Islamic University of Technology
The Organization of the Islamic Cooperation
(OIC)

An Approach of Finding Visual Intention through Low Cost Gazing

Submitted by:
Shahed Anzarus Sabab
114405
Sayed Rizban Hussain
114406

Supervised by:
Dr. Md. Kamrul Hasan
Co – Supervised by:
Dr. Md. Hasanul Kabir

Department Of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

October, 2015

Thanks to Allah (SWT), the most gracious and the most merciful. The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the people whose ceaseless cooperation made it possible.

Acknowledgements

We would like to express our heartfelt thanks our supervisor

Dr. Md. Kamrul Hasan

Associate Professor

And other founding members of Systems and Software Lab

Hasan Mahmud

Assistant Professor

And

Md. Mohiuddin Khan

Assistant Professor

For their continuous and invaluable support throughout our thesis work. Their constructive ideas helped us a lot to find right way in our research.

Abstract: Human intention process is closely related to design an interactive interface which is dynamic. **Human Computer Interaction (HCI)** largely depends on human cognition. Cognition is what goes in our heads when we carry out our everyday activities .Human perception can be gathered from auditory or visual aspects. Visual information signifies important classification of human intention. **Intension** is the set of attention that represents an approach to achieve a goal by a decision making process. Attention involves our auditory and visual senses. Human intention analysis is a big open space for the researchers. Human intention can be defined by analyzing the gaze information of the humans. Eye gaze is the focus point where one is looking. A low cost approach of gazing involves using webcam to specify the gaze points. Defining our own feature space and alongside a training dataset for classifying the content of an interface components using the perceived value of eye movement. Based on the content and results we are able to define human intention on a particular interface.

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and investigation carried out by Shahed Anzarus Sabab and Sayed Rizban Hussain under the supervision of Dr. Md. Kamrul Hasan and co supervision of Dr. Md. Hasanul Kabir in the Department of Computer Science and Engineering (CSE), IUT, Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Shahed Anzarus Sabab
Student ID: 114405

Sayed Rizban Hussain
Student ID: 114406

Supervisor:

Dr. Md. Kamrul Hasan
Associate Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Table of Contents

1 INTRODUCTION:

Human Computer Interaction largely depends on human cognition. Cognition is what goes on in our heads when we carry out our everyday activities. It involves cognitive processes, like thinking, remembering, learning, daydreaming, and decision making, seeing, reading, writing and talking. From human cognition, human. Intention [8,9,13] is the set of attention[6] that represents an approach to achieve a goal by a decision making process. Human intention represents human interest and goal. The way to gain information about something can be represented by human intention. Designing an interactive interface largely depends on user experience. An interface basically contains two types of information. Like in different websites there are bulk of information which are represented by text or graphical representations but the region of interest varies from person to person [16]. Some likes to gain information through textual information, some likes to see pictorial information. Among these several ways of gaining information which way is preferred by which class of people can be found by analyzing human intention. Works on human intention analysis is not new. Several works are done on this topic. But they all are using their own feature space [8,15]. For our experimentation we are declaring our own feature vector which can be extracted by only using a webcam. These features are distinguishing enough to differentiate among text and pictorial information in an interface. By analyzing this we can comment on how information should be represented on different types of applications based on the class of people they are targeting as their user.

The normal traditional procedure of designing an interface for web or desktop application is largely dependent on user experience. It’s a whole wide research area to design any interface along with human intuition [8,9,10]. The way we design the physical interface of the interactive product must not conflict with the user's cognitive

processes involved in achieving the task. Different design guidelines and standards exists to help the designers create better designs by learning from user's experience.
The interface may include several information based on the type of application. Basically any interface includes two type of information either text or images. Some of the user may like to read text information written on a specific portion of the interface again some may like pictorial representation on a certain portion of the interface. The choice of interest on the Human behavioral and cognition process of the selectively concentrating on a discrete aspect of information that is provided by an interactive interface. That is any user at a time concentrating on a decision making process to achieve a goal. So for better interaction with the interfaces our approach is to motivate designers to design interfaces that can allow user to achieve his/her goal within minimal time limit and trade off following human perception of intention[8,9,10].

Often the complication comes from out of the mind, how the intention [8,9,10] is related to interactive design? In human cognition a large part consists of knowledge acquisition part. Where people follows several ways for knowledge acquisition based on his or her intention. So an interface should be interactive enough to allow users to achieve his/her goal within a minimal time limit and less trade off following human perception of intention. Human intention is the choice of interest on the human behavioral and cognition process of selectively concentrating, on a discrete aspect of information that is provided by an interactive interface. The whole phenomenon is described considering human focus point on the screen. We call this term as the fixation point. Fixation point refers to the focus point on which currently user is focusing. So our solution is straight forward and simple. We are using normal webcam to take the user image, when user is looking at a particular interface. And with the gazing information [7,10,11,13] we will track user fixation point on the screen. Recording specific time interval of gazing information and saccades we will try to classify which part of the interface carries textual information and which part contains pictorial information. And from the classification we can also conclude if the user intention [8,9,10] is related on the text reading or picture viewing placed on a particular point of the interface.

Several works are done on human intention with devices like Tobii or other head mounted devices which can extract unique features like pupil size variation which are extraordinary features but extracting these features are costly as well [8,15]. So are using only a webcam to extract features and declaring our own feature space [20] and feature vectors [20]. Our extracted features are also distinguishing enough to classify and as well as cost efficient extraction process. We are working on a system that is an integrated version of image processing and machine learning. Using tools of machine learning and image processing we are analyzing the feature values and defining human intention. So we are proposing very low cost solution to detect human intention using a webcam. Using only the gazing information of human eye, classify human intention [8,9,10] information relative to a particular interface. Define cost efficient way to get human gaze information like Fixation points, Eye movement vibration, and Fixation time. From the collected information of gazing classify the contents type of an interface is text or images. Based on the contents and gaze information finding human intention of acquiring knowledge. By this process we have also increased the accuracy of detecting pupil position of human eye. Not only detection accuracy has increased but also we have smoothened the pupil movement path throughout the whole interface. Through our system we are eliminating the burden of carrying an extra head mounted device and which is a cost effective solution as well [15].

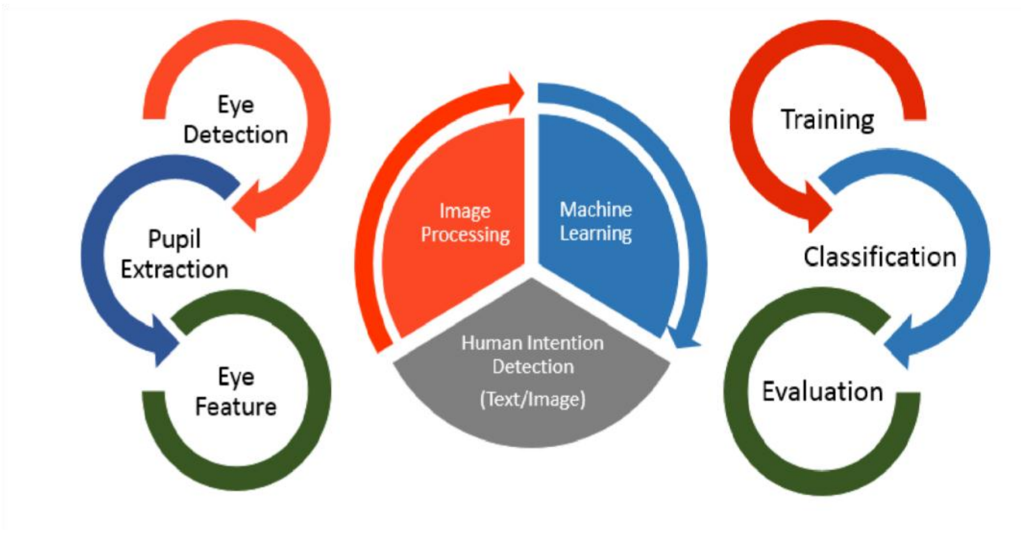


Figure 1: Proposed Method

The proposed system of us is divided into two basic parts. Image processing part and machine learning part. For extracting eye features we first need to detect human eye from the video input. At first we have applied some algorithms to detect the human face from the video input. Then from the detected face we have extracted the possible eye region by doing some heuristics calculation. Then by using some image processing tools we have extracted the pupil position from the possible eye region. After that by measuring the movement of pupil position we calculated the eye features. We have worked with features like Fixation time, Fixation point, Movement variation, Maximum Fixation, Minimum fixation, Maximum fixation. These feature values represent distinguishing characteristics between text reading and viewing an image. By this way we have prepared our own dataset and with these dataset we trained our machine [20]. As we are using supervised learning [20] method after training our machine we have prepared some test data for classification. After the classification process we evaluated [20] the machine by measuring the accuracy of right classification. Thus how we prepared our own classification tool for classifying the contents of an interface is text or images. Which in turns defines human intention on a particular interface.

2 Related Works:

Works on human intention recognition is not new. It is a huge area of research. Though we are working on a small domain of human intention several works are done different domains of human intention. Few of them are given below.

2.1 Tobii 1750 Eye Tracking System:

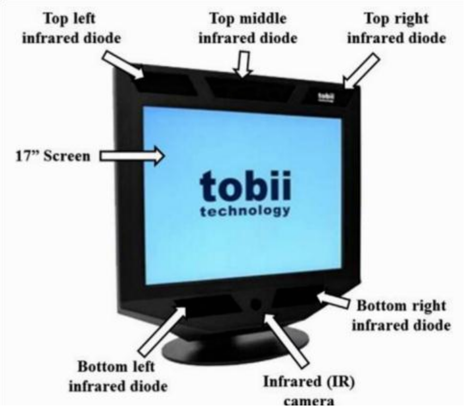


Figure 2: Tobii Eye Tracker

Tobii eye tracking system uses an inferred camera to gain human gaze information. By using the inferred camera it can find several gaze information like.

- Eye ball movement pattern [8].
- Pupil size vibration [8].
- Scan path [8, 6].
- Fixation length [8].
- Area of interest [6].
- Vision theory [6].
- Saccades [6].

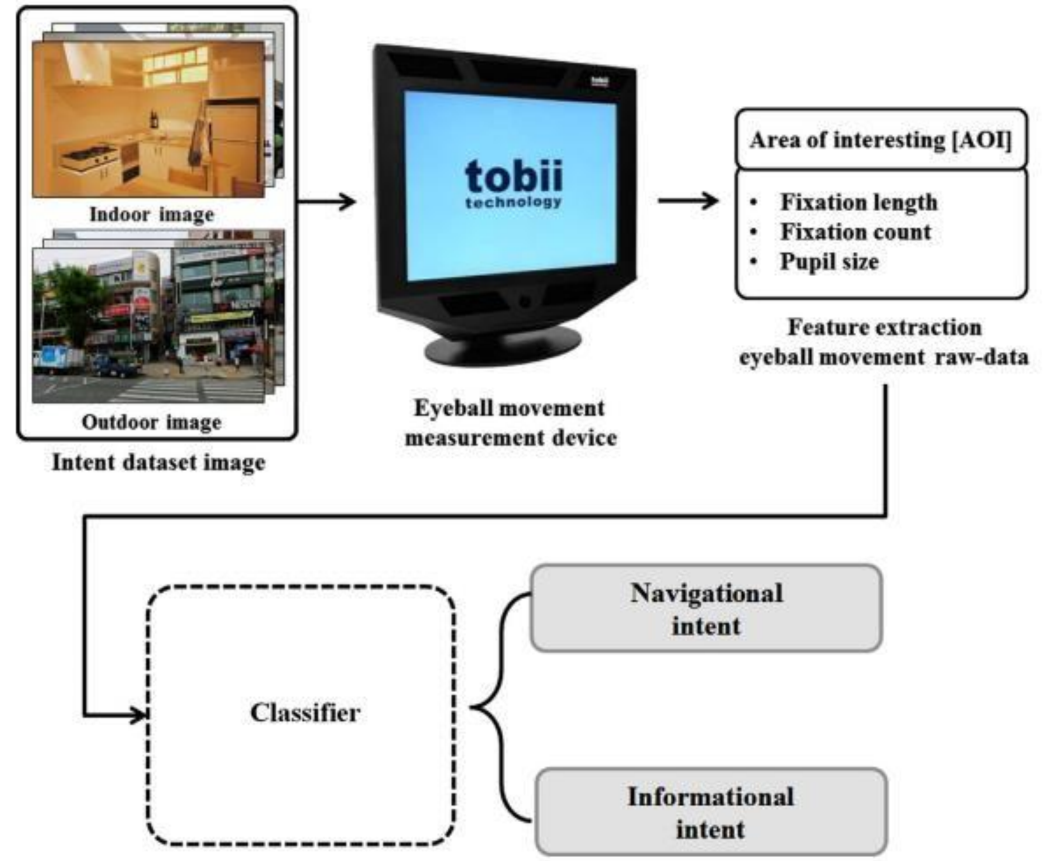


Figure 3: Block Diagram of the Tobii System for Recognizing Human's Implicit Intention

All these informations are directly provided by the Tobii 1750 graze tracking device. Then this informations are analyzed and based on those analysis human intention is detected. It classifies human intention based on navigational intent and informal intent. Navigational intent refers to the idea to find some interesting object in a visual input. And informal intent refers to the movement of the eye to find a particular object of interest. They worked on several domains of human intention. Their main goal was to calculate these feature's values to detect human's implicit intention. This system provides the area of interest based on the classification. The accuracy rate of classification is very high. As Tobii technology uses inferred camera to extract eye features the feature values are accurate. But the consequence of using this device is cost. It is an expensive device which is not available for regular research.

2.2 Eye Gaze Tracking Glasses:

Figure 4: Eye Gaze Tracking Glass

This is a head mounted device. People need to wear this device which has sensors attached with it. Sensors are used to take gaze information. This device has sensors which can detect information like.

- Eye movement detection [15].
- Fixation point extraction [15].
- Track out coordinate points when eye is focusing on the screen [15].

After getting information from the sensors these data are used for further analysis. Again using this device creates a problem of carrying an extra burden. In some cases that can create problems.

3 Background knowledge:

3.1 Definitions:

3.1.1 Saccades:

A saccade is a quick simultaneous movement of both eyes between two phases of fixation in the same direction. Saccades means the movement of eye between two fixation points.

3.1.2 Fixation:

Fixation or visual fixation is the maintaining of visual gaze [7,10,11,13] on a single location. Whenever someone finds something of his interest his eyes fixes at that direction this is actually what is called fixation.

3.1.3 ROI:

A region of interest is a selected subset of samples within a dataset identified for a particular purpose. Here the possible position of the human eye is the region of our interest [6].

3.1.4 Eye movement:

Eye movement refers to voluntary or involuntary movement of the eye, helping in acquiring, fixating and tracking visual stimuli. The path followed while the movement of eye is the movement path [21].

3.1.5 Fixation Time:

Fixation time is the time until the visual gaze [7,10,11,13] is on a single location. It is the time until someone is watching a particular object.

3.1.6 Movement Variation:

Movement variation is the movement pattern of the eye while watching different kinds of objects. Like while someone reads a document the movement of the eye is more like horizontal movement rather than vertical movement. In the case of images the movement is more scattered.

3.1.7 Filters and Preprocessing:

Filtering is used for different kinds of image processing purpose. To improve the quality of the image and to find out the ROI area we have used several kind of filters

3.1.7.1 Gray Scale Image:

A gray scale digital image is image with the pixel value of each pixel is a single sample, that is, it carries only intensity information. Image of this sort is also known as black and-white image [21].

3.1.7.2 Negative Image:

Negative image is a kind of image in which the dark part of the original image will look brighter and the brighter part will look darker [21].

3.1.7.3 Gaussian Filter:

Gaussian filter is filter whose impulse response is a Gaussian function. Its output is a smooth image with less noise. This is mainly used to remove salt paper noises from the image [21].

3.1.7.4 Hough Transform:

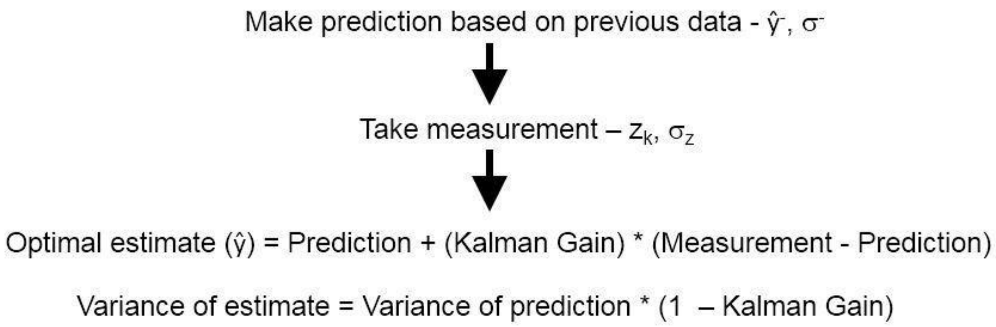
The Hough Transform [2,23] is a feature extraction technique used to image analysis, computer vision and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that explicitly constructed by the algorithm for computing the Hough transform.

3.1.7.5 Histogram Equalization:

Histogram equalization is a method in image processing of contrast adjustment. Its output is usually a high contrast image. It is used to improve the quality of the image. The brightness of the image is very important to detect any object from the image. Using histogram equalization detection can be done in a low lightning condition [21].

3.1.7.6 Kalman Filter:

Kalman Filter [17] is a recursive data processing algorithm. It generates optimal estimate of desired quantities given the set of measurements. For a linear system and Gaussian errors, Kalman filter gives the best estimate based on the previous measurements. The conceptual overview of Kalman filter can be visualize by this diagram,



There are some basic equations which are used to calculate the estimated value in Kalman filter. As it provides an estimated value based on the previous measurement it is used to smoothen the movement of the eye. Equations are,

Prediction (Time Update)

(1) Project the state ahead

$$\hat{y}^-_k = Ay_{k-1} + Bu_k$$

(2) Project the error covariance ahead

$$P^-_k = AP_{k-1}A^T + Q$$

Correction (Measurement Update)

(1) Compute the Kalman Gain

$$K = P^-_k H^T (HP^-_k H^T + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{y}_k = \hat{y}^-_k + K(z_k - H\,\hat{y}^-_k)$$

(3) Update Error Covariance

$$P_k = (I - KH)P^-_k$$

Figure 5 Equations of Kalman Filter

3.1.8 Calibration:

In different lightning conditions threshold values regarding image processing is different. So for different lightning conditions we need to calibrate the system so that it can detect the eye pupil position and track the eye movements. There are several threshold values for different image processing techniques. There are thresholds like binary threshold, canny threshold, and accumulator threshold, minimum and maximum radius of the detected circle. These values change due to change in the lightning condition. For this purpose for using the system threshold values must be set to its appropriate values based on the lightning condition. This is known as calibration here.

3.1.9 Feature Extraction:

Feature extraction [20] one of the main parts of machine learning. Features are unique characteristics of different models. Several features are extracted using sensors to differentiate among several models. To accurately differentiate among models the feature extractor needs to extract unique features from the feature space that properly differentiate them. A feature must have several properties. Like

- The features must be invariant. Which means that features must not change their values because of the absolute location of the object.
- Again they must be invariant to translation, whether horizontal and vertical.
- Rotation is also invariant to classification, so we would like the features to be invariant to rotation.
- We also want features to be invariant to scale. That means size of the object must not impact the values of the features. Objects of different sizes and models must be classified accordingly.
- Occlusion means details of an object is hidden by another part of the object. We must not take those details as our features. Because due to occlusion those parts might not be visible to the sensors in some case. Which will create the problem of not finding the desired feature value.
- If the distance between the object and the sensor can change then the image is subject to projective distortion. Those features should not be taken into

consideration which cannot differentiate among models due to this projective distortion.

3.1.10 Classifier:

A classifier [20] is used to classify objects into models. The design of pattern recognition system usually entails the repetition of number of different activities like, data collection, feature choice, training, and evaluate.

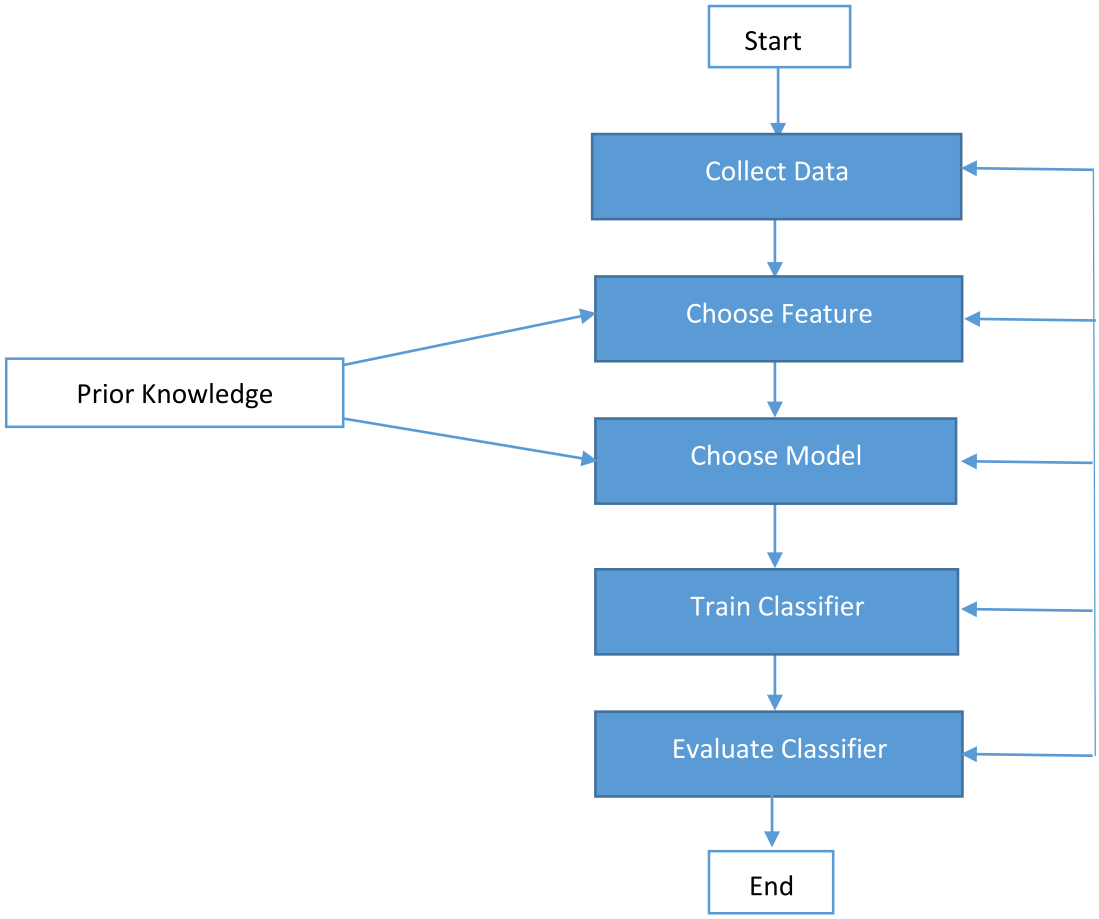


Figure 6: The Design of a Pattern re-cognition System

The very first task of the classifier is to collect data. The data is taken using sensor. Based on the classification system different types of sensor are used.

Then according to the feature properties feature extractor extracts the features from the feature space. Features are extracted so that the performance of the classifier could increase. Again the number of features must be chosen carefully whether there is any redundant feature or not that has no impact on the performance of the classifier.

Again models or classes are defined to which the objects needed to be classified. Classes are defined based on the requirements. Classifier can be of two classes to multiple classes.

Then the classifier must be trained with training dataset. In general, the process of using data to determine the classifier is referred to as training the classifier.

Based on the training data the new objects are classified. Finally the evaluation of the classifier depends on the rate to which the classifier can correctly classify. Evaluation is important both to measure the performance of the system and to identify the need for improvements in its components.

3.1.11 Dataset matching:

Dataset matching [20] means matching the input with the training data. Using machine learning we need to build our dataset. Then we can match the input data with those training dataset to find out the intention [8,9,10] of the people. Here for detecting the eye we have used an existing dataset. Here one harcascade is used to find the human face and then from that we have used another harcascade to find the left and right eye region from the face. Then for detecting the text or pictorial information from an interface we have extracted some eye features from the feature space. Here two classes are defined of which one represents textual information and one represents visual information. Based on the feature values and class levels we have trained the classifier using training samples. After that new test data is used to find whether the classifier could classify the test data by matching them with the training dataset. By this we can also evaluate the performance of the classifier.

3.2 Previous Work:

Previously we have worked with a project named Eye Pointer. Which also takes gaze information. That system was designed to enable physically impaired people to use computer. It also ensures a more interactive way to communicate with a computer rather than using a mouse or keyboard.

3.2.1 Eye Pointer:

Human computer interaction is a vital issue for researchers. Devices we know to interact with a computer are mainly mouse and keyboard. Computer mouse is arguably the most fundamental element that lets us interact with our computers. It's been with us for several years now and we heavily rely on it to perform everyday computer tasks. At times computers were only used by the programmers. Now almost every house has a computer and we are using it to do our daily tasks. As the usage of computer has increased along with the number of users so the interaction devices should come up with advanced and user friendly techniques. Again human attention and intention are closely related to eye and head gestures and voice commands. Human intention can be expressed with auditory and visual information. Works on this topic is not very old. Researchers are working on this topic to introduce new way of computer interaction. From an experiment it is proved that the reaction time (RT) latencies of the hand is slower that the RT latencies of the eye and head when the subject had to make a button press response with either the index or middle finger of the right hand dependent upon whether the stimulus occurred to the right or left of the control fixation point [23].

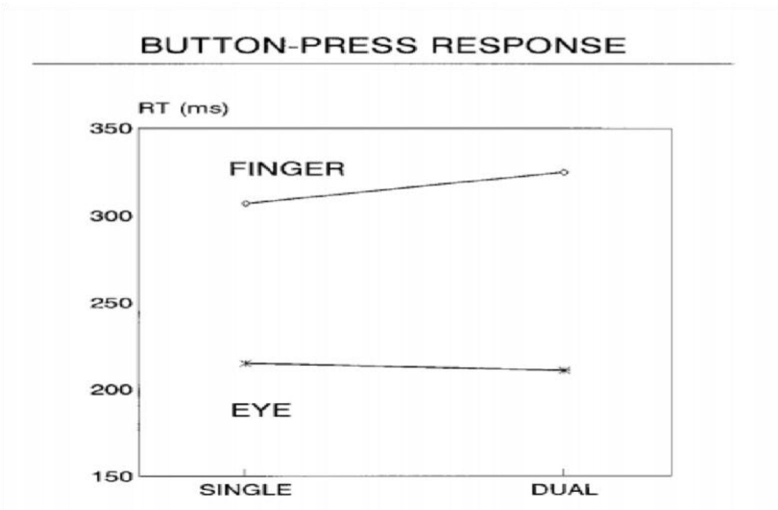


Figure 7 : Mean RT latencies of eye and button press response

Again the number of computer users are increasing day by day. Among the huge population of the world the number of disabled people are of concern. Among them AMPUTEES (people who lost hands in accident) and ARCHEIROPODIA (born without hands and feet) are in major numbers cannot have the facilities of using a computer because of hands as the only interaction method with these traditional input devices requires pair of working hands. For the people who does not have hands there is no efficient solution for them to interact to the computer. Aged people also face problems to use current input devices. Considering all these facts we are introducing a new way of interaction with a computer. In our system we are only using a webcam and a microphone. Webcam to take the head and eye gestures and synchronizing mouse pointer with it. Microphone is used to take voice command which is interpreted as command for the computer. As eye and head gestures along with voice commands are used to interact with computers this interaction method is fast more like interacting in real time. For the physically disabled people this can be a one and only way of interacting with a computer. Our system is more user friendly than using a traditional mouse and keyboard so we believe that out interaction technique will bring a new remark.

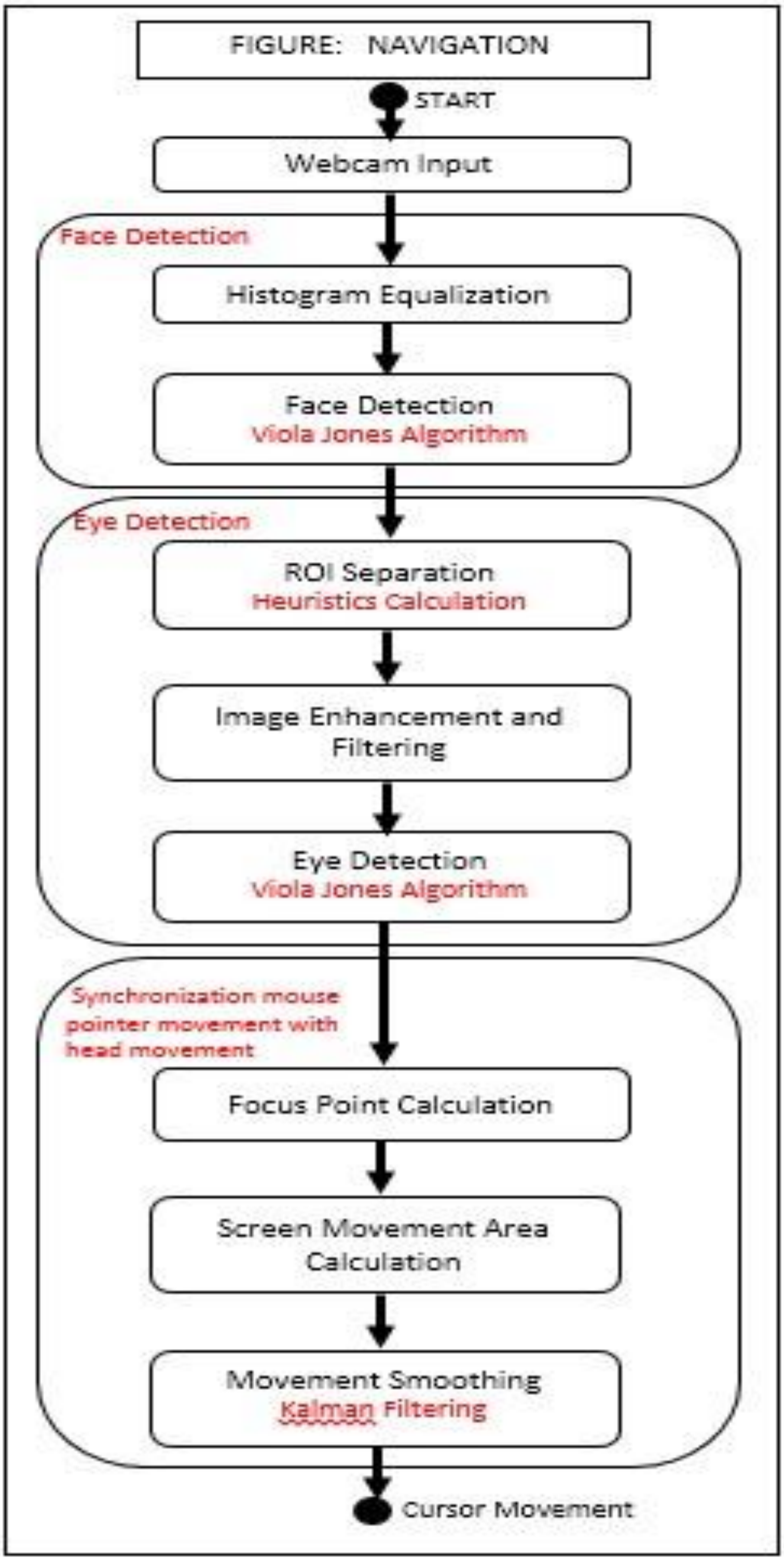
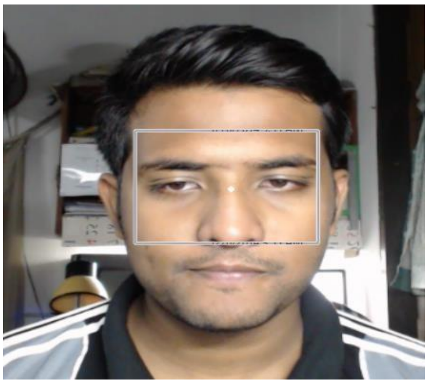


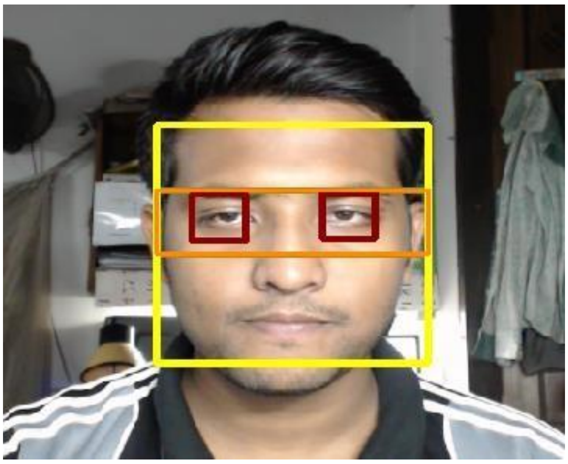
Figure 8: System Architecture of Eye Pointer

Face detection methodology is applied by using viola jones algorithm [18]. It is a machine learning based approach. This technique employs a Haar-features based approach for object detection, which makes the rapid and accurate object detection possible. In our case we have used a sample dataset provided by Intel which includes thousands of negative and positive facial image information. For enhancing the poor image input taken by the webcam and contrast stretching Histogram Equalization [21] methodology is applied first.

Eye detection is followed by face detection. After face detection methodology a heuristic calculation is applied to calculate the possible area of eye within the detected face region. This area is named as Region of Interest (ROI).In this ROI image enhancement and several filtering methods are applied. After that Viola Jones [18] algorithm is again applied. This time haar feature based classifiers are used for eye detection. We have used a dataset provided by intel for eye detection. Face and eye detection is a complex procedure, require much computations to solve this classification problem. Our approach has proved to be effective in order to real-time processing of the frames.



After completion of the detection process synchronization mouse pointer movement with head movement is needed. For that we have considered a focus point calculation mechanism. We have considered the focus point as the middle point of the two eyes. The considered focus point is synchronized later as the mouse pointer



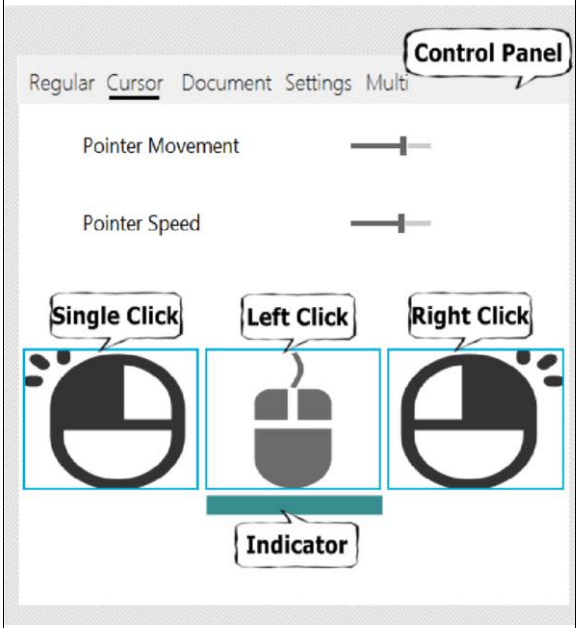
Screen movement area calculation is the base region, inside where we have taken the relative focus point movement. This area is the subdivision of the total screen resolution. The movement of the focus point inside this region is multiplied with a factor to interpret that point throughout the total screen resolution. As a result the little movement of the focus point inside the region is interpreted as the cursor movement within the screen resolution. Though this rapid detection and calculation process in real time is very much effective, there are some mechanical noises which result in some disturbance in the movement path of the cursor. Again we have found blinking effect. Therefore, accuracy in the navigation process decreases. To compensate with this we have implemented a set of mathematical equations that provides an efficient computational means to estimate the state of the movement states. This computational process is known as Kalman filtering [17]. It supports estimation of past, present and future states and it can do so even when the precise nature of the modeled system is unknown. Finally, after applying Kalman filtering we have got a very smooth movement path results in greater accuracy in the navigation process.

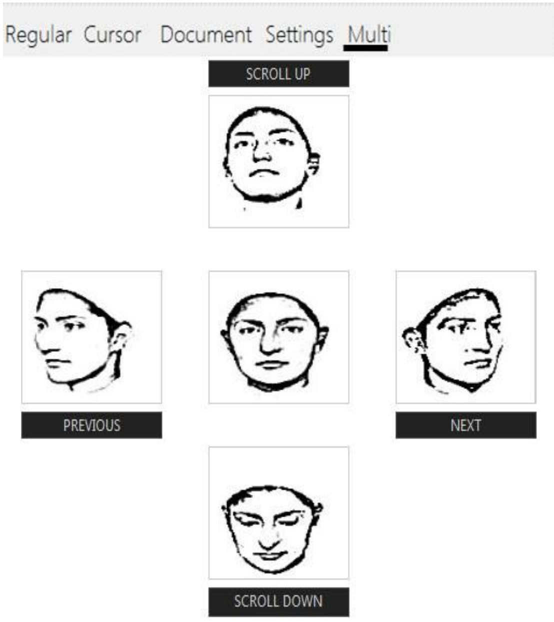
3.2.2 The main interface of desktop application:

Figure 9: System Interface

Indicator	Functionalities	Interface
1	• Play button which switch the system to on mode. • It enables the video to take input and do the processing. • User can use the system for interaction.	

2	• Pause button which switch the system to off mode. • It restrict the Webcam to render video and do the processing. • Pauses the system to normal mode when user can use mouse to do the interaction.	
Control Panel	• Shows different settings. • First chosen mode is regular mode contains two switches to switch eye and blink control. • Blink control activates click events using eye blinks.	

3	• Switches to cursor control mode. • In this mode user can navigate through computer by pointing the mouse pointer with the head movement. • In this mode mouse pointer movement depends on the head movement. • Pointer movement and pointer speed controls the cursor movement & speed. • Different shows the clicks that has been done using eye blinks when blink control is activated.	

4	• Switches system to multimedia mode. • In this mode user controls the multimedia operations. • Different functionalities are done using head gestures. • Left movement of head causes rewind of a video. • Right movement of head causes fast forward of a video. • For scrolling down any document head down gesture is used. • For scrolling up any document head up gesture is used.	

<table>
<tr>
<td>5</td>
<td>

- This is security setting panel of our security system

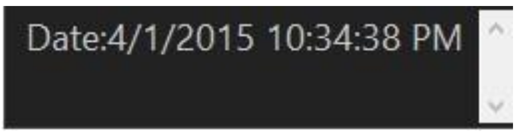
- Here you can see the latest uploaded photo of the person who wanted to access your computer.

- There are two buttons to show previous and next photos saved on the server.

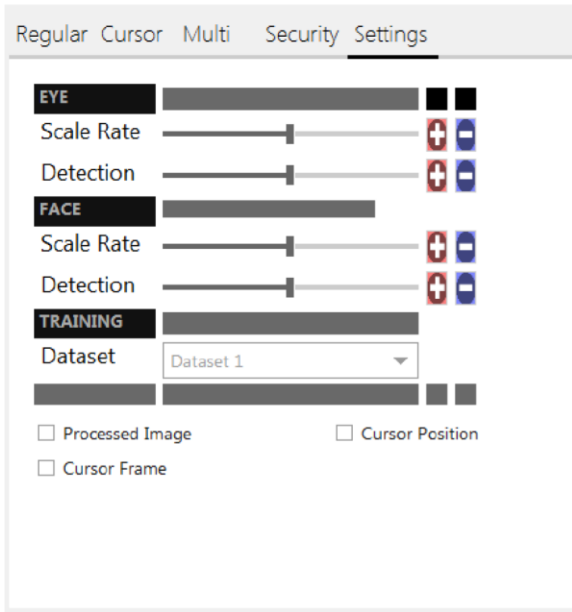
</td>
<td>


</td>
</tr>
<tr>
<td>5</td>
<td>

- This is the box showing the date and time when the photo is captured

</td>
<td>


</td>
</tr>
</table>

5	• Here you can enable the SMS option • You can provide the number to which you want to send SMS when there is any unauthorized access.	
6	• Switches to setting panel. • Different setting options is provided here for system manipulation.	
8	• Indicates which mode is activated.	

It is not very old that researchers are working on building a computer interaction technique that will make the interaction methodology more user friendly and less expensive. We are introducing a versatile and unique technique of computer interaction. This kind of system is not introduced previously. Our system is providing whole computer navigation facility along with the security of the computer. So this kind of system will bring a new remark on computer navigation and security. We designed the system keeping in mind human cognition, intention and normal interaction techniques that a person follows while interacting in his or her daily life. Computer navigation is done by taking the gaze information of the human head and eye and the commands are passed through voice input just like the way a person expresses his or her intentions. Our system has two major parts. Desktop part and Mobile app.

The desktop part is divided into three basic modes to make the computer navigation system more user friendly. There are different functionalities in different modes.

The first mode is named as the Cursor control mode. In this mode you can simply navigate throughout the computer. You can move the cursor with eye and head gestures. Here basic voice commands are implemented to do click options. You can single click or double click or right click in this mode. You can perform click events with voice commands as well as eye blinks. There is a specific voice command to activate this mode. If you activate this mode an activation confirmation feedback will be returned from the system. So you will be able to know in which mode you are currently working on.

The second mode is named as Multi mode. This mode is mainly introduced for multimedia purpose. First of all you need to activate this mode by giving specific voice command. Different voice commands are introduced here for different multimedia purpose. Here you can watch video clips with gestures. You can fast forward the video or rewind the video using gestures. Other voice commands are introduced to do different tasks. There are voice commands for adjusting volume or making full screen. There are commands to stop play and pause the video. So now the users can enjoy and express his desire in a more flexible way. In this mode you can also view photo albums using head gestures. You will be able to read pdf files by sliding it up and down using gestures. So it makes the reading a more flexible and interesting technique.

The last one is the security mode. A mobile app is necessary for this mode. This feature is a unique addition of our system. By activating this mode you can ensure the security of your computer. For activation you need to give the specific voice command. After that there is a voice password which must be matched to activate this mode. Whenever this mode will be activated your computer will be in a stealth situation. If anyone tries to access your computer it will capture the photo of that person and upload it into a server. The user will be notified through sms message into his cell phone. This is where the mobile app comes into action. Using this mobile app user can see what’s going on in front of his pc by watching the webcam stream of his computer. He can also watch what’s going on his monitor. This unique feature is introduced in our system which will ensure the safety of your computer. Now the user can take necessary action like lock or shutdown the computer anytime by using this app.

4 Proposed Approach and Research Methodology:

4.1 System Architecture:

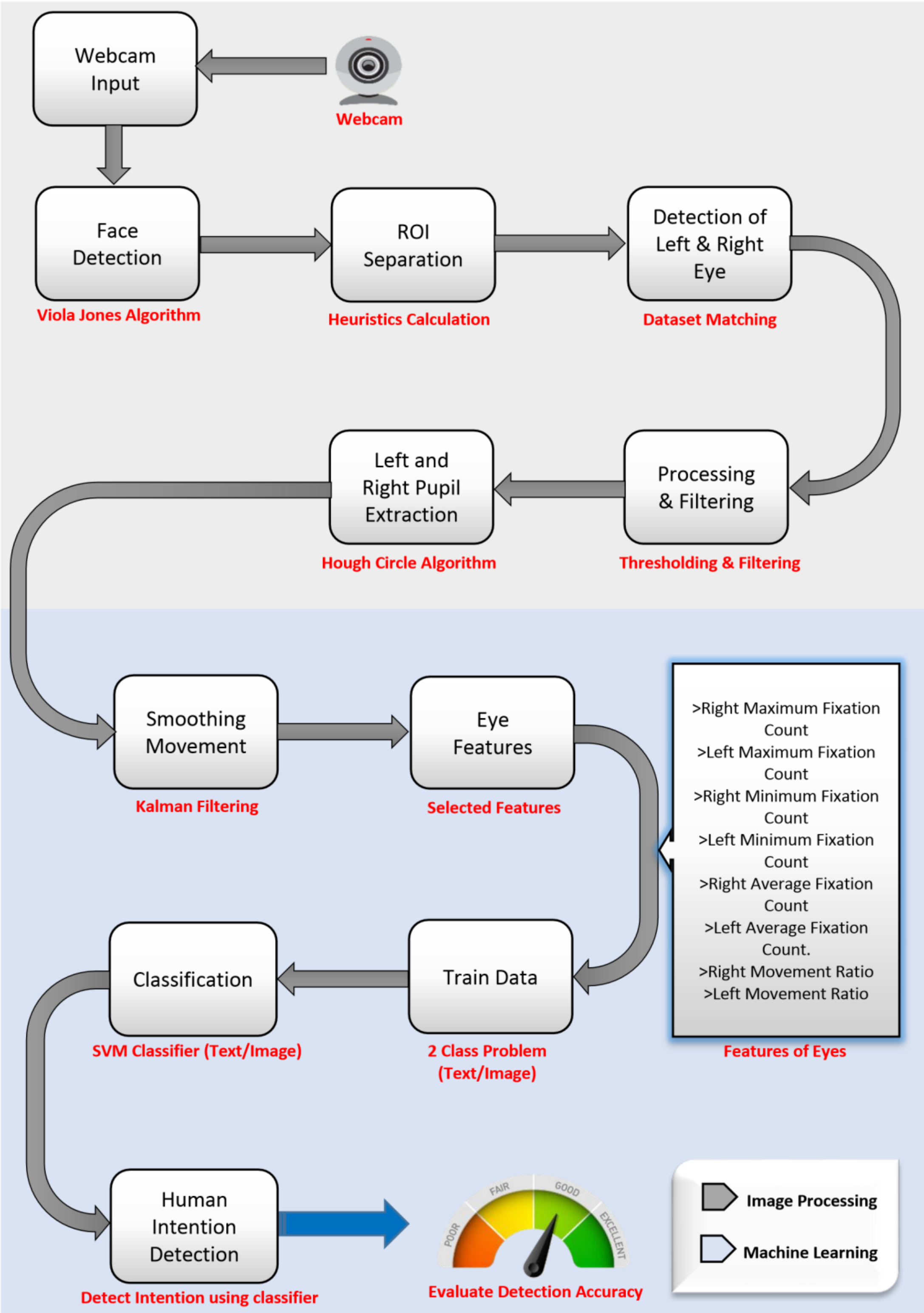


Figure 10: System Architecture

4.2 Image Processing:

Image processing part of this research includes detection of face and eyes, tracking iris and pupil for both left and right eye, image pre and post processing techniques.

4.2.1 Input Image:

The process begins with taking input image from an ordinary webcam. As the main challenge in this research is to work with standard resolution image. We have considered image resolution as 800x600. The frame rate is set to standard as 30 fps. Webcam is set on the top of the monitor so that the face of the user can be recorded.

Figure 11: Input from Webcam

4.2.2 Face Detection :

A face detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. One natural framework for considering this problem is that of binary classification, in which a classifier is constructed to minimize the misclassification risk. Since no objective distribution can describe the actual prior probability for a given image to have a face, the algorithm must minimize both the false negative and false positive rates in order to achieve an acceptable performance. This task requires an accurate numerical description of what sets human faces apart from other objects. It turns out that these characteristics can be extracted with a remarkable committee learning algorithm called Adaboost [20], which relies on a committee of weak classifiers to form a strong one through a voting mechanism. A classifier is weak if, in general, it cannot meet a predefined classification target in error terms. An operational algorithm must also work with a reasonable computational budget. Techniques such as integral image and attentional cascade make the

ViolaJones algorithm [18] highly efficient: fed with a real time image sequence generated from a standard webcam, it performs well on a standard PC.

The **Viola-Jones** [18] algorithm uses Haar-like features, that is, a scalar product between the image and some Haar-like templates. More precisely, let I and P denote an image and a pattern, both of the same size N × N (see Figure 12). The feature associated with pattern P of image I is defined by

$$\sum_{1\leq i\leq N}\sum_{1\leq j\leq N} I(i,j)1_{P(i,j)\text{ is white}} - \sum_{1\leq i\leq N}\sum_{1\leq j\leq N} I(i,j)1_{P(i,j)\text{ is black}}.$$

To compensate the effect of different lighting conditions, all the images should be mean and variance normalized beforehand. Those images with variance lower than one, having little Information of interest in the first place, are left out of consideration. In practice, five patterns are considered. The derived features are assumed to hold all the information needed to characterize a face. Since faces are by and large regular by nature, the use of Haar-like patterns seems justified. There is, however, another crucial

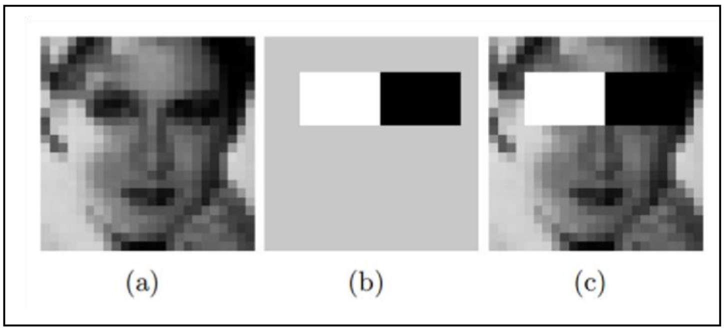


Figure 12: Haar like features. Here as well as below, the background of a template like (b) is painted

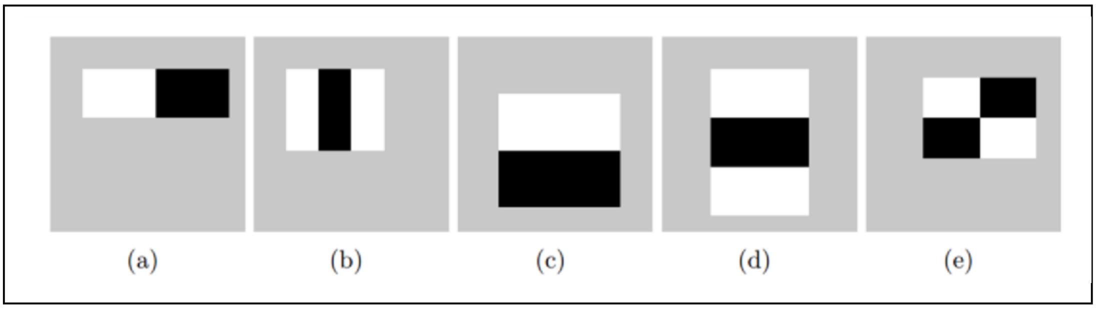


Figure 13: Five Haar-like patterns. The size and position of a pattern's support can vary provided its black and white rectangles have the same dimension, border each other and keep their relative positions. Thanks to this constraint, the number of features one can draw from an image is somewhat manageable: a 24 × 24 image, for instance, has 43200, 27600, 43200, 27600 and 20736 features of category (a), (b), (c), (d) and (e) respectively, hence 162336 features in all.

element which lets this set of features take precedence: the integral image which allows to calculate them at a very low computational cost. Instead of summing up all the pixels inside a rectangular window, this technique mirrors the use of cumulative distribution functions. The integral image II of I

$$\mathrm{II}(i,j) := \begin{cases} \sum_{1\le s\le i} \sum_{1\le t\le j} I(s,t), & 1 \le i \le N \text{ and } 1 \le j \le N \\ 0, & \text{otherwise} \end{cases},$$

is so defined that

$$\sum_{N_1\le i\le N_2} \sum_{N_3\le j\le N_4} I(i,j) = \mathrm{II}(N_2,N_4) - \mathrm{II}(N_2,N_3-1) - \mathrm{II}(N_1-1,N_4) + \mathrm{II}(N_1-1,N_3-1), \quad (1)$$

Holds for all N1 ≤ N2 and N3 ≤ N4. As a result, computing an image's rectangular local sum requires at most four elementary operations given its integral image. Moreover, obtaining the integral image itself can be done in linear time: setting N1 = N2 and N3 = N4 in (1), we find

$$I(N_1,N_3) = \mathrm{II}(N_1,N_3) - \mathrm{II}(N_1,N_3-1) - \mathrm{II}(N_1-1,N_3) + \mathrm{II}(N_1-1,N_3-1).$$

4.2.3 Feature Selection with Adaboost:

Some terminology. A classifier maps an observation to a label valued in a finite set. For face detection, it assumes the form of f : R d 7→ {−1, 1}, where 1 means that there is a face and −1 the contrary (see Figure 14) and d is the number of Haar-like features extracted from an image. Given the probabilistic weights w· ∈ R+ assigned to a training set made up of n observation-label pairs (xi , yi), Adaboost [20] aims to iteratively drive down an upper bound of the empirical loss

$$\sum_{i=1}^{n} w_i \mathbf{1}_{y_i \neq f(x_i)},$$

Under mild technical conditions (see Appendix A). Remarkably, the decision rule constructed by Adaboost [20] remains reasonably simple so that it is not prone to over fitting, which means that the empirically learned rule often generalizes well. For more details on the method, we refer to [2, 3]. Despite its groundbreaking success, it ought to be said that Adaboost [20] does not learn what a face should look like all by itself because it is humans, rather than the algorithm, who perform the labeling and the first round of feature selection, as described in the previous section.

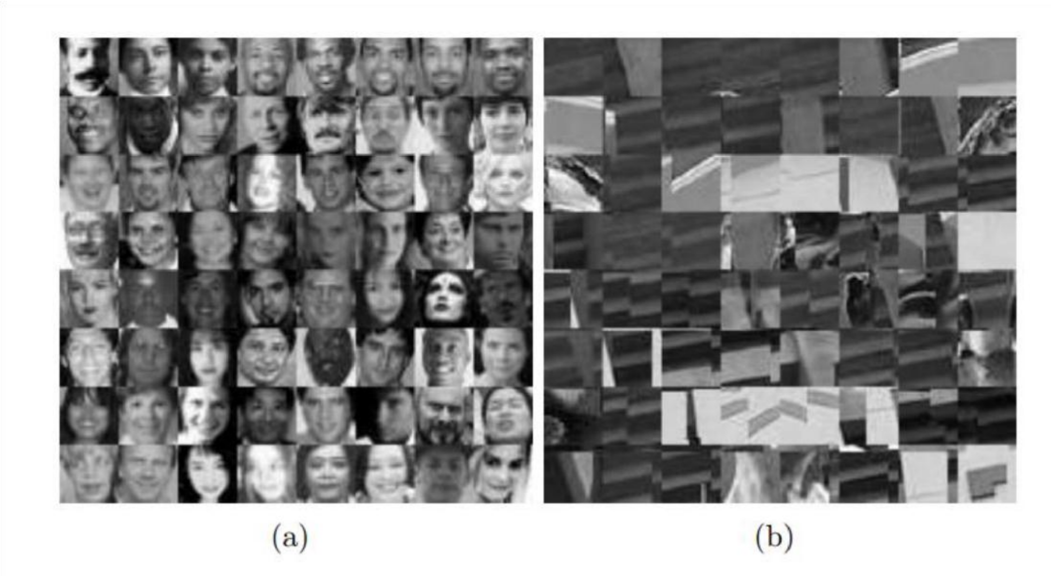


Figure 14: Some supervised examples: (a) positive examples (b) negative examples. All of them are 24 × 24 grayscale images.

The building block of the Viola-Jones [18] face detector is a decision stump, or a depth one decision tree, parameterized by a feature f ∈ {1, · · ·, d}, a threshold t ∈ R and a toggle T ∈ {−1, 1}. Given an observation x ∈ R d, a decision stump h predicts its label using the following rule

$$h(x) = (1_{\pi_f x \geq t} - 1_{\pi_f x < t})\mathcal{T} = (1_{\pi_f x \geq t} - 1_{\pi_f x < t})1_{\mathcal{T}=1} + (1_{\pi_f x < t} - 1_{\pi_f x \geq t})1_{\mathcal{T}=-1} \in \{-1, 1\}, \qquad (2)$$

where πfx is the feature vector's f-th coordinate. Several comments follow:

1. Any additional pattern produced by permuting black and white rectangles in an existing pattern (see Figure 13) is superfluous. Because such a feature is merely the

opposite of an existing feature, only a sign change for t and T is needed to have the same classification rule.

2. If the training examples are sorted in ascending order of a given feature f, a linear time exhaustive search on the threshold and toggle can find a decision stump using this feature that attains the lowest empirical loss

$$\sum_{i=1}^{n} w_i \mathbf{1}_{y_i \neq h(x_i)}, \tag{3}$$

on the training set. Imagine a threshold placed somewhere on the real line, if the toggle is set to 1, the resulting rule will declare an example x positive if πfx is greater than the threshold and negative otherwise. This allows us to evaluate the rule's empirical error, thereby selecting the toggle that fits the dataset better. Since margin

$$\min_{i:\, y_i=-1} |\pi_{\mathfrak{f}} x_i - \mathfrak{t}| + \min_{i:\, y_i=1} |\pi_{\mathfrak{f}} x_i - \mathfrak{t}|,$$

and risk, or the expectation of the empirical loss (3), are closely related [3, 6, 7], of two decision stumps having the same empirical risk, the one with a larger margin is preferred . Thus in the absence of duplicates, there are n + 1 possible thresholds and the one with the smallest empirical loss should be chosen. However it is possible to have the same feature values from different examples and extra care must be taken to handle this case properly.

By adjusting individual example weights, Adaboost [19] makes more effort to learn harder examples and adds more decision stumps in the process. Intuitively, in the final voting, a stump ht with lower empirical loss is rewarded with a bigger say when a T-member committee (vote-based classifier) assigns an example according to

$$f^T(\cdot) = \text{sign}\left[\sum_{t=1}^{T} \alpha_t h_t(\cdot)\right].$$

Figure 15 shows an instance where Adaboost [19] reduces false positive and false negative rates simultaneously as more and more stumps are added to the committee. For notational simplicity, we denote the empirical loss by

$$\sum_{i=1}^{n} w_i(1) 1_{y_i \sum_{t=1}^{T} \alpha_t h_t(x_i) \leq 0} := \mathbb{P}(f^T(X) \neq Y),$$

where (X, Y) is a random couple distributed according to the probability P defined by the weights wi(1), $1 \leq i \leq n$ set when the training starts. As the empirical loss goes to zero with T, so do both false positive $P(f^T(X) = 1|Y = -1)$ and false negative rates $P(f^T(X)=-1|Y = 1)$ owing to

$$\mathbb{P}(f^T(X) \neq Y) = \mathbb{P}(Y = 1)\mathbb{P}(f^T(X) = -1|Y = 1) + \mathbb{P}(Y = -1)\mathbb{P}(f^T(X) = 1|Y = -1).$$

Thus the detection rate must tend to 1.

$$\mathbb{P}(f^T(X) = 1|Y = 1) = 1 - \mathbb{P}(f^T(X) = -1|Y = 1),$$

Thus the size T of the trained committee depends on the targeted false positive and false negative rates. In addition, let us mention that, given n− negative and n+ positive examples in a training pool, it is customary to give a negative (resp. positive) example an initial weight equal to 0.5/n− (resp. 0.5/n+) so that Adaboost [19] does not favor either category at the beginning.

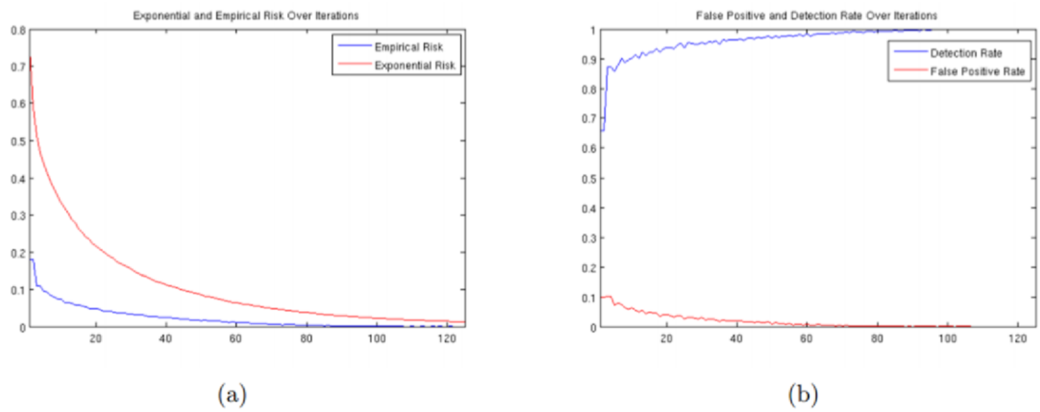


Figure 15: Figure (a) shows that the empirical risk and its upper bound, interpreted as the exponential loss (see Appendix A), decrease steadily over iterations. This implies that false positive and false negative rates must also decrease, as observed in (b).

4.2.4 ROI Selection:

Face detection is a preprocess for finding eyes. Eye confined in a small region inside detected face. For reducing the search space, the region where eyes can be found have to be predefined. In real time as the head is moving, eyes reside on that frame will also move.

So, dynamic tracking of eyes with head is needed. We have developped heuristic calculation, by which we can predict an area where eyes can be.

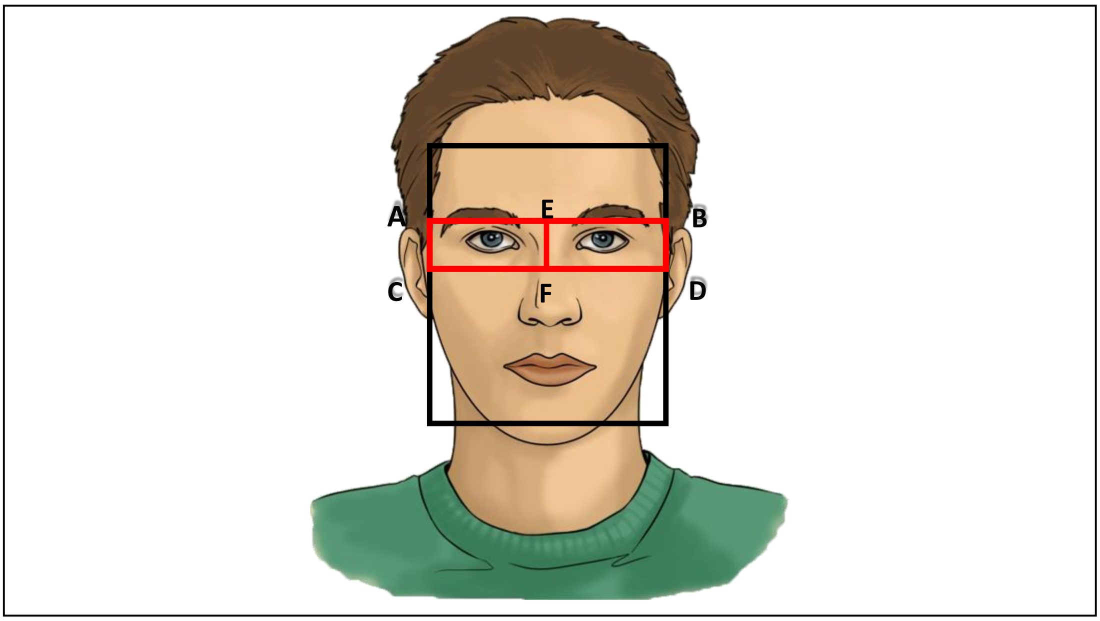


Figure 16: Possible ROI region where eyes can be within the detected face

From “Figure 16” it is clear that we have to find the region enclosed by A, B, C, and D. In 2d space each point (A, B, C, D) has x and y coordinate values. We have calculated the points from the following methods:

YCoordStartSearchEyes = face.rect.Top + (face.rect.Height * 3 / 11);

StartingPointSearchEyes (A) = Point (face.rect.X, YCoordStartSearchEyes);
EndingPointSearchEyes (B) = new Point ((face.rect.X + face.rect.Width), yCoordStartSearchEyes);
SearchEyesAreaSize = new Size (face.rect.Width, (face.rect.Height * 2 / 9));

Point lowerEyesPointOptimized = new Point (face.rect.X, yCoordStartSearchEyes + searchEyesAreaSize.Height);

```
Size eyeAreaSize = new Size (face.rect.Width / 2, (face.rect.Height * 2 / 9));
StartingLeftEyePointOptimized = new Point (face.rect.X + face.rect.Width / 2,
yCoordStartSearchEyes);
Rectangle possibleROI_eyes = new Rectangle (startingPointSearchEyes, searchEyesAreaSize);
PossibleROI_rightEye = new Rectangle (startingPointSearchEyes, eyeAreaSize);
PossibleROI_leftEye = new Rectangle (startingLeftEyePointOptimized, eyeAreaSize);
```

From the aforementioned calculation we have detected two regions. The regions enclosed with AEFC from "Figure 16" is defined as the possible region of interest for right eye and the region enclosed EBDF from "Figure 16" is defined as the possible region of interest for left eye.

4.2.5 Eye Detection:

Eye detection follows similar methodology to face detection. It follows Viola – Jones Algorithm and Haar like feature matching for detection mechanism (See section 1.2.2 and section 1.2.3 for details). From the previous section we have already extracted the possible region of interest for both left and right eyes. In that reduced space dataset matching is implemented to correctly identify the eye region. As it highly depends on the detected face image, misclassification of face detection may also result in misclassification of eye detection. In every input frame we can get several information: - detected face region, ROI for left and right eyes, detected eye region.

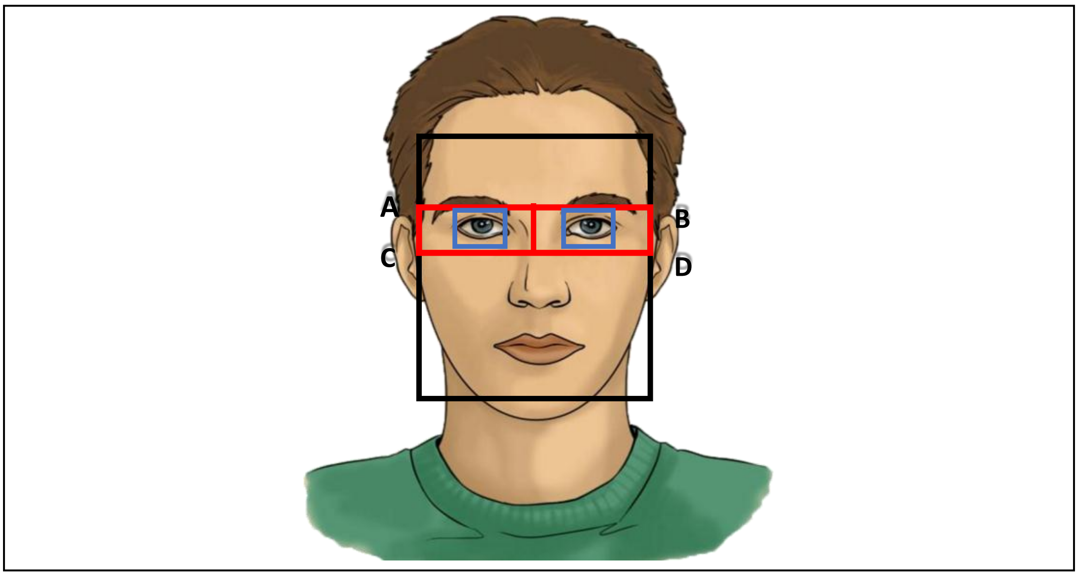


Figure 17: Detected Eye Region (blue square) inside ROI for left and right eye.

4.2.6 Processing and Filtering:

Extracted eye region includes fewer pixels considering the whole image. This need some medication for further processing. Inside eye region iris can be found. Iris confined within a small region of eye, so noisy image of eye will give false positive result for iris detection. Some processing is needed to compensate with the erroneous detection.

Processing & Filtering	Descriptions
➢ Image Inversion	▯ Implemented in Black & White image or grayscale image. ▯ After inversion black becomes white and white becomes black. ▯ Inverted black and white image can be thought of as a digital negative of the original image.
➢ Median filter	▯ Outputs a smooth image. ▯ Removes unnecessary noises from the image.
➢ Gaussian filter	▯ Output is a smooth image. ▯ Removes salt paper noises from the image.
➢ Laplacian filter	▯ Output is a sharper image. ▯ Used to detect edges of an image. ▯ Here used to detect the eye edges.
➢ Histogram equalization	▯ Output is a high contrast image. ▯ Used to improve the quality of the image. ▯ Here we used to improve the contrast of the image. ▯ So that detection can be done in low lightning conditions.

➢ Global Threshold	• Global thresholding is a dynamic way of setting a threshold value. • Used to extract the desired object from the image. • Global thresholding sets an optimal threshold value that can help to differentiate.
➢ Binary Threshold	• Used to separate out regions of an image corresponding to objects which we want to analyze. • Separation is based on the variation of intensity between the object pixels and the background pixels.

4.2.7 Iris and Pupil Detection:

Processed ROI includes iris as well as pupil information. To detect pupil from that region, iris have to be detected first. Again, we know that pupil is the center of iris. So, if we can detect iris, we can also predict the position of the pupil. From the shape point of view we know that iris is circular. Therefore, if we can detect circular object inside the ROI we will have the iris information and from that we can also predict the relevant information of pupil also.

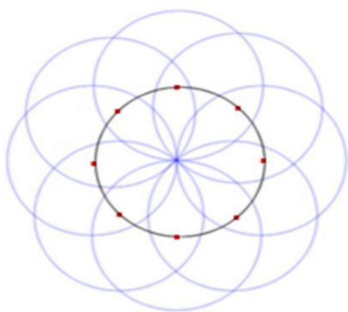


Figure 18: The points (in red) on the circle (in black) each create a ring of votes (in blue) around themselves. Where vote rings overlap the votes combine (more intense blue) and the greatest vote value indicates the center of the circle. The radius of the vote rings is determined by the radius of the circle being searched for.

Circular Hough Transform (CHT) [] is a specified algorithm to detect circle. We assume that the image is taken from above and that the vast majority of craters will be roughly circular. The program then uses a CHT to look for circular patterns of a set radius (perhaps repeating many times for different radii). A CHT essentially takes each image pixel that represents an edge and uses it to generate a vote for all possible circles that the edge could be on. The centers of all these possible image circles form a circle of votes around the edge point. If we do this for every edge point then the votes tend to build up at points that really are the centers of circles. In this procedure we can detect iris region bounded by a circle. The pupil is considered as the center point of the detected iris.

4.3 Machine Learning:

The machine learning part of this research deals with movement smoothing filtering, feature selection and classification problem.

4.3.1 Kalman Filtering:

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects:

It supports estimations of past, present, and even future states, and it can do so even When the precise nature of the modeled system is unknown.

The Kalman filter addresses the general problem of trying to estimate the $X \in R^n$ state of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \qquad (1.1)$$

with a measurement $X \in R^m$ that is

$$z_k = Hx_k + v_k. \qquad (1.2)$$

The random variables w_k and v_k represent the process and measurement noise (respectively).They are assumed to be independent (of each other), white, and with normal probability distributions

$$p(w) \sim N(0, Q), \tag{1.3}$$

$$p(v) \sim N(0, R). \tag{1.4}$$

In practice, the process noise covariance Q and measurement noise covariance R matrices might change with each time step or measurement, however here we assume they are constant.

The nxn matrix A in the difference equation (1,1) relates the state at the previous time step k-1 to the state at the current step k, in the absence of either a driving function or process noise. Note that in practice A might change with each time step, but here we assume it is constant. The nx1 matrix B relates the optional control input $u \in R^l$ to the state x. The mxn matrix H in the measurement equation (1.2) relates the state to the measurement z_k . In practice H might change with each time step or measurement, but here we assume it is constant.

We will begin this section with a broad overview, covering the "high-level" operation of one form of the discrete Kalman filter (see the previous footnote). After presenting this high-level view, we will narrow the focus to the specific equations and their use in this version of the filter. The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the priori estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Indeed the final estimation algorithm resembles that of a predictor corrector algorithm for solving numerical problems as shown below

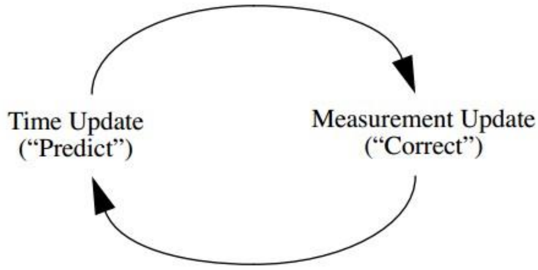


Figure 19: The ongoing discrete Kalman filter cycle. The time update projects the current state estimate ahead in time. The measurement update adjusts the projected estimate by an actual measurement at that time.

The specific equations for the time and measurement updates are presented below

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (1.9)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (1.10)$$

Discrete Kalman filter time update equations.

Again notice how the time update equations in above project the state and covariance estimates forward from time step k-1 to step k. A and B are from (1,1), while Q is from (1,3). Initial conditions for the filter are discussed in the earlier references.

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (1.11)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (1.12)$$

$$P_k = (I - K_k H)P_k^- \quad (1.13)$$

Discrete Kalman filter measurement update equations.

The first task during the measurement update is to compute the Kalman gain, K_k Notice that the equation given here as (1.11) is the same as (1.8). The next step is to actually measure the process to obtain z_k, and then to generate an a posteriori state estimate by incorporating the measurement as in (1.12). Again (1.12) is simply (1.7) repeated here for completeness. The final step is to obtain an a posteriori error covariance estimate via (1.13). After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. This recursive nature is one of the very appealing features of the Kalman filter—it makes practical implementations much more feasible than (for example) an implementation of a Wiener filter [Brown92] which is designed to operate on all of the data directly for each estimate. The Kalman filter instead recursively conditions the current estimate on all of the past measurements.

In the actual implementation of the filter, the measurement noise covariance R is usually measured prior to operation of the filter. Measuring the measurement error covariance R is generally practical (possible) because we need to be able to measure the process anyway (while operating the filter) so we should generally be able to take some off-line sample measurements in order to determine the variance of the measurement noise.

The determination of the process noise covariance Q is generally more difficult as we typically do not have the ability to directly observe the process we are estimating. Sometimes a relatively simple (poor) process model can produce acceptable results if one "injects" enough uncertainty into the process via the selection of Q. Certainly in this case one would hope that the process measurements are reliable.

In either case, whether or not we have a rational basis for choosing the parameters, often times superior filter performance (statistically speaking) can be obtained by tuning the filter parameters Q and R. The tuning is usually performed off-line, frequently with the help of another (distinct) Kalman filter in a process generally referred to as system identification.

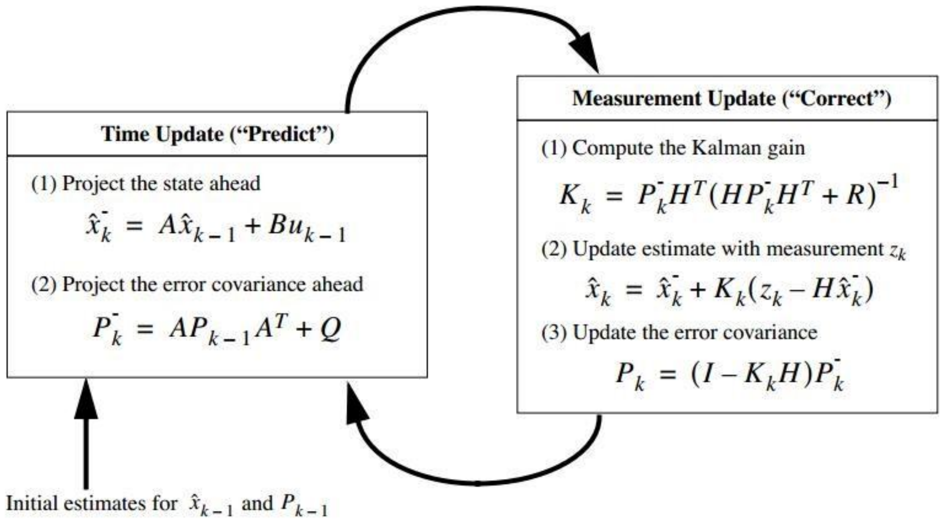


Figure 20: complete picture of the operation of the Kalman filter, combining the high-level diagram of Figure 1-1 with the equations from

In closing we note that under conditions where Q and R .are in fact constant, both the estimation error covariance P_k and the Kalman gain K_k will stabilize quickly and then remain constant. If this is the case, these parameters can be pre-computed by either running the filter off-line, or for example by determining the steady-state value of P_k as described in [Grewal93].

It is frequently the case however that the measurement error (in particular) does not remain constant. For example, when sighting beacons in our optoelectronic tracker ceiling panels, the noise in measurements of nearby beacons will be smaller than that in far-away beacons. Also, the process noise Q is sometimes changed dynamically during filter operation—becoming Q_k —in order to adjust to different dynamics. For example, in the case of tracking the head of a user of a 3D virtual environment we might reduce the magnitude of Q_k if the user seems to be moving slowly, and increase the magnitude if the dynamics start changing rapidly. In such cases Q_k might be chosen to account for both uncertainty about the user's intentions and uncertainty in the model.

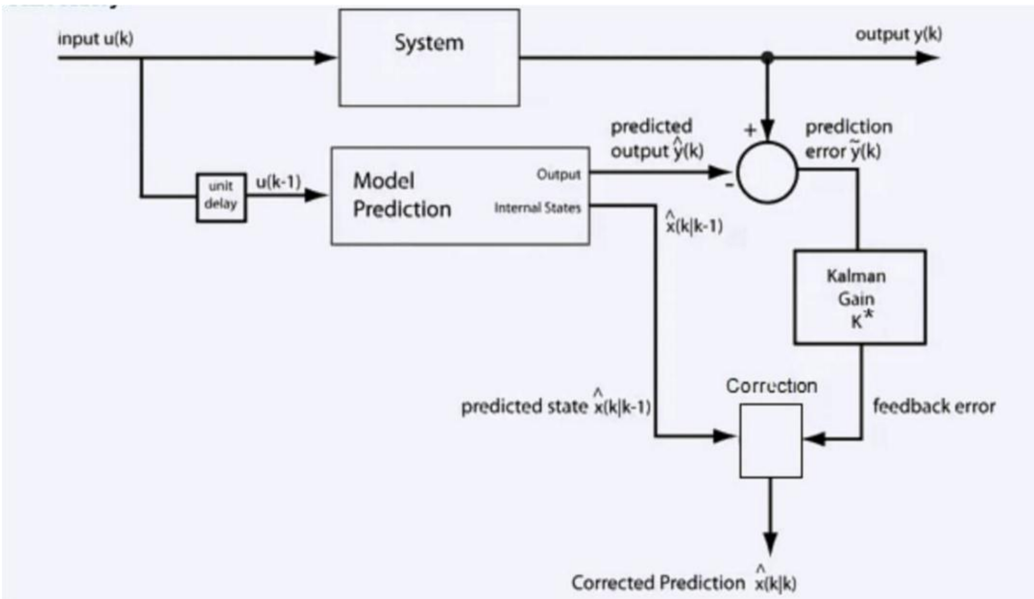


Figure 21: Kalman Filtering Overview

Extracted pupil information indulge with some noises. Movement of pupil in the 2d coordinate space is not smooth enough. We have implemented kalman filtering for smoothing the movement path of pupil. This reduces the noises in a great extent.

4.3.2 Feature Extraction:

After pupil extraction the critical problem is to select feature from the detected pupil. Feature must be distinguishable enough to correctly classify multiple class problem. In our research, the motive is to detect human intention. Main target is to detect either a user is reading a document or he is looking at a picture. So, the classification problem is to determine differences between a textual information from the pictorial information. Therefore, the features selected in our case are:

1. Right eye Maximum Fixation count.
2. Left eye Maximum Fixation count.
3. Right eye Minimum Fixation count.
4. Left eye Minimum Fixation count.
5. Right eye Average Fixation count.
6. Left eye Average Fixation count.

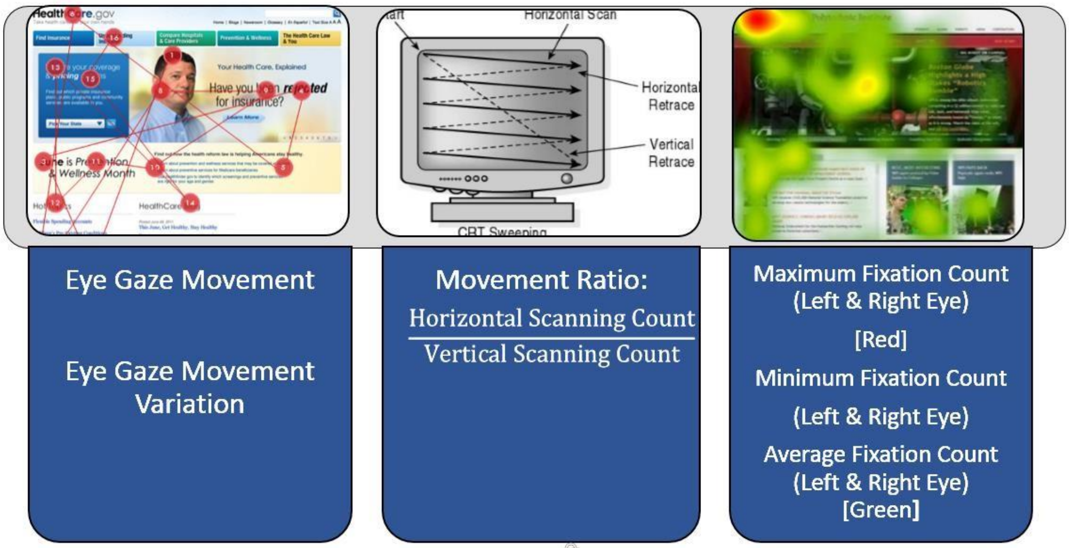
7. Right eye Movement Ratio.

8. Left eye Movement Ratio.

Fixation count defines the focus point in the screen. Maximum fixation count deals with the maximum duration of focus point. Accordingly minimum fixation count means minimum duration of a focus point and average fixation count means averaging all the focus duration onto a particular interface. Research shows that in a textual information user may have small duration of fixation count as when a user read a document, the movement of eye pupil traverse from point to point. Again, in case of pictorial information a user try to gain information from a certain point. So, in that case fixation count is greater amount. Pictorial information includes discrete point of focus and greater fixation count whereas, textual information includes continuous point of focus and lesser fixation count. So this is easily distinguishable. Again, two eyes have to different perception of viewing a single point. As a result, we have found that using two eyes information increases the information gain as well as accuracy.

Another distinguishable feature is movement ratio. We have defined pupil movement ratio as the following;

$$\text{Movement Ratio} = \frac{Horizontal\ Movement\ duraion\ of\ Pupil}{Vertical\ Movement\ duration\ of\ pupil}$$



In case of textual information it is found that horizontal movement is greater comparing to the vertical movement and in case of pictorial information vertical and horizontal movement are of similar amount. So, ratio for textual information is lot higher than the ratio for pictorial information.

Eye gaze movement and eye gaze movement variation are nothing but the consideration of the pupil movement and its variation. Distinguishable characteristics of pupil movement are noted from different context which helps to determine the intention of a user.

4.3.3 Classification:

Classification is a general process related to categorization, the process in which ideas and objects are recognized, differentiated, and understood. A classification system is an approach to accomplishing classification. Feature selection is an important part of classification. After selecting features classifier has to be trained.

The choice of an algorithm for classification is in many ways the easiest part of developing a scheme for object classification. There are several ``off-the-shelf" approaches available (though there is obviously still room for improvement.) There are two major hurdles to be faced before these methods can be used, though: a training set must be constructed for which the true classifications of the objects are known, and a set of object parameters must be chosen that are powerful discriminators for classification. Once a possible classifier has been identified, it is necessary to measure its accuracy.

Adding many irrelevant parameters makes classification harder for all methods, not just the nearest neighbor methods. Training classifiers is an optimization problem in a many-dimensional space. Increasing the dimensionality of the space by adding more parameters makes the optimization harder (and the difficulty grows exponentially with the number of parameters.) It is always better to give the algorithm only the necessary parameters rather than expecting it to learn to ignore the irrelevant parameters.

In machine learning, support vector machines (SVMs, also support vector network) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a nonlinear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. A schematic example is shown in the illustration below. In this example, the objects belong either to class GREEN or RED. The separating line defines a boundary on the right side of which all objects are GREEN and to the left of which all objects are RED. Any new object (white circle) falling to the right is labeled, i.e., classified, as GREEN (or classified as RED should it fall to the left of the separating line).

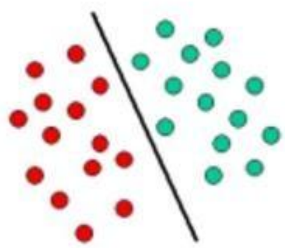


Figure 22: Linear SVM decision boundary

The above is a classic example of a linear classifier, i.e., a classifier that separates a set of objects into their respective groups (GREEN and RED in this case) with a line. Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, i.e., correctly classify new objects (test cases) on the basis of the examples that are available (train cases). This situation is depicted in the illustration below. Compared to the previous schematic, it is clear that a full separation of the GREEN and RED objects would require a curve (which is more complex than a line). Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyper plane classifiers. Support Vector Machines are particularly suited to handle such tasks.

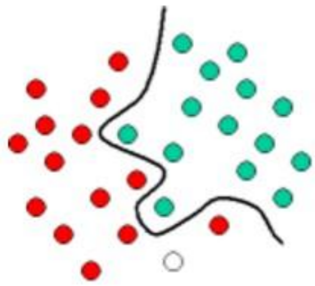


Figure 23: Non-Linear SVM decision boundary

According to the form of the error function, SVM models can be classified into four distinct groups:

4.3.3.1 CLASSIFICATION SVM TYPE 1

For this type of SVM, training involves the minimization of the error function:

$$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i$$

Subject to the constraints:

$$y_i\left(w^T \phi(x_i) + b\right) \geq 1 - \xi_i \text{ and } \xi_i \geq 0,\ i = 1,\ldots,N$$

Where C is the capacity constant, w is the vector of coefficients, b is a constant, and ξ_i represents parameters for handling non separable data (inputs). The index i labels the N training cases. Note that $y \in \pm 1$ represents the class labels and xi represents the independent variables. The kernel ϕ is used to transform data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over fitting.

4.3.3.2 CLASSIFICATION SVM TYPE 2

In contrast to Classification SVM Type 1, the Classification SVM Type 2 model minimizes the error function:

$$\frac{1}{2}w^T w - \nu\rho + \frac{1}{N}\sum_{i=1}^{N}\xi_i$$

Subject to the constraints:

$$y_i\left(w^T\phi(x_i)+b\right) \geq \rho - \xi_i,\ \xi_i \geq 0,\ i=1,\ldots,N \text{ and } \rho \geq 0$$

In a regression SVM, estimates the functional dependence of the dependent variable y on a set of independent variables x y = f(x) + noise

The task is then to find a functional form for f that can correctly predict new cases that the SVM has not been presented with before. This can be achieved by training the SVM model on a sample set, i.e., training set, a process that involves, like classification (see above), the sequential optimization of an error function. Depending on the definition of this error function, two types of SVM models can be recognized:

4.3.3.3 REGRESSION SVM TYPE 1

For this type of SVM the error function is:

$$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i + C\sum_{i=1}^{N}\xi^{\bullet}{}_i$$

which we minimize subject to:

$$w^T\phi(x_i) + b - y_i \leq \varepsilon + \xi_i^{\bullet}$$
$$y_i - w^T\phi(x_i) - b_i \leq \varepsilon + \xi_i$$
$$\xi_i, \xi_i^{\bullet} \geq 0, i = 1,\ldots,N$$

4.3.3.4 REGRESSION SVM TYPE 2

For this SVM model, the error function is given by:

$$\frac{1}{2}w^T w - C\left(\nu\varepsilon + \frac{1}{N}\sum_{i=1}^{N}\left(\xi_i + \xi_i^{\bullet}\right)\right)$$

which we minimize subject to:

$$\left(w^T\phi(x_i)+b\right)-y_i \le \varepsilon+\xi_i$$
$$y_i-\left(w^T\phi(x_i)+b_i\right) \le \varepsilon+\xi^{\bullet}{}_i$$
$$\xi_i, \xi^{\bullet}{}_i \ge 0, i=1,\ldots,N, \varepsilon \ge 0$$

There are number of kernels that can be used in Support Vector Machines models. These include linear, polynomial, radial basis function (RBF) and sigmoid: **4.3.3.5 Kernel Functions**

$$K(\mathbf{X}_i, \mathbf{X}_j)=\begin{cases} \mathbf{X}_i \cdot \mathbf{X}_j & \text{Linear} \\ (\gamma \mathbf{X}_i \cdot \mathbf{X}_j + \mathrm{C})^d & \text{Polynomial} \\ \exp(-\gamma \,|\, \mathbf{X}_i - \mathbf{X}_j \,|^2) & \text{RBF} \\ \tanh(\gamma \mathbf{X}_i \cdot \mathbf{X}_j + \mathrm{C}) & \text{Sigmoid} \end{cases}$$

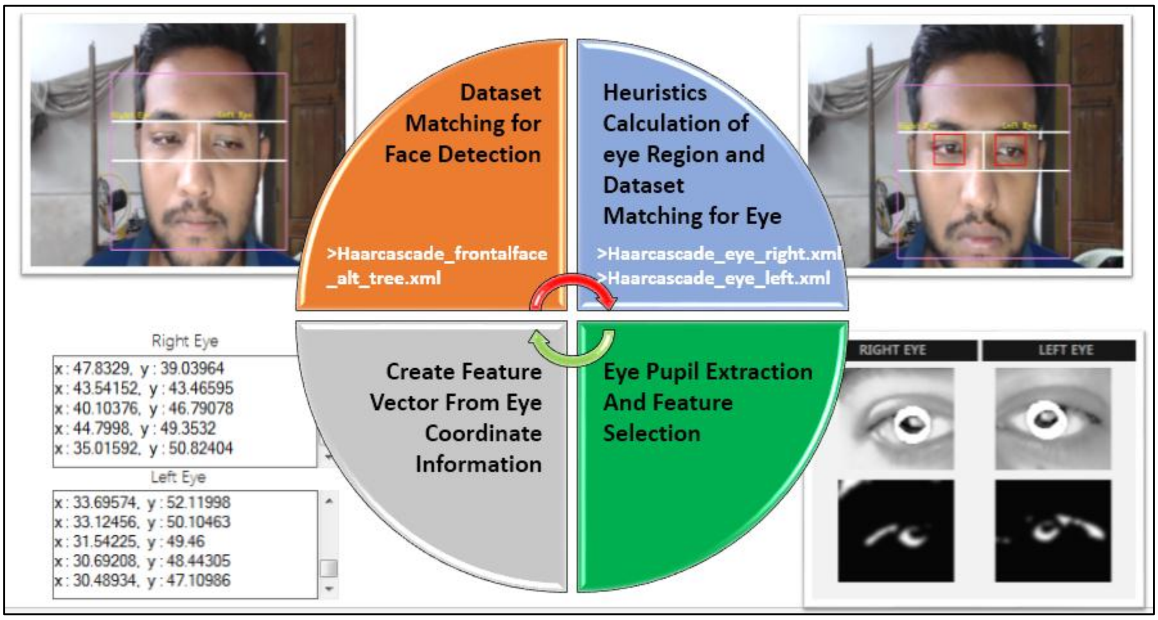
where $K(\mathbf{X}_i, \mathbf{X}_j)=\phi(\mathbf{X}_i)\bullet\phi(\mathbf{X}_j)$

that is, the kernel function, represents a dot product of input data points mapped into the higher dimensional feature space by transformation ϕ.

Process	Steps
Eye Detection:	• Face detection using viola jones algorithm. • The Region of interest that means the possible eye area is determined by a heuristics measurement. • Using Haarcascade matching left and right eye region is detected from the ROI. • Preprocessing such as thresholding, filtering is applied on the detected right and left eye area. • Circular Iris is detected using Hough circle algorithm [1,2] from the left and right eye. Again as we know pupils are the center of iris, so the center of iris is calculated. • Left and right pupil [3,4,5] is detected. • Pupil movement is smoothen using kalman filtering.

Intention and Attention detection	• Pupil movement in the 2d screen is recorded. • It generates x,y coordinate values. • For the feature extraction this coordinate values are used. • Distinguishable features are selected. • Features are calculated from the coordinate values that we have get from the pupil tracking. • Features include information from both eyes (left and right eye). • This features are used to generate the feature vector. • Feature vectors are used to train the Support Vector Machine Classifier. • In SVM we have considered 2 class label. • As intention includes text or image classification. • We have considered 2 class problem to classify text or image information using SVM.

5 Training and Evaluation:



Here in our research we have used an Intel given dataset named as

Haarcascade_frontalface_alt_tree.xml for face detection. Viola jones algorithm is used for this purpose. Then some heuristics calculation is made to detect the possible region of the eye. Again for detecting left and right eye we have used two separate datasets respectively called Haarcascade_eye_right.xml and Haarcascade_eye_left.xml. That gives and output like that. Then on this extracted images we applied some image processing tools to increase the quality of the image. Then on the processed image Hough circles detection algorithm is applied to detect the pupil position. This will provide us the gaze information. The bottom left box outputs the result of the pupil position based on the screen pixels. These extracted coordinate values are used to calculate feature values. Like, maximum fixation count, minimum fixation count, average fixation count, Right eye movement ratio and left eye movement ration.

From here we have generated our own dataset for training the machine. We have set the environment for the experiment. Several users are asked to read multiple text documents. And few are asked to see several pictures. When they are reading or watching the picture their pupil position’s coordinate values are detected. Then these values are used to calculate the feature vector. Which is a training sample for the classifier. We have used ten users to do this task with five different text and five different images. Two of the samples respectively one text and one image that we have used for training are given below.

It�s a matter of delight that department of CSE, IUT is going to organize �IUT 6th National ICT Fest 2014� a great pride and honour of IUT computer society. ICT Fest provide the platform to the novice of technology. Again, it�s a platform for new discovery and new challenge to the students of Public and Private Universities, including schools and colleges of Bangladesh.
On this important occasion, I extend my warm congratulations and felicitations to the participants and organizers of the contest. I anticipate that this event will be enthusiastic as well as competitive to the newcomer. I firmly believe IUTCS will make it a continuous process in the years to come giving opportunities to the talented students to prove their ability in various aspects of ICT and continue their due share for building a Digital Bangladesh.
I have come to know that IUT Computer Society (IUTCS) was formed by the students of Department of Computer Science and Engineering (CSE) in 2008.It is also appreciable that IUTCS is also appreciating students from different sectors to participate in the programming contests and classes, application development classes, co-curricular aid and projects, workshops and seminars and also �ICT Fest�.
The idea of national ICT Fest conceived by IUTCS encourages knowledge exchange, research, development, ICT awareness, identity prospective fields and collaborations among the various parties involved for the overall development of the ICT sector.
Events like National ICT Fest bring together key ICT professionals, researchers and the mass population on a large

Figure 24: Sample Text

Figure 25: Sample Image

After doing the test with several people we have come up with a training dataset like this.

Features Label	Max Fixation Count (Right)	Max Fixation Count (Left)	Min Fixation Count (Right)	Min Fixation Count (Left)	AVG Fixation Count (Right)	AVG Fixation Count (Left)	Move ment Ratio (Right)	Move ment Ratio (Left)
TEXT (Label-1)	37	45	2	1	1.115	1.272	2.385	2.040
	6	15	2	2	0.303	1.667	1.417	1.827
	5	11	1	1	0.759	1.30	2.423	2.222
	15	45	2	3	1.148	2.173	2.740	3.230
IMAGE (Label-2)	106	104	4	6	7.894	8.520	0.466	0.493
	105	109	7	8	6.244	6.388	0.4925	0.647
	119	113	5	6	6.462	6.857	1.066	0.758
	120	125	9	6	7.852	8.998	0.666	0.785

As we have used supervised learning methodology for real time detection we passed this training dataset to the SVM classifier alongside the Testing data. Then SVM classifier was able to classify the classes of the testing dataset. At first we have passed 20 training samples and 10 testing samples. In this situation classifier could classify 5 of the testing data correctly. So it had accuracy rate of 50%. Then we trained the classifier with 30 training samples and with the same 10 testing samples. This time the classifier could classify 6 samples correctly. Then we took 50 training samples and same 10 testing samples. This time it could classify 8 of them correctly which means an accuracy rate of 80%. Finally we took 60 training samples. This time classifier is more accurate. It classified 9 samples correctly so that means a very high accuracy rate of 90%.

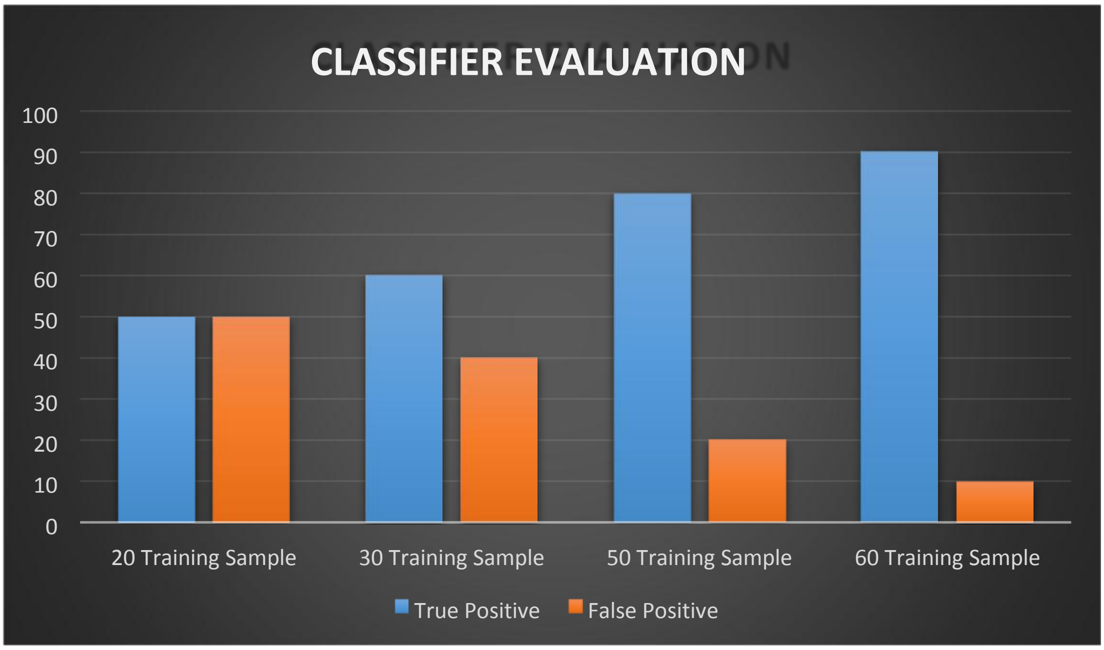


Figure 26:Classifier Accuracy Rate (%)

Form the graph it is shown that as the number of training samples increase the accuracy rate will also increase. Our target is to create a dataset of five thousand samples. This will increase the accuracy of the classifier. So the new testing samples will be classified properly.

6 Application Overview:

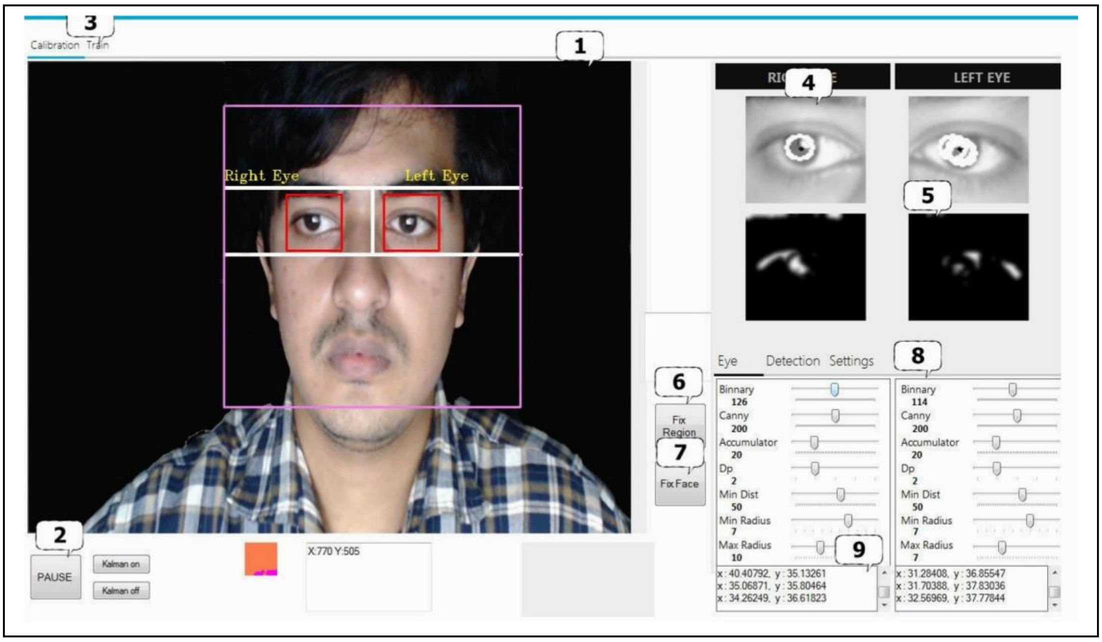
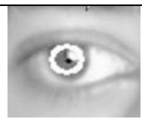
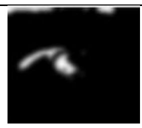
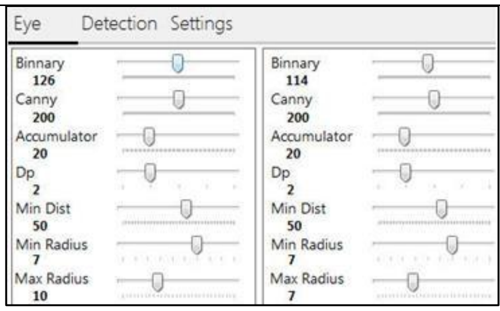
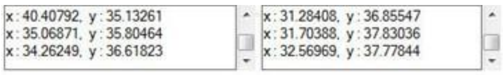


Figure 27: Application Interface

Indicator	Functionalities	Interface
1	• Image box take input image. • Webcam input image is taken.	
2	• Used to play / pause input video stream	
3	• Tab control includes two different tabs. One for calibration and other for training dataset.	
4	• Grey conversion of the original input image.	
5	• Processed image of the grey image.	

6	• Used to fix the detected region of face.	
7	• Used to fix the region of eyes.	
8	• Control panel resides several setting options for calibration and controlling different threshold values. Eye eye has own setting options. • Control panel has three different tabs for eye pupil calibration, detection control and setting. • Each setting has different controls over the environmental condition.	 
9	• It includes x and y coordinate values. Gives 2d coordinate position values of the detected pupil.	

7 Limitations And Advantages:

7.1 Limitations of the current system:

- **Costly:** Existing systems are not cost effective. Most of the devises used for gaze [7,10,11,13] tracking is beyond the reach of the normal people. Like now a days for gaze tracking “tobii gaze tracker” is used. Which is really expensive for the regular people to use.
- **Head mounted device:** Here for tracking eye people need to wear different head mounted devices. Which is a redundant gear for the users to wear. So this is not feasible for the users to wear an extra gear to use the system.
- **Content not classified:** No traditional device can detect which type of objects are there in the area of interest. The human attention [6,14] is on texts or images cannot be specified through the existing systems.
- **Static Way of Finding User Experience Goal:** Now the way finding the user experience goal is static. A prototype is made and provided to the users for testing. The testing result is noted statically. Then analyzed together. Then necessary changes are made. This becomes a time consuming process.

7.2 Advantages of proposed system:

- **Cost Effectiveness:** The main advantage of our system is cost effectiveness. Rather than using a traditional gaze tracker we are only using a webcam to track the human eye. This reduces the cost of detecting eye gaze [7,10,11,13].
- **No Head Mounted Device:** We need no head mounted device. So the user do not need to wear any extra gear to use the system. A simple webcam is placed in front of the user. This thing is more feasible then using those devices
- **Content Classification:** We are classifying the contents of the interface. We are detecting texts and images of an interface. So it becomes easy for the designer to understand the user intention [8,9,10] and attention[6,14]. Users intention [8,9,10] and attention[6,14] is upon which type of object is detected so this will be helpful for the designers to know where to place which type of object.
- **Dynamic way of finding user experience goal:** We are proposing a dynamic way of finding user experience goals. The given prototype will be tested and our system will detect the human intention [8,9,10] and attention[6,14]. And it will analyze these data to give a report on this. By watching the report the designer will be able to see what are the problems in the current design. Then he can take necessary steps to improve the interface design.

8 Operational Environment:

8.1 Lighting Condition:

The accuracy of using an image processing tool largely depends on the lighting conditions. As the amount of light changes with time the processing parameters should be changed accordingly. Our proposed method is dependent on the detection algorithm which is a subject of image processing. For better performance we must consider good lighting conditions. The algorithms we have used require several threshold values. Based on the lighting conditions threshold values should be set. These algorithms finally produce the required features. Based on the quality of image the features values vary. For better classification we need distinguishing feature values which can be achieved by setting the threshold values based on the lighting condition. Without considering the lighting condition there might be noises in the feature extraction process. Which may lead to miss classification.

8.2 Stability of Head Movement:

Our approach is considered with less head movement. As we have used a lost cost solution of using only a webcam to detect gaze information so greater head movement may hampers the total classification in the intention detection. Since we have used a webcam only so the experiment must be conducted in a remote environment. The feature values depend on the pupil movement of the human eye. So if the movement of the head position is too large then it would not be possible for the webcam to keep track of the pupil position. For continuous tracking of the pupil position head position must be at a stable state. Without stable head position the tracking might lead to misleading feature values which in turns leads to misclassification. Therefore, for better performance we will use only our eye movement when head is in a stable movement.

8.3 Calibration:

System calibration is very much important aspects as we have considered. In our system whenever a new user sits in front of the computer and also for different lighting conditions system must be calibrated. There are several values which must be adjusted for different persons as the skin tone of one person might vary from another person. Again the light around the environment might vary as well. For this reasons calibration is the basic requirement of our system. For example there is a binary threshold value which must be set for reducing binary noises. Canny threshold value must be set for detecting the edges in the processed image. Accumulator threshold value is required for applying Hough transform algorithm [2,22]. Hough transform [2,22] is a voting procedure where accumulator threshold value is needed. Minimum distance among the detected circles, Minimum and maximum radius of the circles, all these values must be set according to the user and the environmental condition. That is why calibration process is another important operational requirement of our system.

8.4 Output Result:

In the previous works on human intention researchers are using dedicated gaze trackers to extract gaze information. But in our system we are using only a webcam to detect gaze information. So the quality of the webcam largely affects the performance of gaze tracking process. So get better out result we have to use HD webcam instead of using normal webcam. So that the webcam can provide error free gaze information. As, the quality of input image defines the detection success so a quality webcam is one of the main requirements of our system.

9 Future Scope:

The accuracy of our system is up to now very high because of the differentiable features selection. But still there are some points where we can work more and come up with a more accurate and sophisticated system. There are some works which can be taken as a future goal. Like

- Still now calibration is a manual process. Research is required to make it an automated one. The system will calibrate itself according to the users of different skin tones and also based on the lighting conditions.
- We will look for more features so that the classification accuracy will increase. Currently we are working with eight features. Adding more features might increase the accuracy again it can decrease the accuracy or keep it just the same. So considering these cases is an important issue. So this is a big open space for future research.
- Currently we have used sixty training samples to train our classifier. Which is not yet a handsome amount. We want to increase our training samples to at least five thousand. This will increase the accuracy of the classifier for classifying new testing samples.
- So far we are working on human intention based on a particular interface. Whether the human intention is on textual information or pictorial information. But we want to generalize this idea. At the same time we want to work on several more aspects of human intention.
- Our future plan is to research on finding Human attention. Our main target will be to find a dynamic way of defining human attention based on gaze information.

10 Conclusion:

Nowadays designing an interactive system is a long process. The interaction design must not conflict with the human cognition. So there should be a system that will introduce an automatic finding of user experience goals. Our proposed system will work keeping in mind the human cognition and based on that it will produce an output for the designers. So this will be a big step forward towards an interactive interaction design principle. Later on based on this research it can be concluded that human intention varies from person to person. A user may feel interest on a textual interface whereas other user may try to seek information from pictorial interface. In any design issues such as designing commercial websites it is inevitable to design interface based on user requirement. Users try to seek information in a feasible way where they have the flexibility to gain information in less time. So, the intention also differs with the context. Again, another important issue is age. An aged user may not like an interface with too much pictorial information. Because this type of user try to seek writings in the interface. On the other hand a younger aged user may feel attraction in an interface with pictorial information. So, it is clear that intention is not a fix measurement to be predefined. Human intention study is a must in designing interactive system for better user experience.

Reference:

[0] Michael Padilla and Zihong Fan, “EE368 Digital Image Processing Project - Automatic Face Detection Using Color Based Segmentation and Template/Energy Thresholding”, Stanfor University, Spring 2002-2003

[1] Prateek Verma, Maheedhar Dubey, Somak Basu, Praveen Verma, “Hough Transform Method for Iris Recognition-A Biometric Approach”, June 2012.

[2] Hideki Kashima* Hitoshi Hongo** Kunihito Kato* Kazuhiko Yamamoto, “An Iris Detection Method Using the Hough Transform and Its Evaluation for Facial and Eye Movement”, Faculty of Engineering, Gifu University **HOIP, Softopia Japan/JST, January 2002.

[3] A.Ektesabi, A.Kapoor, “Exact pupil and iris Boundary Detection”, 2011.

[4] Gomai, 1, A. El-Zaart, 2, and H. Mathkour3, “A New Approach for Pupil Detection in Iris Recognition System”, Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia, 2012.

[5] Yoshinobu Ebisawa, “Robust pupil detection by image difference with positional compensation”, Faculty of Engineering, Shizuoka University, May 2009.

[6] Jing Zhang, Li Zhuo, Zhenwei Li, Yingdi Zhao, Signal and Information Processing Laboratory, “An Approach of Region of Interest Detection Based on Visual Attention and Gaze Tracking”, Beijing University of Technology, Beijing, China, 2011.

[7] Ngip-Khean Chuan, Ashok Sivaji, “Combining Eye Gaze and Hand Tracking for Pointer Control in HCI, Developing a more robust and accurate interaction system for pointer positioning and clicking”, Product Quality & Reliability Engineering (PQRE), MIMOS Berhad, Kuala Lumpur, Malaysia, December 2012.

[8] Young-Min Jang, a, Rammohan Mallipeddi, Sangil Lee, Ho-Wan Kwak, Minho Lee, “Human intention recognition based on eyeball movement, pattern and pupil size variation”, Department of Psychology Kyungpook National University, 1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, Republic of Korea, 2014.

[9] , Ernesto Arroyo, 1, Ted Selker, “Attention and Intention Goals Can Mediate Disruption in Human-computer Interaction”, Department of ICT, Universitat Pompeu Fabra, Barcelona, Spain, ernesto.arroyo@upf.edu, Carnegie Mellon Silicon Valley, Moffett California, USA. ted.selker@sv.cmu.edu, 2012.

[10] , Manu Kumar, Andreas Paepcke, Terry Winograd, “EyePoint: Practical Pointing and Selection Using Gaze and Keyboard”, Stanford University, HCI Group , 353 Serra Mall, Stanford, CA 94305-9035 , {sneaker, paepcke, winograd}@cs.stanford.edu, 2008.

[11] , Manu Kumar,Terry Winograd, Andress Paepcke, “Gaze-enhanced Scrolling Techniques”, Stanford University, HCI Group, 2007.

[12] So-Jeong Youn, and Kyung-Whan Oh, “Intention Recognition using a Graph Representation”, World Academy of Science, Engineering and Technology, 2007.

[13] Ibraheem Frieslaar, 2734760, Supervisors: Mr Mehrdad Ghaziasgar and Mr James Connan,, “Moving the Mouse Pointer Using Eye Gazing, Computer Science”, Department of Computer Science, University of the Western Cape, 2014.

[14] *HSUAN-YINGCHEN ANDJIN-JANGLEOU, “Visual Attention Region Detection Using Texture and Object Features”, Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, 621 Taiwan, 2010.

[15] M. Kowalik , “How to build low cost eye tracking glasses for head mounted system”, West Pomeranian University of Technology in Szczecin, 2010.

[16] Soussan Djamasbi “Transactions on Human-Computer Interaction”, June 2014

[17] “An Introduction to the Kalman Filter”, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, July 2006.

[18] P. Viola and M. J. Jones, “Robust real-time face detection”, International Journal of Computer Vision, 57 (2004), pp. 137–154. http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb.

[19] Y. Freund and R. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting”, Journal of Computer and System Sciences, 55 (1997), pp. 119–139.http://dx.doi.org/10.1006/jcss.1997.1504.

[20] Richard O. Duda, Peter E. Hart and David G. Stork, Pattern Classification, 2nd ed:

[21] R. Gonzalez and R. Woods, Digital Image Processing, 3rd ed.: Pearson, Education, 2009.

[22] Dana H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes”, 1981.

[23] Harold Bekkering, Jos J. Adam, Herman Jingma, A. Huson, H.T.A Whiting, “Reaction time latencies of eye and hand movements in singleand dual-task conditions”,in Experimental Brain Research, 7 September 1993.