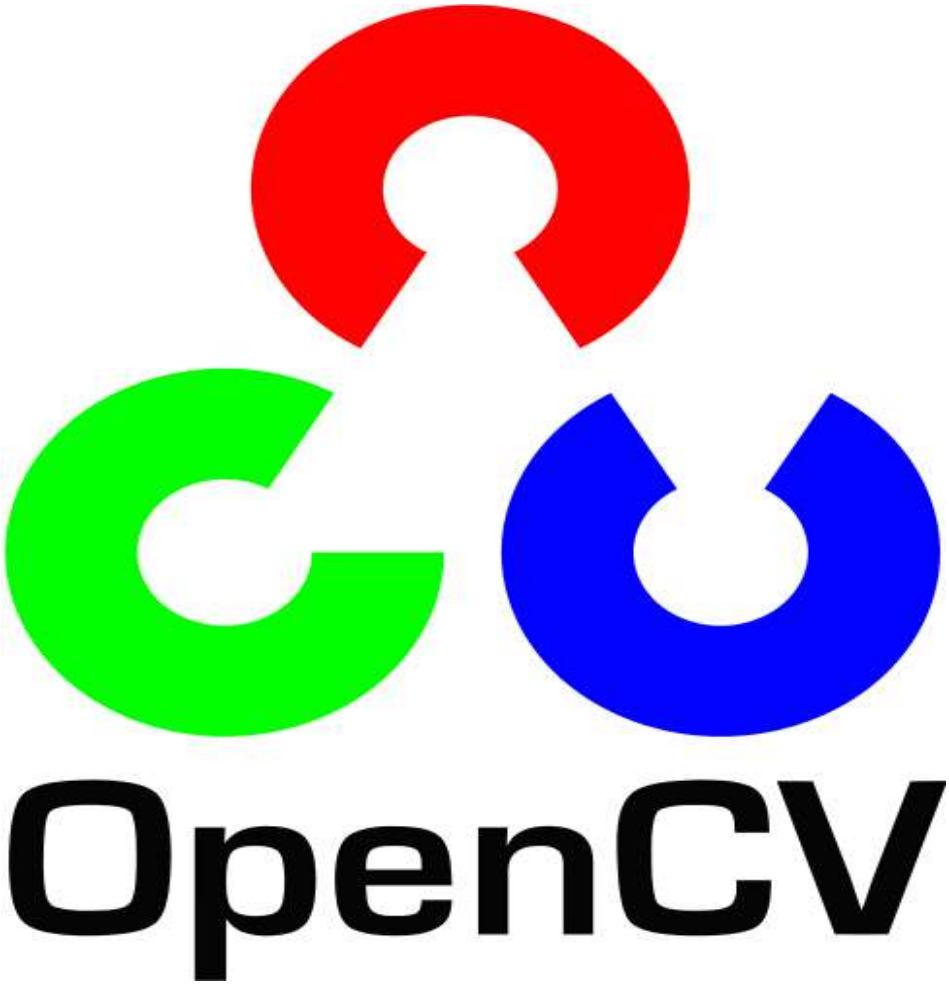

Figure 3: The OpenCV logo

Basic Software Based Surveillance Technology

Most techniques use the following procedure for performance evaluation:

1. Database creation- A database of criminals is created. This database consists of attributes such as basic criminal details eg.name, location, age, sex, height etc. and/or criminal photographs and sketches.

2. Setting up CCTV networks- If the software aims at tracking the location of criminals, CCTV cameras being used should form a network so that mobile criminals can be hunted down.

3. CCTV capture- The CCTV footage is captured and images from the footage are stored.

4. Comparison Processing - The CCTV images are then compared to the database. If a match occurs, an alert is generated.

5. If match is not obtained, the process is carried on by capturing images after a given amount of time.[6]

Unfortunately it is not possible now, nor will it be possible in the foreseeable future to make a computing machine that actually 'understands' what is sees. The level of vision and understanding which is instinctive to us (humans) is still far out of the reach of our silicon creations.

The ability to understand that the above image is not just a collection of pixels but is of a camouflaged frog on a log and to be able to identify exactly where the frog ends and log begins on the image is truly incredible. The fact that half of a primates cerebral cortex is dedicated to visual processing underlies the difficulty of this task (Zeki, 1993). This faculty is a result of millions of years of evolution and it would be naïve to think that we can enable computers to perform similar tasks.

But technically, why are computer vision problems so hard to solve? After all, while laudable results have been obtained in other artificial intelligence areas such as natural language processing, game theory, forecasting, control and even speech processing, computer vision seems to have lagged behind.

Major Issues in Parametric Approach

In the parametric approach, some major research issues are:

• How many images per criminal are sufficient for database?

• Do the images require some transformation before computing the features, for example resizing, deslanting or smoothing?

• Would the reference images be updated regularly? If yes, how would this is done?

• What is the database capacity? How many criminal features can it store?

Major Issues in Functional Approach

In the functional approach, the following major research issues arise:

• How is the comparison to be carried out given that images of an individual tend to have some variation amongst them?

• How fast will the recognition be?

Algorithms Used

Viola-Jones Object Detection Algorithm

The Viola–Jones object detection framework 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 trained to detect a variety of object classes, it was motivated primarily by the problem of face detection. This algorithm is implemented in OpenCV as cvHaarDetectObjects().

- Three major contributions/phases of the algorithm:
  - Feature extraction
  - Classification using boosting
  - Multi-scale detection algorithm
- Feature extraction and feature evaluation.
  - Rectangular features are used, with a new image representation their calculation is very fast.
- Classifier training and feature selection using a slight variation of a method called AdaBoost.
- A combination of simple classifiers is very effective

Feature types and evaluation

The features employed by the detection framework universally involve the sums of image pixels within rectangular areas. As such, they bear some resemblance to Haar basis functions, which have been used previously in the realm of image-based object detection. However, since the features used by Viola and Jones all rely on more than one rectangular area, they are generally more complex. The figure at right illustrates the four different types of features used in the framework. The value of any given feature is always simply the sum of the pixels within clear rectangles subtracted from the sum of the pixels within shaded rectangles. As is to be expected, rectangular features of this sort are rather primitive when compared to alternatives such as steerable filters. Although they are sensitive to vertical and horizontal features, their feedback is considerably coarser. However, with the use of an image representation called the integral image, rectangular features can be evaluated in *constant* time, which gives them a considerable speed advantage over their more sophisticated relatives. Because each rectangular area in a feature is always adjacent to at

least one other rectangle, it follows that any two-rectangle feature can be computed in six array references, any three-rectangle feature in eight, and any four-rectangle feature in just nine.

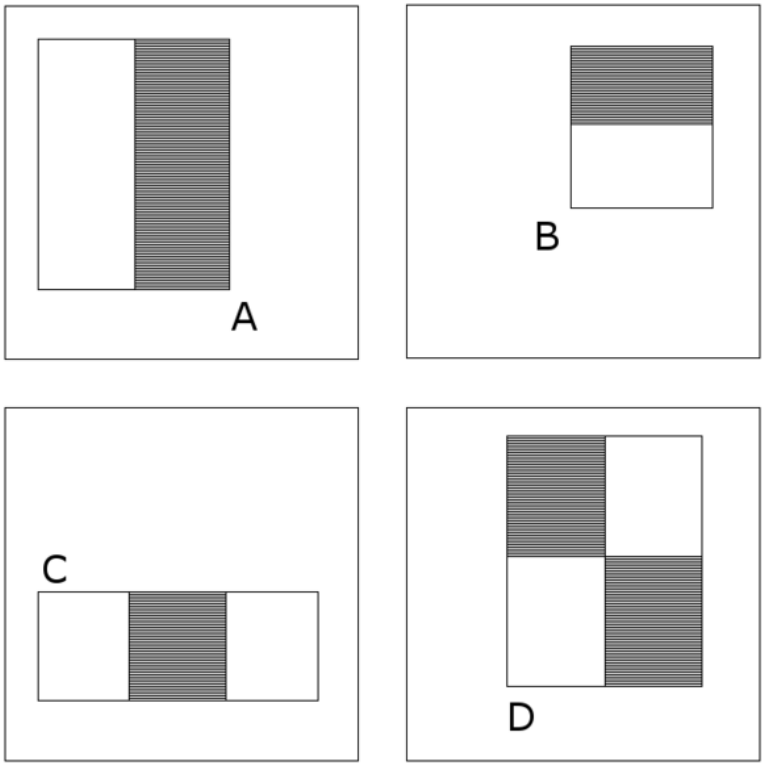

Figure 4: Example rectangle features shown relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature.

Learning algorithm

The speed with which features may be evaluated does not adequately compensate for their number, however. For example, in a standard 24x24 pixel sub-window, there are a total of 162,336 possible features, and it would be prohibitively expensive to evaluate them all. Thus, the object detection framework employs a variant of the learning algorithm AdaBoost to both select the best features and to train classifiers that use them.

Cascade architecture

The evaluation of the strong classifiers generated by the learning process can be done quickly, but it isn't fast enough to run in real-time. For this reason, the strong classifiers are

arranged in a cascade in order of complexity, where each successive classifier is trained only on those selected samples which pass through the preceding classifiers. If at any stage in the cascade a classifier rejects the sub-window under inspection, no further processing is performed and continued on searching the next sub-window. The cascade therefore has the form of a degenerate tree. In the case of faces, the first classifier in the cascaded called the attentional operator uses only two features to achieve a false negative rate of approximately 0% and a false positive rate of 40%. The effect of this single classifier is to reduce by roughly half the number of times the entire cascade is evaluated.

The cascade architecture has interesting implications for the performance of the individual classifiers. Because the activation of each classifier depends entirely on the behaviour of its predecessor, the false positive rate for an entire cascade is:

$$F = \prod_{i=1}^{K} f_i.$$

Similarly, the detection rate is:

$$D = \prod_{i=1}^{K} d_i.$$

Thus, to match the false positive rates typically achieved by other detectors, each classifier can get away with having surprisingly poor performance. For example, for a 32-stage cascade to achieve a false positive rate of 10^{-6}, each classifier need only achieve a false positive rate of about 65%. At the same time, however, each classifier needs to be exceptionally capable if it is to achieve adequate detection rates. For example, to achieve a detection rate of about 90%, each classifier in the aforementioned cascade needs to achieve a detection rate of approximately 99.7%.[4]

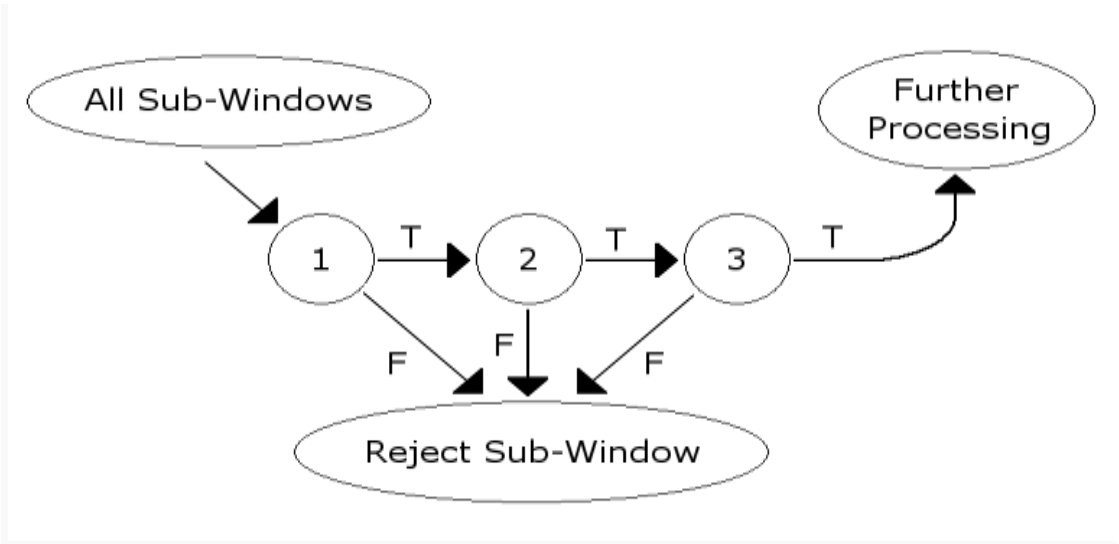

Figure 5: Cascade architecture used by Viola and Jones in their real-time object detection framework.

KLT Algorithm

Detect a Face

First, we must detect the face. We use the vision.CascadeObjectDetector System object to detect the location of a face in a video frame. The cascade object detector uses the Viola-Jones detection algorithm and a trained classification model for detection. By default, the detector is configured to detect faces, but it can be used to detect other types of objects.

```
% Create a cascade detector object.
faceDetector = vision.CascadeObjectDetector();

% Read a video frame and run the face detector.
videoFileReader = vision.VideoFileReader('tilted_face.avi');
videoFrame      = step(videoFileReader);
bbox            = step(faceDetector, videoFrame);

% Convert the first box to a polygon.
% This is needed to be able to visualize the rotation of the object.
x = bbox(1, 1); y = bbox(1, 2); w = bbox(1, 3); h = bbox(1, 4);
bboxPolygon = [x, y, x+w, y, x+w, y+h, x, y+h];

% Draw the returned bounding box around the detected face.
videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon);

figure; imshow(videoFrame); title('Detected face');
```

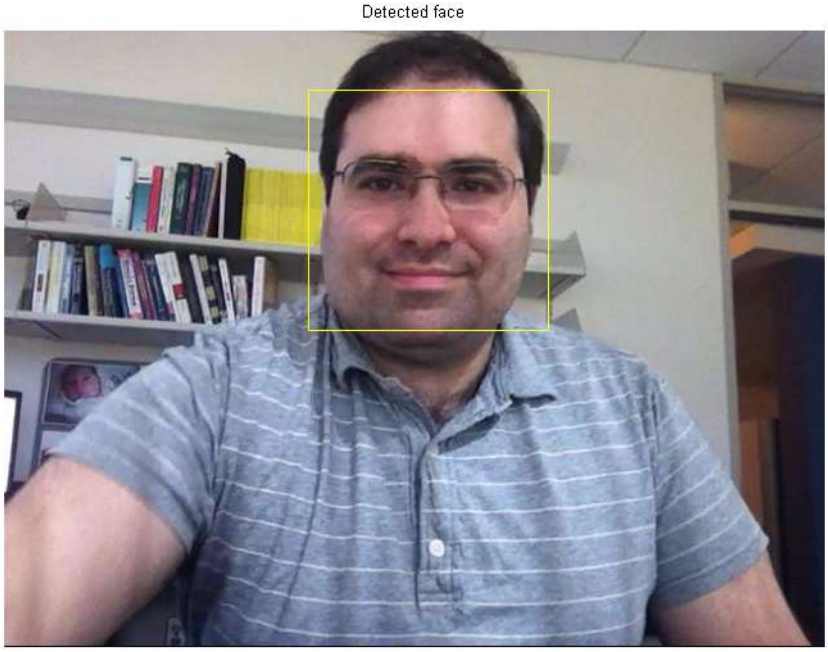

Figure 6: Face detection using KLT algorithm

To track the face over time, this example uses the Kanade-Lucas-Tomasi (KLT) algorithm. While it is possible to use the cascade object detector on every frame, it is computationally expensive. It may also fail to detect the face, when the subject turns or tilts his head. This limitation comes from the type of trained classification model used for detection. The example detects the face only once, and then the KLT algorithm tracks the face across the video frames.

Initialize a Tracker to Track the Points

With the feature points identified, you can now use the vision.PointTracker System object to track them. For each point in the previous frame, the point tracker attempts to find the corresponding point in the current frame. Then the estimateGeometricTransform function is used to estimate the translation, rotation, and scale between the old points and the new points. This transformation is applied to the bounding box around the face.

```
% Create a point tracker and enable the bidirectional error constraint to
% make it more robust in the presence of noise and clutter.
pointTracker = vision.PointTracker('MaxBidirectionalError', 2);

% Initialize the tracker with the initial point locations and the initial
% video frame.
points = points.Location;
initialize(pointTracker, points, videoFrame);
```

Initialize a Video Player to Display the Results

Create a video player object for displaying video frames.

```
videoPlayer  = vision.VideoPlayer('Position',...
    [100 100 [size(videoFrame, 2), size(videoFrame, 1)]+30]);
```

Track the Face

Track the points from frame to frame, and use estimateGeometricTransform function to estimate the motion of the face.

```
% Make a copy of the points to be used for computing the geometric
% transformation between the points in the previous and the current frames
oldPoints = points;

while ~isDone(videoFileReader)
    % get the next frame
    videoFrame = step(videoFileReader);

    % Track the points. Note that some points may be lost.
    [points, isFound] = step(pointTracker, videoFrame);
    visiblePoints = points(isFound, :);
    oldInliers = oldPoints(isFound, :);

    if size(visiblePoints, 1) >= 2 % need at least 2 points

        % Estimate the geometric transformation between the old points
        % and the new points and eliminate outliers
        [xform, oldInliers, visiblePoints] = estimateGeometricTransform(...
            oldInliers, visiblePoints, 'similarity', 'MaxDistance', 4);

        % Apply the transformation to the bounding box
        [bboxPolygon(1:2:end), bboxPolygon(2:2:end)] ...
            = transformPointsForward(xform, bboxPolygon(1:2:end),
bboxPolygon(2:2:end));

        % Insert a bounding box around the object being tracked
        videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon);

        % Display tracked points
        videoFrame = insertMarker(videoFrame, visiblePoints, '+', ...
            'Color', 'white');

        % Reset the points
        oldPoints = visiblePoints;
        setPoints(pointTracker, oldPoints);
    end

    % Display the annotated video frame using the video player object
    step(videoPlayer, videoFrame);
end

% Clean up
release(videoFileReader);
release(videoPlayer);
release(pointTracker);
```

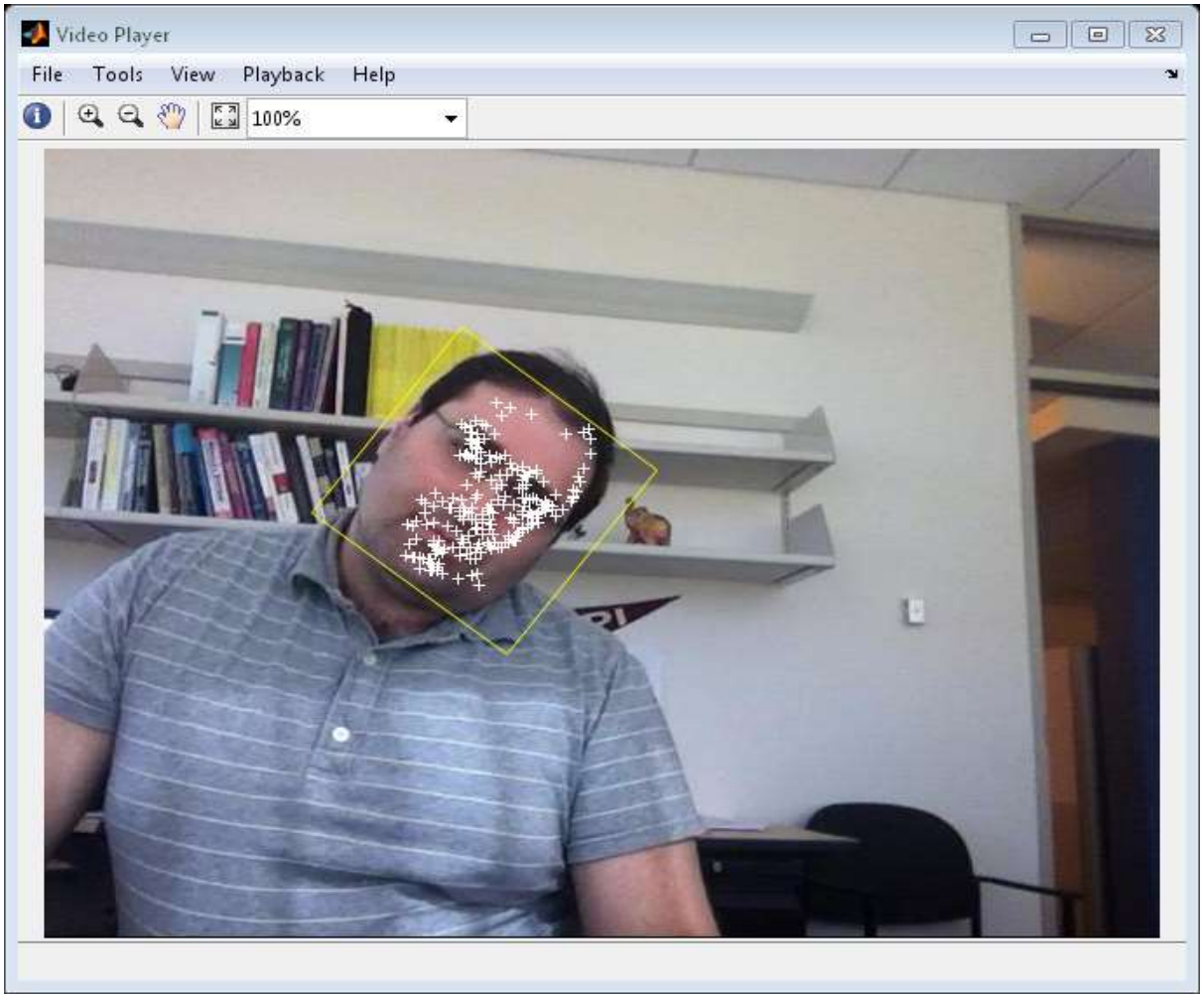

Figure 7: Face tracking using KLT algorithm

Hidden Markov Model

A hidden Markov model (HMM) is a triple (π,A,B).

$\Pi = (\pi_i)$ the vector of the initial state probabilities;

$A = (a_{ij})$ the state transition matrix; $Pr(x_{i_t} \mid x_{j_{t-1}})$

$B = (b_{ij})$ the confusion matrix; $Pr(y_i \mid x_j)$

Each probability in the state transition matrix and in the confusion matrix is time independent - that is, the matrices do not change in time as the system evolves. In practice, this is one of the most unrealistic assumptions of Markov models about real processes.

Once a system can be described as a HMM, three problems can be solved. The first two are pattern recognition problems: Finding the probability of an observed sequence given a HMM (evaluation); and finding the sequence of hidden states that most probably generated an observed sequence (decoding). The third problem is generating a HMM given a sequence of observations (learning).[3]

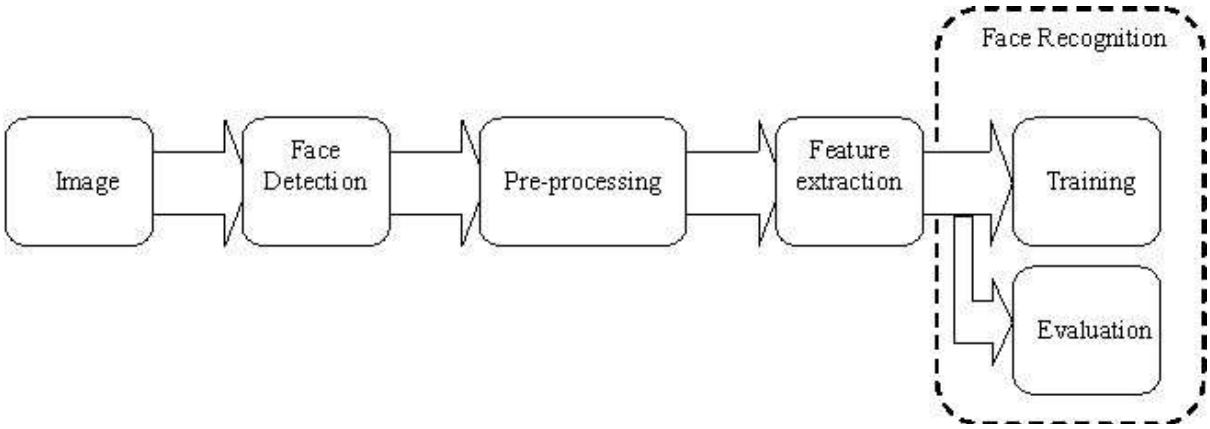

Figure 8 : Working of HMM

1. Evaluation

Consider the problem where we have a number of HMMs (that is, a set of (π,A,B) triples) describing different systems, and asequence of observations. We may want to know which HMM most probably generated the given sequence. For example, we may have a `Summer' model and a `Winter' model for the seaweed, since behaviour is likely to be different from season to season - we may then hope to determine the season on the basis of a sequence of dampness observations.

We use the forward algorithm to calculate the probability of an observation sequence given a particular HMM, and hence choose the most probable HMM.

This type of problem occurs in speech recognition where a large number of Markov models will be used, each one modelling a particular word. An observation sequence is

formed from a spoken word, and this word is recognised by identifying the most probable HMM for the observations.

2. Decoding

Finding the most probable sequence of hidden states given some observations

Another related problem, and the one usually of most interest, is to find the hidden states that generated the observed output. In many cases we are interested in the hidden states of the model since they represent something of value that is not directly observable.

Consider the example of the seaweed and the weather; a blind hermit can only sense the seaweed state, but needs to know the weather, i.e. the hidden states.

We use the Viterbi algorithm to determine the most probable sequence of hidden states given a sequence of observations and a HMM.

Another widespread application of the Viterbi algorithm is in Natural Language Processing, to tag words with their syntactic class (noun, verb etc.) The words in a sentence are the observable states and the syntactic classes are the hidden states (note that many words, such as wind, fish, may have more than one syntactical interpretation). By finding the most probable hidden states for a sentence of words, we have found the most probable syntactic class for a word, given the surrounding context. Thereafter we may use the primitive grammar so extracted for a number of purposes, such as recapturing `meaning'.

3. Learning

Generating a HMM from a sequence of observations

The third, and much the hardest, problem associated with HMMs is to take a sequence of observations (from a known set), known to represent a set of hidden states, and fit the most probable HMM; that is, determine the (π,A,B) triple that most probably describes what is seen.

The forward-backward algorithm is of use when the matrices A and B are not directly (empirically) measurable, as is very often the case in real applications.

HMMs, described by a vector and two matrices (π,A,B) are of great value in describing real systems since, although usually only an approximation, they are amenable to analysis. Commonly solved problems are:

1. Matching the most likely system to a sequence of observations -evaluation, solved using the forward algorithm;
2. Determining the hidden sequence most likely to have generated a sequence of observations - decoding, solved using the Viterbi algorithm;
3. Determining the model parameters most likely to have generated a sequence of observations - learning, solved using the forward-backward algorithm.

Baum-Welch Algorithm

This method can be derived using simple ``occurrence counting" arguments or using calculus to maximize the auxiliary quantity

$$Q(\lambda, \bar{\lambda}) = \sum_{q} p\{q|O, \lambda\} \log[p\{O, q, \bar{\lambda}\}]$$

over $\bar{\lambda}$ [],[, p 344-346,]. A special feature of the algorithm is the guaranteed convergence. To describe the *Baum-Welch algorithm,* (also known as *Forward-Backward algorithm*), we need to define two more auxiliary variables, in addition to the forward and backward variables defined in a previous section. These variables can however be expressed in terms of the forward and backward variables.

First one of those variables is defined as the probability of being in state *i* at *t=t* and in state *j* at *t=t*+1. Formally,

$$\xi_t(i,j) = p\{q_t = i, q_{t+1} = j|O, \lambda\} \tag{1.10}$$

This is the same as,

$$\xi_t(i,j) = \frac{p\{q_t = i, q_{t+1} = j, O|\lambda\}}{p\{O|\lambda\}} \tag{1.11}$$

Using forward and backward variables this can be expressed as,

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})} \tag{1.12}$$

The second variable is the a posteriori probability,

$$\gamma_t(i) = p\{q_t = i|O, \lambda\} \tag{1.13}$$

that is the probability of being in state *i* at *t=t,* given the observation sequence and the model. In forward and backward variables this can be expressed by,

$$\gamma_t(i) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \right] \tag{1.14}$$

One can see that the relationship between $\gamma_t(i)$ and $\xi_t(i,j)$ is given by,

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), \quad 1 \leq i \leq N, \quad 1 \leq t \leq M \tag{1.15}$$

Now it is possible to describe the Baum-Welch learning process, where parameters of the HMM is updated in such a way to maximize the quantity, $p\{\boldsymbol{O}|\lambda\}$. Assuming a starting model $\lambda = (A, B, \pi)$,we calculate the 'α's and 'β's using the recursions 1.5 and 1.2, and then 'ξ's and 'γ's using 1.12 and 1.15. Next step is to update the HMM parameters according to eqns 1.16 to 1.18, known as *re-estimation formulas*.

$$\bar{\pi}_i = \gamma_1(i), \quad 1 \leq i \leq N \tag{1.16}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq N \tag{1.17}$$

$$\bar{b}_j(k) = \frac{\sum_{\substack{t=1 \\ O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \tag{1.18}$$

These re-estimation formulas can easily be modified to deal with the continuous density case too.[2]

IP-Surveillance design

IP-Surveillance is a term for a security system that gives users the ability to monitor and record video and/or audio over an IP (Internet Protocol-based) computer network such as a local area network (LAN) or the Internet. In a simple IP-Surveillance system, this involves the use of a network camera (or an analog camera with a video encoder/video server), a network switch, a PC for viewing, managing and storing video, and video management software.

Unlike analog video systems that use dedicated point-to-point analog cabling from the camera location to the viewing/recording station, IP-Surveillance (or network video) uses the IP network technology as the backbone for transporting information. In an IP-Surveillance application, digitized video and/or audio streams can be sent to any location even around the world if desired via a wired and/or wireless IP network, enabling video monitoring and recording from anywhere with network access.

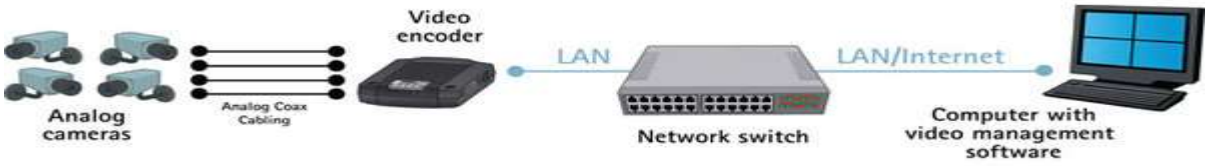

Figure 9 : An IP-Surveillance or network video system

Because of the digital nature and method of video distribution, IP-Surveillance provides a host of benefits and advanced functionalities that gives you greater control and management of live and recorded video, as well as alarm events. This makes the system highly suited to security surveillance applications.

The advantages include:

1) **Remote accessibility:** You can access live and recorded video at any time and from virtually any networked location in the world. Multiple, authorized users at different locations may be able to access live or recorded video. This is advantageous if your company wants a third-party, such as a security firm, to benefit from and have access to the video. In a traditional analog CCTV system, you need to be in a specific, on-site monitoring location to view and manage video, and off-site video access would not be possible without some additional equipment, such as a video encoder or a network DVR (digital video recorder).

2) **High image quality:** High image quality is essential in a security surveillance application. You want to be able to clearly capture an incident in progress and identify persons or objects involved. In a network video system, the quality of images produced can be more easily retained than in an analog surveillance system. With an analog video system, the captured images are degraded with every conversion that the images make between analog and digital formats and with the cabling distance. The further the analog video signals travel, the weaker they become. In a fully digital IP-Surveillance system, images from a network camera are digitized once and they stay digital with no unnecessary conversions and no image degradation due to distance traveled. In addition, digital images can be more easily stored and retrieved than is the case with the use of analog video tapes. A network camera that uses progressive scan technology provides clearer images of moving objects because the whole image is presented at one time. With an analog video signal, two consecutive interlaced fields of lines are presented to form an image, and when displayed on a PC monitor, blurriness occurs when objects move between the image capture of the two interlaced fields.

3) **Easy, future-proof integration:** Network video products based on open standards can be easily integrated with computer and Ethernet-based information, audio and security systems, video management and application software, and other digital devices. For instance, a network camera can be linked to specialized software programs that could, for example,

integrate video with a Point of Sales system, or analyze the visual and/or audio data to detect wanted persons in a crowd or unauthorized access to specific areas.

4) **Scalable and flexible:** An IP-Surveillance system can grow with your needs. You can add as many network video products to the system as desired without significant or costly changes to the network infrastructure. You can place and network the products from virtually any location, and the system can be as open or as closed as you wish.

5) **Cost-effective:** An IP-Surveillance system has a lower total cost of ownership than a traditional analog CCTV surveillance. Management and equipment costs are lower since back-end applications and storage run on industry standard, open systems-based servers not on proprietary hardware such as a DVR in the case of an analog CCTV system. Additional cost savings come from the infrastructure used. IP-based video streams can be routed around the world using a variety of interoperable infrastructure. IP-based networks such as LANs and the Internet, and various connection methods such as wireless are much less expensive alternatives than traditional coaxial and fiber needed for an analog CCTV system. In addition, an IP infrastructure can be leveraged for other applications across the organization.

6) **Event management and intelligent video:** There is often too much video recorded and lack of time to properly analyze them. Advanced network cameras/video encoders with built-in intelligence or analytics take care of this by reducing the amount of uninteresting video recorded and enabling programmed responses. Advanced network cameras/video encoders have such features as built-in video motion detection, audio detection alarm, active tampering alarm, I/O connections, and alarm and event management functionalities. These features enable the network cameras/video encoders to be constantly on guard in analyzing inputs and waiting for an impulse to kick-start an action or a series of actions. Having intelligence/ analytics conducted at the network camera/video encoder rather than at the recording server reduces network bandwidth usage and storage needs since only actionable data (video) is sent over the network. In addition, less computing power is required from the recording server. Event management functionalities can be configured using the network video product user interface or a video management software program. Users can define the alarms/events by setting the type of triggers to be used and when, as well as the responses (e.g., recording to one or multiple sites whether local and/or off-site for security purposes; activation of external devices such as alarms, lights and doors; and notification messages to users). A security personnel's ability to protect people, property and assets can be enhanced by the flexibility and power of IP-Surveillance technology. IP-Surveillance systems have been installed in indoor/outdoor and private/public spaces; for example, in stores, homes, day care centers, schools, banks, government offices, factories, warehouses, railway/subway stations and airports.[7]

Network Camera

A network camera can be described as a camera and computer combined in one unit. It has a compression chip, an operating system, a built-in web server, FTP (File Transfer Protocol) server, FTP client, e-mail client, alarm management and much more. A network camera, unlike a web camera, does not need to be attached to a PC; it operates independently and connects, as with a PC, directly to an IP network. It can be placed wherever there is a wired or wireless network connection. The network camera captures and sends live images, enabling authorized users to locally or remotely view, store and manage video over a standard IP-based network infrastructure.

Many types of network cameras are available today, and no matter what your needs are, there is a network camera available to meet them. Although analog cameras are available in a similar variety, network cameras can now offer more benefits, including better image quality and greater installation flexibility. For some special applications, such as very high image resolution or wireless needs, network cameras are the only option.

PTZ network cameras

The camera's view can be remotely controlled, either manually or automatically, for panning from side to side, tilting up and down, and zooming in and out of an area or object. There are now mechanical as well as non-mechanical PTZ cameras.

Mechanical PTZ cameras can pan, tilt and zoom through manual or automatic control. In a manual operation, an operator can use a PTZ camera to follow, for instance, a person in a retail store. PTZ cameras are mainly used indoors and in applications where an operator is employed and where the visibility of the camera's viewing angle is desirable or not an issue. The optical zoom on PTZ cameras typically ranges from 10X to 26X. A PTZ camera can be mounted on a ceiling or wall.

A difference between PTZ cameras and PTZ domes is that many PTZ cameras do not have full 360-degree pan due to a mechanical stop that prevents the cameras from making a continuous circular movement. It means that the camera cannot follow a person walking continuously in a full circle around the camera. An exception is the 215 PTZ Network Camera, which thanks to its auto-flip functionality, can instantly flip the camera head 180 degrees and continue to pan beyond its zero point. The camera can then continue to follow a passing person or object, regardless of the direction. Another difference between PTZ cameras and PTZ domes is that PTZ cameras are not made for continuous automatic operation or so-called 'guard tours'.

A non-mechanical PTZ camera uses a megapixel sensor and a wide-angle lens to enable it to have a viewing angle of 100 degrees to 180 degrees (or even wider in same cases). Such a camera allows an operator to zoom in on any part of a scene without any mechanical movement. The key advantage is that there is no wear and tear since the camera has no moving parts. Zooming in on a new area of a scene is immediate. In a traditional PTZ camera, this can take up to 1 second. Since a non-mechanical PTZ camera's viewing angle is not visible, it is ideal for discreet installations. To obtain good image quality, pan, tilt and

zoom should be limited. If such a camera has a 3 megapixel sensor, the recommended maximum viewing angle is 140 degrees with a 3X zoom capability. This type of camera is typically mounted on a wall.

Wireless: A network camera with built-in wireless support is a consideration when running a cable between a LAN and a network camera is impractical, difficult or expensive. Wireless network cameras are suitable for use in outdoor situations, in environments such as historic buildings where the installation of cables would damage the interior, or in cases where there is a need to move cameras to new locations on a regular basis, such as in a supermarket. Ensure that the wireless network camera supports security protocols such as IEEE 802.1X and WPA/WPA2 (Wi-Fi Protected Access), which will help secure the wireless communication.

Security and management: At a basic level, a video surveillance network camera should provide different levels of password-protected access to a network camera. For instance, some authorized users may only have access to view images from specific cameras; others have operator-level access, and a few have access to administer all settings in a network camera. Beyond multi-level password protection, a network camera may offer HTTPS encryption for secure communication; IP address filtering, which gives or denies access rights to defined IP addresses; IEEE 802.1X to control network access; and user access log.

Network management features: They include support for Quality of Service (QoS), which can prioritize and reserve network capacity for mission-critical surveillance in a QoS aware network, and support for Internet Protocol version 6 (IPv6) in addition to IPv4 addresses.

Network

The next consideration to make is assessing your network needs.

Network switches allow devices such as network cameras, servers and PCs to communicate with each other to share information and, in some cases, a common Internet connection. Network designs can take many forms and may vary in terms of performance and security.

First, determine what your company is using the network for and how congested your local area network (LAN) or wide area network (WAN) is.

If you are implementing a smaller surveillance system involving 8 to 10 cameras, you should be able to use a basic 100-megabit (Mbit) network switch without having to consider bandwidth limitations. Most companies can implement a surveillance system of this size using their existing network.

If you are implementing 10 cameras or more, you should try to estimate the load on the network using a few rules of thumb:

> A camera will use approx. 2 to 3 megabits of bandwidth when configured to deliver high-quality images at high frame rates.

> With more than 12 to 15 cameras, you should consider using a switch with a gigabit (Gbit) backbone. If a gigabit-supporting switch is used, the server that runs the video management software should have a gigabit network adapter installed.

Determine the pattern of congestion levels over a given period to find out if you have to install additional bandwidth capacity on your network or whether you can make use of the same network as for general business activities. It may be that the network traffic drops off during nighttime and weekends the times when you may want to activate the video surveillance system. The usage pattern will help you to determine whether you can a) simply use the same network infrastructure for your general purpose needs as for your surveillance needs, or b) use a combination of existing general purpose network as well as a new network for IP-Surveillance. If additional network capacity is needed, new cabling is normally not needed since adding a switch or reconfiguring the patch panel may solve the problem.[7]

Storage

Similar to the way a PC can "save" documents and other files, video can be stored on a server or PC hard disk. Specialized equipment is not needed since a storage solution treats video data like any other large group of files that can be stored, accessed and eventually deleted. Video storage, however, puts new strains on storage hardware because it may be required to operate on a continual basis, as opposed to during normal business hours with other types of files. In addition, video by nature generates very large amounts of data, creating high demands on the storage solution.

Calculating the storage needs

In order to appropriately calculate the storage requirements of a network surveillance system, there are a number of elements to factor in, such as the number of cameras required in your installation, the number of hours a day each camera will be recording, how long the data will be stored, and whether the system uses event triggers such as video motion detection or continuous recording. Additional parameters such as frame rate, compression, image quality and scene complexity (little motion or lots of motion) should also be considered.

The type of video compression employed also effects storage calculations. The H.264 compression format is by far the most efficient video compression technique available today. Without compromising image quality, an H.264 encoder can reduce the size of a digital video file by more than 80 percent compared with the Motion JPEG format and as much as 50 percent more than with the MPEG-4 Part 2 (referred to simply as MPEG-4 in future references) standard. This means much less network bandwidth and storage space are required for an H.264 video file. Or seen another way, much higher video quality can be achieved for a given bit rate.

With Motion JPEG, storage requirements vary depending on the frame rate, resolution and degree of compression. With H.264 and MPEG-4, bit rate is the key factor in determining the corresponding storage requirements.

There is a clear formula for calculating storage requirements when it comes to Motion JPEG (see calculation below) because Motion JPEG consists of one individual file for each image. Calculations are not so clear-cut for H.264 and MPEG-4 because of a number of variables that affect bit rate levels. However, sample calculations for H.264 and MPEG-4 are also provided below:

H.264 calculation:

Bit rate / 8(bits in a byte) x 3600s = Kilobyte (KB) per hour / 1000 = Megabyte (MB) per hour
MB per hour x hours of operation per day / 1000 = Gigabyte (GB) per day
GB per day x requested period of storage = Storage need.[7]

We hope this literature survey has been helpful in providing guidelines for implementing an 'Integrated Security System' using an IP-Surveillance system. While there are many considerations to take into account, it is relatively easy to set up and operate an IP-Surveillance system once we have defined our application requirements and determined the components we require. We have covered most of the software and hardware aspects of our project. The literature survey concentrates on various applications of Smart Surveillance and its needs. It also mentions technologies that are currently using image processing as a tool. It also explains algorithms like KLT, Viola-Jones & HMM which form the foundation of Image recognition. Last thing that we have discussed in our literature survey is the hardware mechanism of a surveillance system. It covers all the essential aspects of a surveillance network.

3. REQUIREMENT ANALYSIS

3.1 SRS DOCUMENT

3.1.1 Introduction

3.1.1.1 Purpose

The main purpose is to develop a system that tracks criminals and suspects with very little manual search. Since the system does the part of looking for a criminal using image processing and database search, the police do not have to spend a lot of time looking for criminals on their own.

3.1.1.2 Scope

The system is developed with an intent of reducing the time taken to look for a criminal or a suspect and also to make it easier to find out these suspects. The system can be used by the police or departments which are dedicated towards providing security to the society.

3.1.1.3 Intended Audience

The intended Audience for this document are Police, Security departments.

3.1.2 Overall Description

The application product, 'Student's Interface' will help to students in comparative studying. Also, by having access to their daily attendance record, they will be notified and motivated to keep their attendance level above the 75% mark. It will tremendously help visually challenged students.

3.1.2.1 Product Perspective

The software is self-contained and dynamic. It uses a large database and performs image recognition efficiently.

3.1.2.2 Product Functions

1. Frame Capturing: Capture frames from continuous videos taken from a CCTV camera.

2. Detection: The required objects such as number plates, faces are detected. This may also be unidentified stationary objects.

3. All the detected objects are checked using Image processing on the frame and comparing the objects with data stored in the database. For stationary objects, the number of these objects are compared with predefined values to check if an alert is required.

4. Alert generation: If there are matches between the objects in the frames and the database values, alert is generated which is then passed on to the cops.

3.1.2.3 User Classes and Characteristics

Ø **User-** Police or other surveillance departments.

Ø **Characteristics-** They should have access to the processed results.

3.1.2.4 Operating Environment

Operating Systems:

1. Windows XP and above

Development Environment:

1. SQL Server Management
2. MATLAB R2010a
3. Microsoft Office Access 2007

Programming Languages:

1. MATLAB has been used throughout the project for image processing, database creation etc.

3.1.2.5 Design and Implementation Constraints

1. The CCTV footage might not be of readable quality.
2. The number of faces on screen might exceed the maximum number of faces the software can detect.
3. The system might slow down due to CCTV lag.

3.1.2.6 Assumptions and Dependencies

- Image matching from all angles isn't achievable without intricate algorithms. Matching is accurate in 90% of the cases.
- Devices should be compatible with MATLAB Compiler Runtime (MCR).

3.1.3 External Interface Requirements

3.1.3.1 User Interface

The interface is easy to use and doesn't need too much training. However since we're dealing with large databases here, to make the system highly secure, a number of authentications may be required.

3.1.3.2 Software Interface

The Server software works on any terminal with Windows platform having jdk1.6.0 and above.

3.1.3.3 Hardware Interface

The system needs a large server and backup memory areas to store the databases. CCTV cameras are used to capture videos.

3.1.3.4 Communication Protocols and Interface

Apache server is required. MATLAB is required for Image Processing.

3.1.4 System Features

(A) Face Detection

Description and Priority

We can detect faces in an image using this software. It uses Viola-Jones algorithm to detect faces in an image. It creates a square box for every face detected.

Validity Checks

The system verifies that the there is a face in the image and if so then it must be detected.

Stimulus or Responses

The system approves the detected face by creating square boxes around the detected faces.

(B) Face Recognition

Description and Priority

Authority stores pictures of a criminal in the database. Later CCTV footage is compared with these pictures used as training set. It will search for a similarity, if the picture in footage matches any database pictures then it returns that set name.

Validity Checks

The availability of same image in our training set.

Stimulus or Responses

On receiving the request for comparison the system returns a valid response implying which set it belongs to.

Error Handling/Response to abnormal situation

In case the image in invalid or there is any glitch, the system reports an error.

(C) Image Database

Description and Priority

We can store various images and faces for training system. We can capture the image or manually store it in database

Validity Checks

It tries detecting faces in given images.

Stimulus or Responses

On receiving the detected image it stores it in the image database for future use.

3.1.5 Other Non-Functional Requirements

3.1.5.1 Performance Requirements

To enhance the performance, high quality CCTV cameras are required. In the absence of these, an image optimization system is needed. However it is more feasible to have better CCTV cameras instead. The database requirement is high i.e. a large amount of memory is required. Apart from this, the processing needs to be very fast so that there is no lag.

3.1.5.2 Safety Requirements

Help will be provided along with FAQs and steps of caution to the users.

3.1.5.3 Security Requirements

Since the system maintains a large database, care is to be taken to make sure that the database is secure. Therefore the information is made available using authentication processes so that only designated individuals have access to it.

3.1.5.4 Software Quality Attributes

Coding standards

Programming style is a set of rules or guidelines used when writing the source code for a computer program. It is often claimed that following a particular programming style will help programmers to read and understand source code conforming to the style, and help to avoid introducing errors.

Reliability

Stable database access system and Stable core system.

Availability

Shall be available 24 hours a day, 7 days a week.

Robustness

The System shall be robust enough to have a high degree of fault tolerance.

Usability

The system shall provide easy to use graphical interface.

Integrity

The system should be secure and must protect the databases.

3.2 USE CASE DIAGRAM

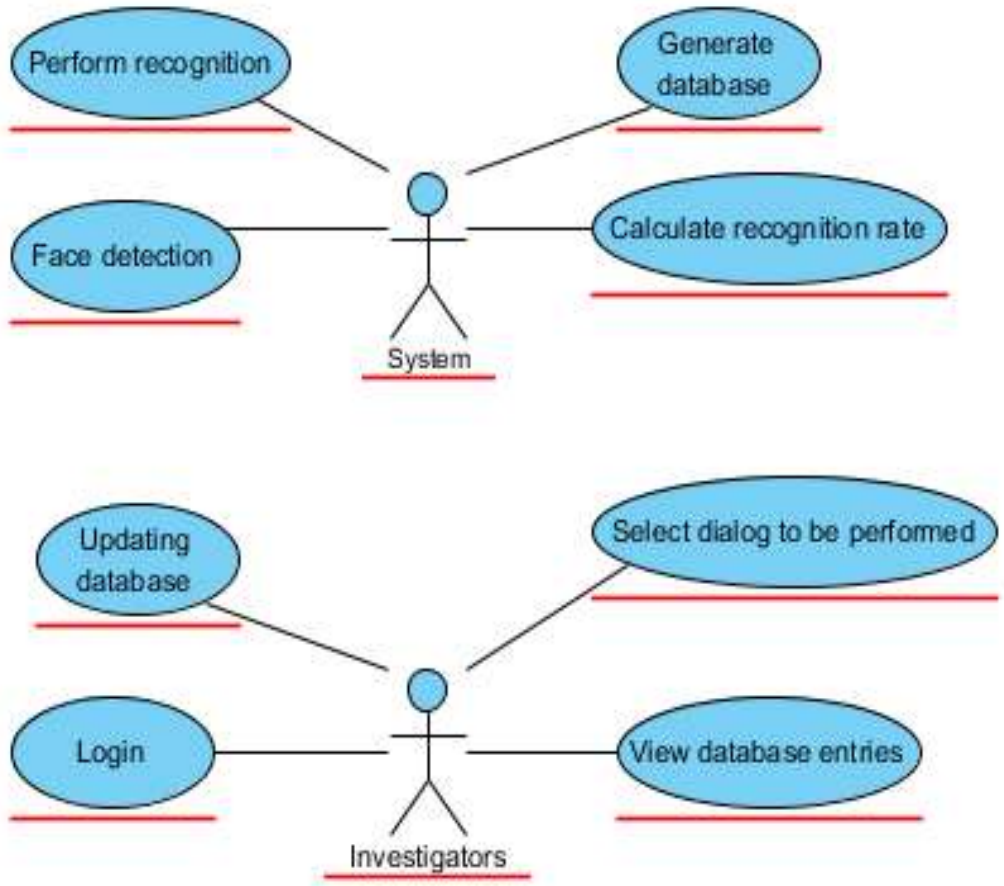

Figure 10: Use Case diagram

3.3 ACTIVITY DIAGRAM

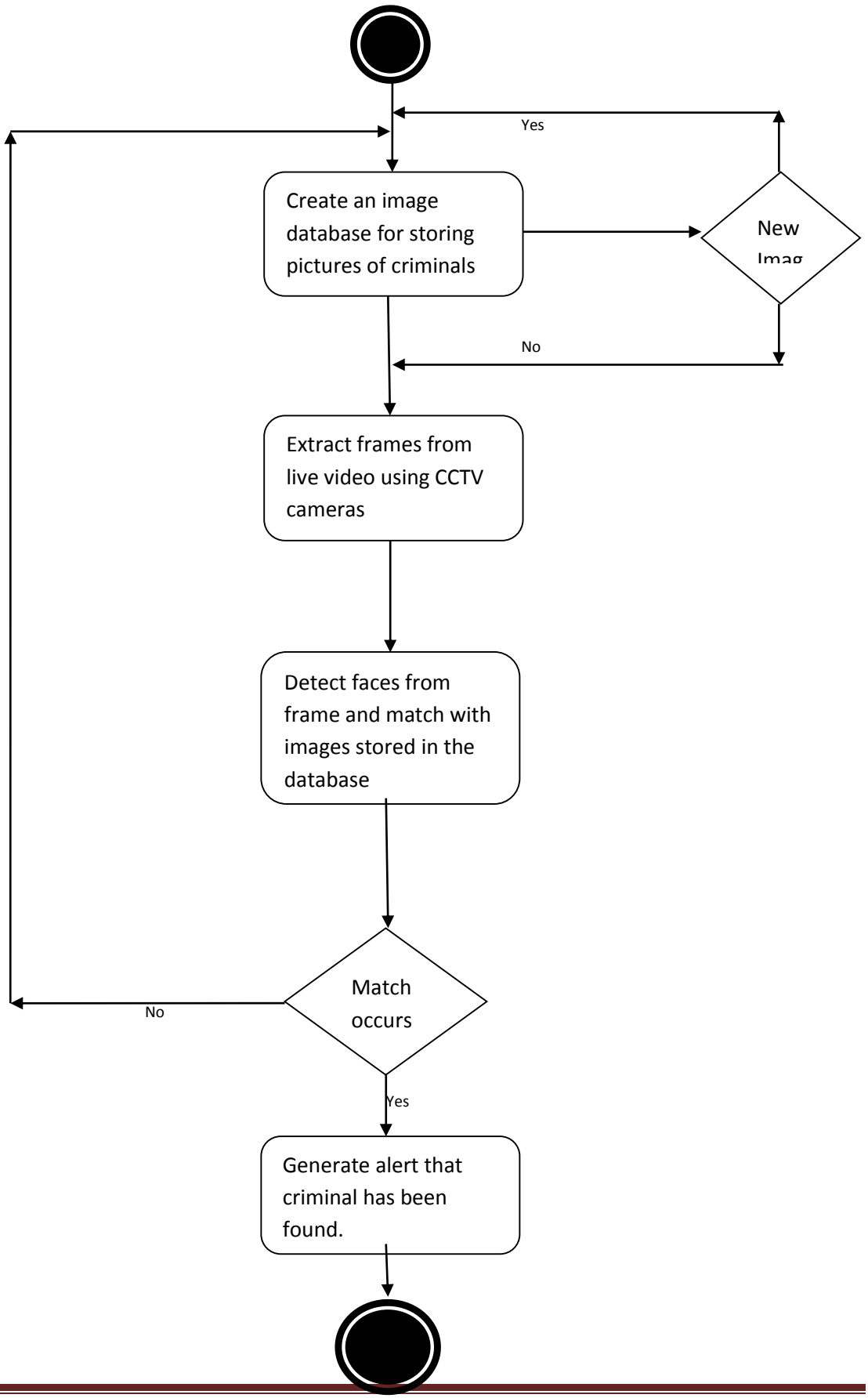

Figure 11: Activity diagram

3.4 STATE DIAGRAM

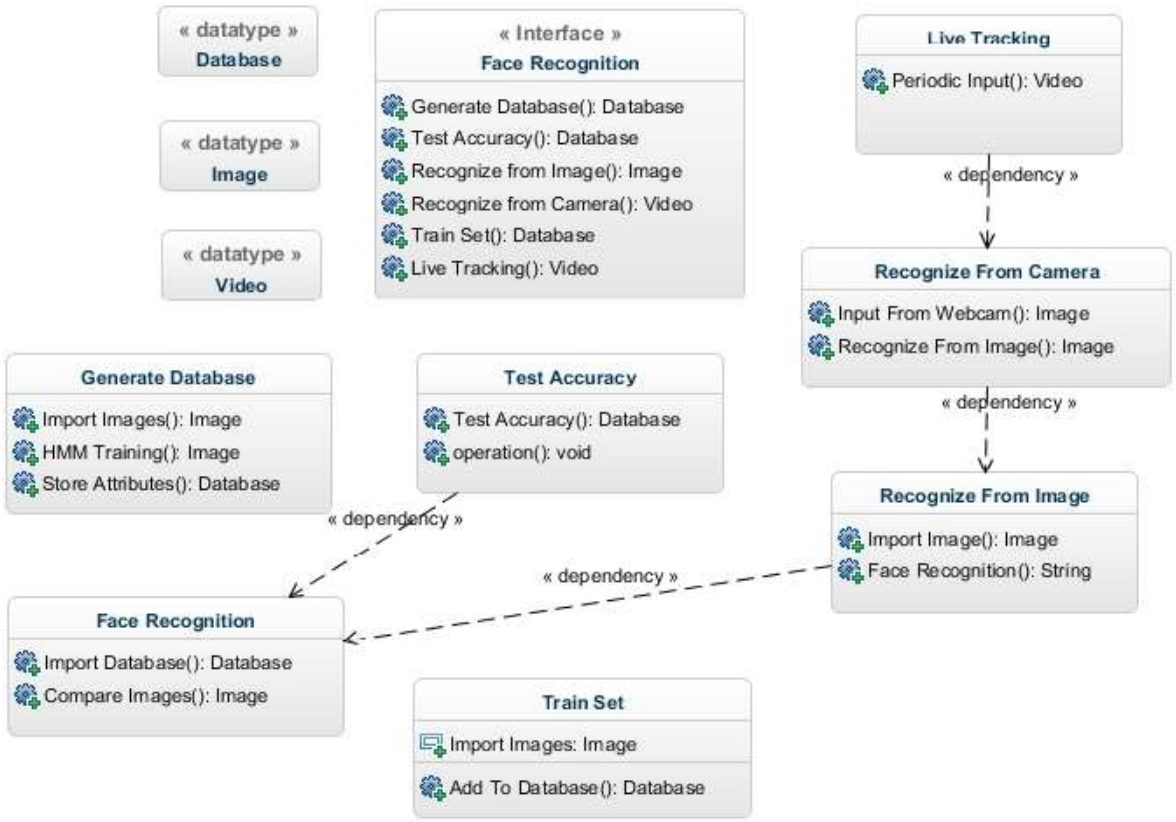

Figure 12: State diagram

4. IMPLEMENTATION

User Interface:

The user gets various options such as ‘generate database’, ‘capture image’ etc.

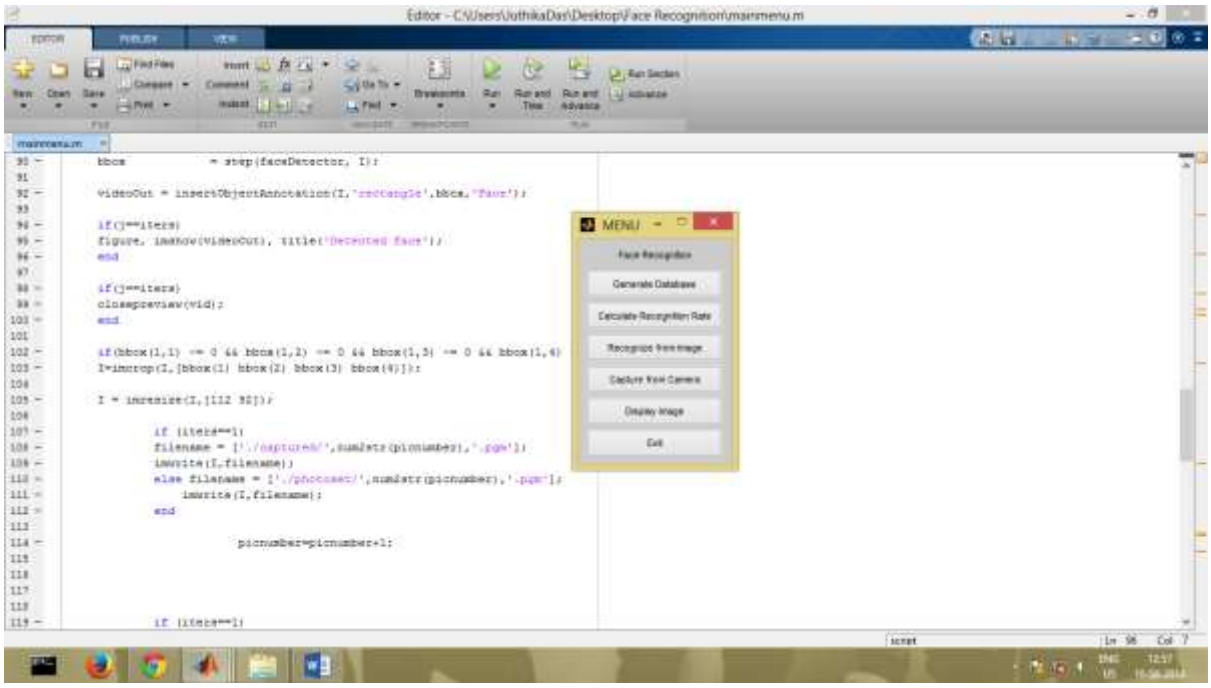

Figure 13: Screenshot of User Interface

Image capture:

User clicks on ‘capture image’ and then ‘train’ so the system takes 10 pictures after a gap of 0.2 seconds.

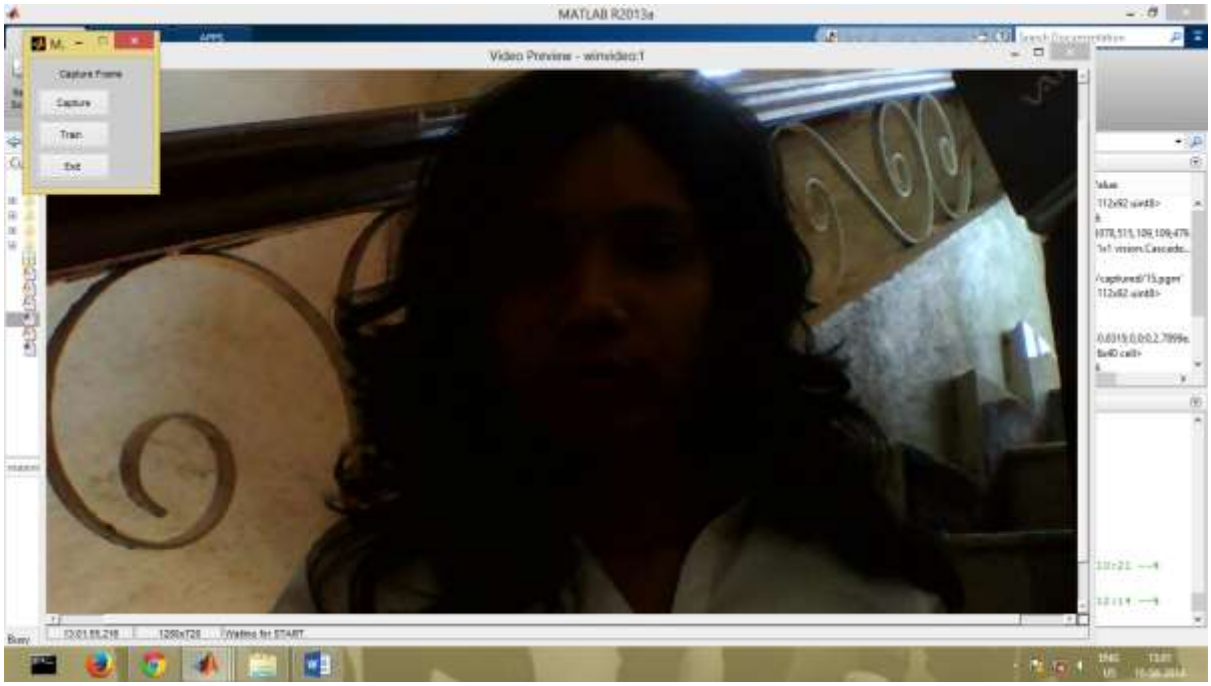

Figure 14: Screenshot of image capture

Training dataset

The images taken are then used to generate a database. Each database is given an ID or can be named by the programmer or the user.

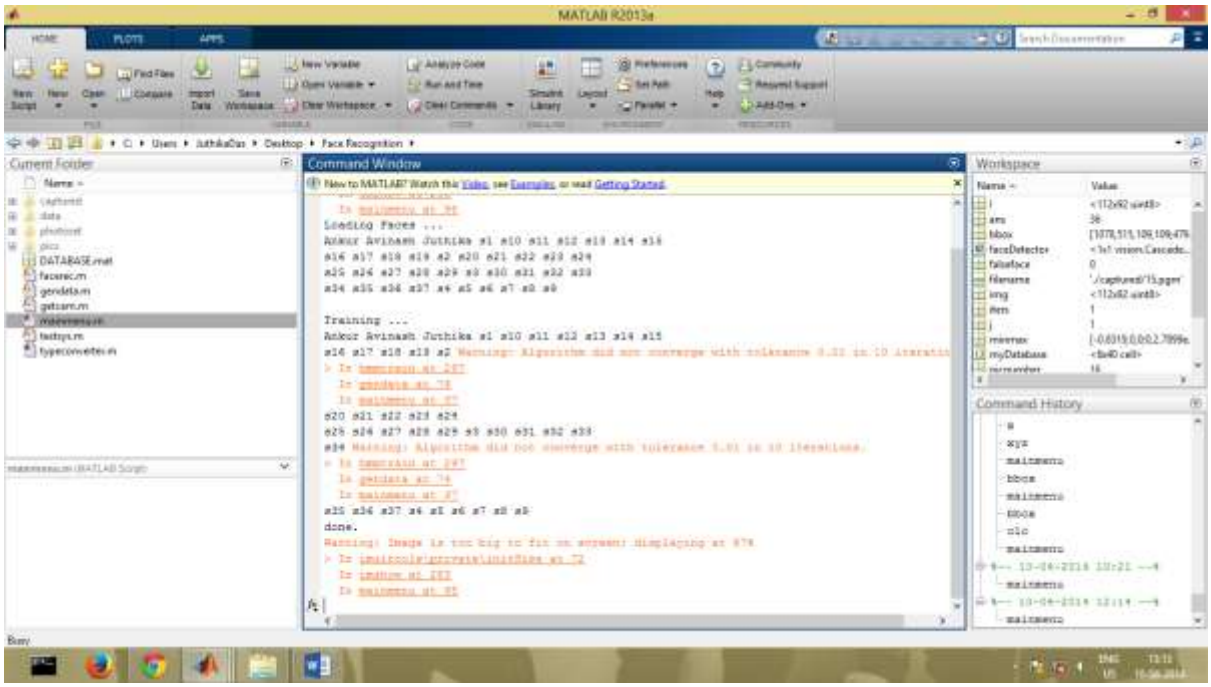

Figure 15: Screenshot of generation of database

Face Detection

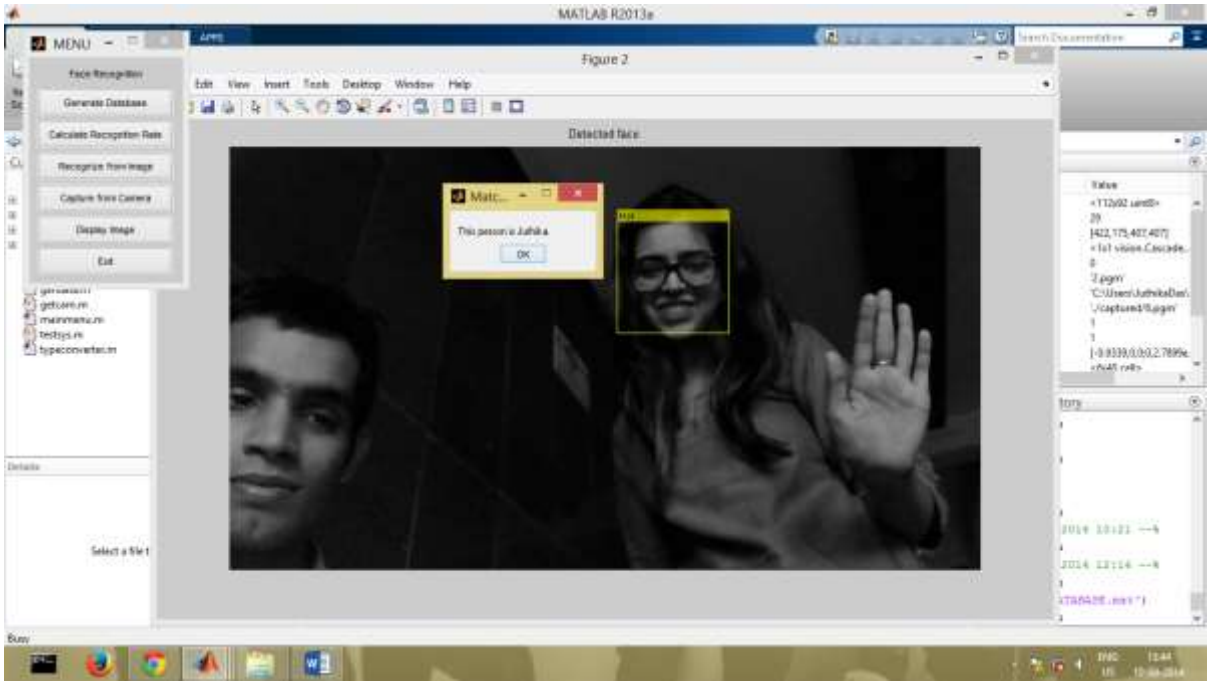

Figure 16: Screenshot of face recognition

5. TESTING

5.1 TESTING METHODS

There are two main methods to design the test case:

WHITE BOX TESTING

It is a test case design method that uses the control structure of the procedure design to derive test cases. The software engineer can derive test cases that do all the following:

- Exercise all independent paths within the model at least once.
- Exercise all logical decisions for both true and false scenarios.
- Execute all loops at their boundaries and within their operational loops.
- Exercise all internal data structure to ensure their validity.

BLACK BOX TESTING

Black box testing attempts to find errors in the external behaviour of the code in the following categories:

- Incorrect or missing functionality.
- Interface errors.
- Errors in data structures used by interfaces.
- Behaviour or performance errors.
- Initialization and termination errors.

UNIT TESTING

Unit testing is the first level of dynamic testing and is first the responsibility of the developers and then of the testers. Unit testing is performed after the expected test results are met or differences are explainable / acceptable.

SYSTEM TESTING

It is a series of tests whose purpose is to fully exercise the computer based system.

Some common types of tests are:

- Recovery Testing
- Security Testing

5.2 TEST CASES

TEST CASE ID	OBJECTIVE	STEPS/ DESCRIPTION	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1.	Displaying the stored images	Click on display images.	Saved images	Selected image is displayed.	Selected image is displayed.	Success
2.	Generating a database	Click on generate database	Image sets for training	Database is generated	Database is successfully generated.	Success
3.	Finding Recognition Rate	Click Calculate Recognition Rate.	Image sets	Displays image recognition rate as percentage	Displays image recognition rate as percentage	Success
4.	Performing image matching	Click Recognize from Image	Selected image	The recognized person's ID	ID matched by the database as the person	Success
5.	Capture frames from the camera	Click on Capture from Camera	Single snapshot from camera	Detects and performs image matching on the face in the snapshot	Detects face and matches it to one present in the database	Success

Table 3: Table of test cases

6. RESULT AND ANALYSIS

Smart Surveillance System was designed to reduce manual burden for security purposes. We aimed at creating a system that can track criminals without almost zero human intervention. The system provides various functionalities as was envisaged at the start of the project.

The system has been turned out to be simple to operate and easy to understand. The project has been developed using MATLAB. The input to this project is real time video which is very close to the expected implementation of this using CCTV cameras and live videos. The GUI is also developed using MATLAB.

In analysis, the system turns out to satisfy almost all requirements specified. The database could be made independent for large scale use. The interface could be made more interactive. However, will the current functionalities of the project, it will help the general population a great deal as a manual process has been automated. It removes the possible human bias from the system making the patient feel secure about various processes. Various modules have been integrated after successful and error free testing. Some modifications could be made to make the interface appear even more interactive.

Specific changes can be made to different modules where customization is required. But these limitations will come to light only when more and more users use the system on a large scale. Many modules with their specific functionality have been implemented. In future, newer technologies could be used to provide even more advanced functionality.

7. CONCLUSION AND FUTURE WORK

- Smart Surveillance system is very important for an efficient criminal tracking system. With a population of 1.2 billion, we surely need to move over manual criminal tracking and embrace this fresh advancement in technology.
- The system takes inputs from CCTV cameras and using little manual help, captures them and matches them with a database of images that can be updated as and when required. The system efficiently recognizes the person whose image is obtained.
- Not only does the system identify the person's face from a given angle, since its training set contains of various angles of the person's face, it tries to identify the person from many angles.
- While making the system, an eye has been kept on making it as user friendly, as cost effective and as flexible as possible.
- This proposed system is getting ready to be implemented in many large scale sectors for their enhanced security. It can also be implemented in areas of maximum threat. This system is used in restricted areas where only the authenticated persons are allowed. The concept of the proposal is the intrusion detection by facial recognition. Since this project is very expensive and difficult to implement in a small scale by students we have showed the simulation results using MATLAB.

8. REFERENCES

PAPERS

[1] Hidden Markov Model-based face recognition using selective attention.
A.A. Salaha, M. Bicegob, L. Akaruna, E. Grossob, M. Tistarellic.

[2] Face Detection and Recognition using Hidden Markhov Models.
Ara V. Nefian, Monson H. Hayes III.

[3] An Embedded HMM - based approach for face detection and Recognition.
Ara V. Nean and Monson H. Hayes III.

[4] Rapid Object Detection using a Boosted Cascade of Simple Features.
Paul Viola, Michael Jones.

[5] An Introduction to Hidden Markov Models.
L. R. .Rabiner and B. H. Juang

[6] Intrusion Detection by Facial Recognition using CCTV Cameras with Video Management System
Mahalakshmi. R, Manjula. M, Saranya. S, Vaishnavi. P, Shalini. J. & Arasa Kumar. R.

[7] IP-Surveillance design guide
Axis Communications

MATLAB

- StackOverflow.com
- Mathworks.com