## 1. Introduction:

Eye tracking has the potential to bridge the gap between humans and computers. In this project we tend to achieve a more natural, intuitive, comfortable, fast and reliable ***HCI (Human Computer Interface)***. For interacting with the computer system, we use input and output devices. They are the only mode of transfer of information to and from computer systems and humans. But we have to think of input and output human operations as an intermediate step in communication of information flow between humans and computers.

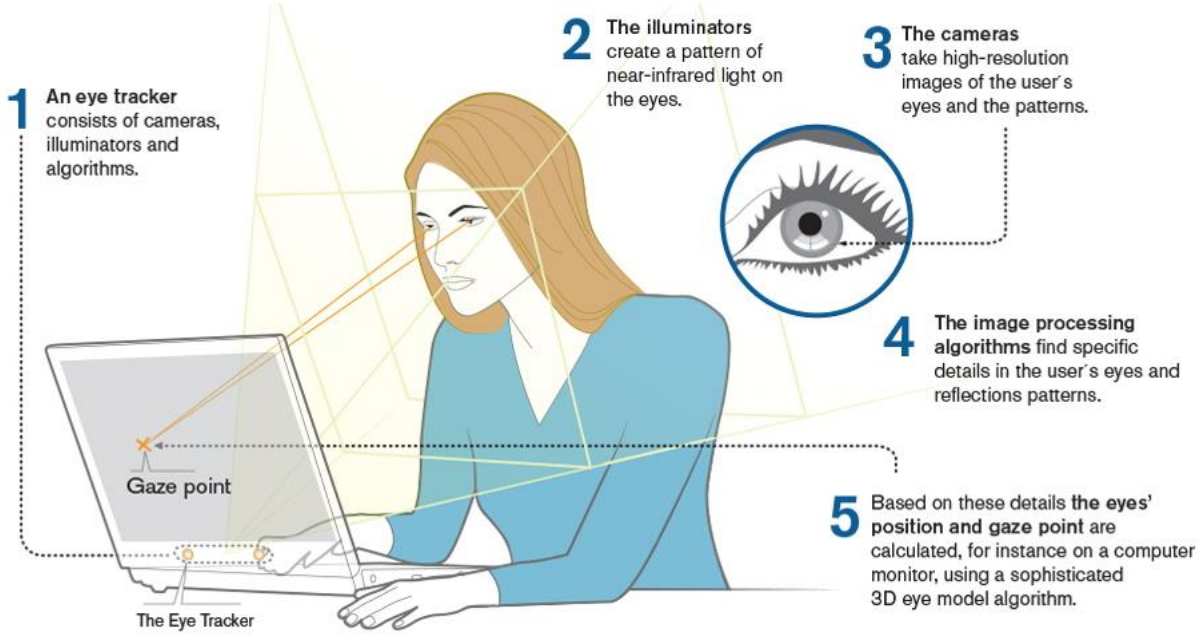



**Fig.1: The most general structure of an eye detection and gaze tracking system [13]**

The aim of this project is just to percept and analyse human intentions and gestures to interact with programs. Traditionally user intentions are to be consciously informed to the computer system through input devices. And the computer provides information to the user through output devices. The output devices are intuitive and very close to how the humans percept the information. It appeals directly to how the user naturally processes information and how he/she naturally interacts. The most popular output devices either present pictorial information or sound output to the users. The pictorial information presentation to the user is through monitor for example. The monitor presents information as written text, pictures, videos, etc on the screen. This appeals to the user gathering the information very naturally as he directly can process on or can assimilate this information. The second category of output to the user is through sounds like speakers and other sound output system. This mode is also natural way for humans to interact as, generally too humans interact with one other through speech. So speech information synthesis is much natural and intuitive for users. The last piece for argument that the current output mode system is much appealing, natural and closer to human thought process is that mode of input is limited by the typing speed of user and the CPU is much faster than our input. But the information output speed is limited by the information processing capabilities of user like the speed of videos is normally set to "1.00x", which can be "1.25x", "1.50x","2.00x" but we usually are not able to cope up with such large video speed and stick to normal "1.00x". Likewise, is for sound, the screen readers or the recordings could be playable at much faster rate but we usually stick to the default normal speed. So, these all observations are sufficient to accept that the interaction of humans with computers has bottleneck at the **USER INPUT** step. The communication interaction of computers with humans are two way; USER INPUT: humans input information to the computer and SYSTEM OUTPUT: system transmits output to user. In this duplex communication, the user input of information to the computer is limited by the current technology and not by the user capabilities to

interact. So, in this project we develop some new input interface to computers that have the capability to replace the traditional input devices like mouse and keyboards. And the user input to the computer is as fast as user reactions and user processing speed. This can be thought of as human conveying information through facial features, eye movements, gestures, mannerisms, behaviour, etc. If the computer system is intelligent enough to recognize these subtle movements and could guess what the user is trying to convey then the user input effort and time both could be minimized drastically [25].

This can be analogously thought of as interacting to a deaf person through sign languages. Had the other person not being specially abled then communication and interaction with it could be very easy, fluent and natural. But now there has to be a conscious effort made to communicate. But now we have to make communication through an intermediate step of using the sign language. Then first we process the information in our head, frame the words and then generate corresponding sign language gesture to communicate.

So, in our endeavour we make computers capable of analysing and recognizing our gestures to process input information. We do not delve into facial emotion recognition or subtle information conveying acts to be detected and accurately the intention of user is guessed. That of course will be the objective of further research. But this project is to set definitive and concrete infrastructure to directly get the inputs from the user without his/her conscious effort. This will also save time delay that the user spends in thinking what he/she wants to input and what corresponding actions it has to perform on input devices. For example, the model can see if the user's gaze is directed to the screen while a video is played or not. The natural action is if the user gaze is not directed to the screen then unnecessarily playing the video is never a use. Traditional methods would need user to consciously press the pause button to pause the video and only then the system recognizes to pause the video.

Eye tracking as an algorithmic approach tends to locate the position of eyeball and further the behaviour of eyes or the gaze direction can be estimated by pupil and iris position [22]. Almost all the major research in this field is in the branch of VOG: VIDEO-OCULOGRAPHY, but the research is not limited to this major area that has the least USER-INTERVENTION but other fields with much increased USER-INTERVENTION and equipment's, majorly three areas, i.e., IROG: INFRARED OCULOGRAPHY, SSC: SCLERAL SEARCH COIL, EOG: ELECTRO-OCULOGRAPHY. But in video – oculography too, there are two major paths, one involves full head gear with a camera mounted at appropriate position to capture inputs from the user and the other one has REMOTE camera fixed to evaluate gaze and other features of eyes. The industry standard products and projects developed have an equipment budget that is impractical to be used by all types of users at all computing devices. And along with the budget there is generally specific need to wear equipment's. The best that we have come till now is the size of equipment has been reduced to smart glasses. And this is the major obstruction for this to be mainstream and eventually is being able to perform as reliable input interface. The EYE Human interface is undebated to be a NATURAL and SMOOTH [17].

## 1.1 WEBCAM AS THE INPUT INTERFACE:

Technically speaking webcam is an input device that captures the analog signal in the form of the light of different colours striking it into digital format. Webcam can capture still images as well as video stream. The video stream is collection of still frames that are displayed in continuity at a rate that is greater than the threshold value to create illusion of continuity. But in our proposal, we use webcam to record the user.

## 1.2 OUR HUMBLE ASSUMPTIONS INCLUDES:

1. The first and simple assumption is that to use the computer system, the user has to be facing the screen.
2. And the second assumption is that the screen has a webcam on it.

Now, these are the easily satisfied conditions on most of computing devices including mobile phones, laptops and desktops with a web-cam.

### 1.3 Why these assumptions are Necessary AND Valid:

Our model takes video stream of webcam to analyse the user. It detects the user and then analyse and recognize the user actions to assume some corresponding input. So, our model can only work till the point we keep getting the user in our webcam stream. So, lack of any user in the webcam stream is assumed to be an idle system without any user. This assumption is very much valid as for the user to use the computer system, the user has to see the screen. And if *screen is in his field of view, he/she too is in the field of view of webcam* mounted on screen. Screen is an important output device in today's computing system and for the use of system or to interact with the system, the screen has to be looked on.

### 1.4 What are the inputs transferred by webcam:

- We analyse the video stream of webcam and first find the face of the user in the stream to see the absence or presence of user.
- Then we locate the eyes knowing the face location.
- *User feature analysis:* Then getting the eyes, we analyse the features of the eyes. The exact status of eyes is recognized and then is evaluated as to what the user is trying to convey. Or what is the need for program to do based on the user behaviour. So, the program is driven by the user reaction and decides its inputs based on it.

This is a great leap from user deciding what inputs it want to give to the computer, now the computer system is driven by the user reactions, movements and gestures. The goal is same to drive the computer system based on user needs and to give the system an input that user wants.

But the change is that the intermediate step has now been eliminated in which user after knowing what it wants to do, consciously says the system an input command accordingly. Now the system tries to analyse and guess what the user wants and what inputs to take based on the eye features and mannerisms of user's eyes.

### 1.5 Eye Features as sufficient input carrying information on user wants:

There have been many research proposals and successful experiments done to validate that EYE FEATURES OFFERS A WINDOW to what the user wants and what should be done. Eye contains in it many information about the user like the:

- ***Gaze intensity:***
  Gaze intensity directly corresponds to the attention level and the interest of the user. The gaze intensity could be directly computed by the amount of eye that is open. This user feature can be a sign that the user wants to **MAGNIFY SOME AREA OF SCREEN**. The sudden user gaze increase means that the user is surprised by the information material and it is logical and natural that MAGNIFY command is transmitted. The user may want to analyse this information on the screen that surprised him/her. Then if the gaze intensity is approximately constant, that means really the action magnify was the want of user. But if the gaze swiftly turns away from the magnified area then the MAGNIFY action should be rollback. The algorithm could be modified to learn from user and the future performance be better than present. That is, in future cases of user gaze instantly increasing, is it really a surprise and if it is a surprise then should magnify command be transmitted. These questions could be better tackled.

- ***Blink:***
  Blink can be a very important conscious signal to the system made by the user. It is intuitive to think blink as a signal intentionally to convey some information. In this conscious effort it is similar in idea to the current input methodologies in which user is trying to convey what he wants to the system via a conscious determined action. We normally also tend to use blink to convey some information to others in day to day life. So, this action or movement is conscious but much easier and more comfortable than today's actions like pressing a button or displace the mouse. The blink could be signalled for **SCREENSHOT** for the sake of illustration here. So, whenever the user wants to capture the screen, he/she just have to perform a action of blink. And the most important fact is, it is easy and practical to distinguish between an involuntary blink and a voluntary blink. The involuntary blink has been much researched on and it is established that it follows a characteristic cycle (categorized by different waves graphically) and could be well tracked. In fact, the particular blink cycle at an instant could reveal information about the anxiety level, mood and state of mind.
- ***Gaze direction:***
  Gaze direction is a feature of eye that involves computation of the relative position of the screen and rays signifying the direction user is looking. The most important information given by this feature is to get a signal that the user is currently not accessing any information on the screen and further information should be duly adapted. The practical implementation could take user gaze direction to keep the MONITER CONTENTS STILL. Like while a video is being played, as soon as user's gaze is shifted away from the screen, the video should be **PAUSED**. As it is definite that the user is not accessing the output information available on screen, so the information on the screen should remain there and the next information should wait till user's gaze is not again shifted to the screen. Thus, this input signal of PAUSE is recognized by the computer without the USER CONCIOUSLY AND DIRECTLY GIVING THE SUITABLE INPUT COMMAND BUT THE ACTIONS OF THE USER ARE ANALYSED AND SENSED FOR INPUT.
- ***Drowsiness:***
  Drowsiness is not a traditional input signal. But the state of drowsiness is one of the very peculiar conditions which suggest to us that we do not merely have to achieve the current input commands through our new input methodology but to extend the command set too. And also, drowsiness is a very popular eye feature extraction task applied in **driver fatigue and drowsiness tracking** system**.** This particular application of eye tracking could debatably be the single most influential and important task at present time. Presently a lot of research and work is going on in this field and to point out this is just the "tip of the iceberg." Drowsiness means the condition where the attention level is at the minimum and the user is not in correct state to continue his work but is prolonging to work. The condition could be easily and accurately be sensed by detecting the state of eyes. In drowsiness state, the eye lids are much closer to each other, the eye movements are slower than normal and the attention span is very small. The input interface model recognizes the eye features and extract the state of the eyes to detect that the user is sleepy and take it as an input. Now there could be varied actions that the system could be configured to take on such circumstances like either ALARM THE USER or BACKUP THE WORK TILL THAT POINT and then only move forward.

To correctly decode each feature of the user eye and guess correctly the state of user's mind is an interdisciplinary task needing the researches and works in human psychology and behaviour along with the knowledge of computer science [19] [20].

**1.6 This project detects simple eye features:**

But here we are to set an infrastructure to this type of input methodology. So we do not delve into human psychology and the science of human behaviours. We analyse most general and simple gestures and features that can just convey one definite want of user. Or would pin point the input to be made recognizing the gesture. So here in our model we totally ignore the aspect of deep feature analysis and then sense emotional and behavioural state of the user to guess input for the running program. But we focus on detecting basic actions and then guess input for these actions which are very simple to corelate.

**1.7 Actions and Eye Features detected by this project and corresponding actions taken:**

- **Open and closed eye: *Play and Pause the Video***

  The simplest detection based on the state of eyes is that it is open or closed. The detection mechanism once detects that the eye state to be closed then, the model has to transmit a signal corresponding the state. The action to be taken on eye state to be closed is set simple in the model and it is to pause the video if a video is playing. So, there is no behavioural and psychological analysis of the detected state required and the corresponding action is choice from a binary set. The binary set has the two values as play and pause. Only one of these is to be chosen and the mapping is straightforward. So, the overhead of analysing the user state of mind and the behavioural sense is not implemented. But detection of closed eyes means pausing of video. So, the action limits to detecting the state of the eyes. Because as soon as we detect the state of eyes to be closed, it directly indicates pause the playing video.
  But the trick here is once the eye is detected to be open this does not means no work to do. Because in the situation where the eye gets open after being closed, we have to play the video back.

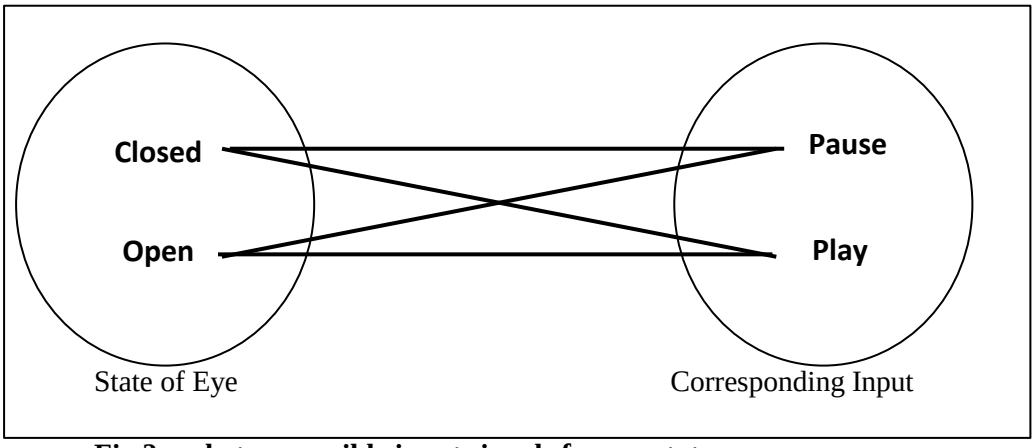



**Fig.2: only two possible input signals for eye state**

So, the simple implementation used for this functionality is to have a variable as a placeholder for the previous state of eye. And if the new state of the eye is same as previous, we do nothing but if the new state of eye if different than the previous than we PUT AN INPUT SIGNAL EQUIVALENT OF PRESSING SPACE BUTTON ON KEYBOARD.
So, the functionality we get is:

**Table1: Input signal mapping to the eye states**

| Previous Eye State | New Eye State | Input Signal |
|---|---|---|
| **Closed** | Closed | - |
| **Closed** | Open | Space button |
| **Open** | Closed | Space button |
| **Open** | Open | - |

*Important:* It is assumed that the input signal to play or pause a video is the space button. If this is not true like the touch screens, please substitute accordingly and the working remains valid.

- **Eye gaze out of screen or not: *Play and Pause the Video***

The user gaze direction estimate is the analysis and recognition to be performed for the state of the eye gaze, based on which we generate corresponding input signal. The detection of the *eye gaze to be in screen or out of screen* needs a sense of the relative eye positions to the screen and the gaze direction. The gaze direction could be obtained by the shape of iris (the central black part of eye). The approximately circular iris indicates the user is looking at the screen nearly at the webcam location. The deformity to the iris could indicate where the gaze of user is directed relative to the screen. The four main directions are top, bottom, left and right and the characteristic pupil ellipse shape easily corresponds them. The characteristics of pupil revealing the user gaze is described in Fig.2. The main characteristics observed are:

**Table2: Eye pupil characteristics according to gaze direction**

| Gaze direction | Pupil concentration | Pupil Ellipse deformation | Cornea Major availability area |
|---|---|---|---|
| **Top** | Centre | Horizontal elongation | Bottom |
| **Bottom** | Centre | Horizontal elongation | Top |
| **Left** | Left | Vertical elongation | Right |
| **Right** | Right | Vertical elongation | Left |

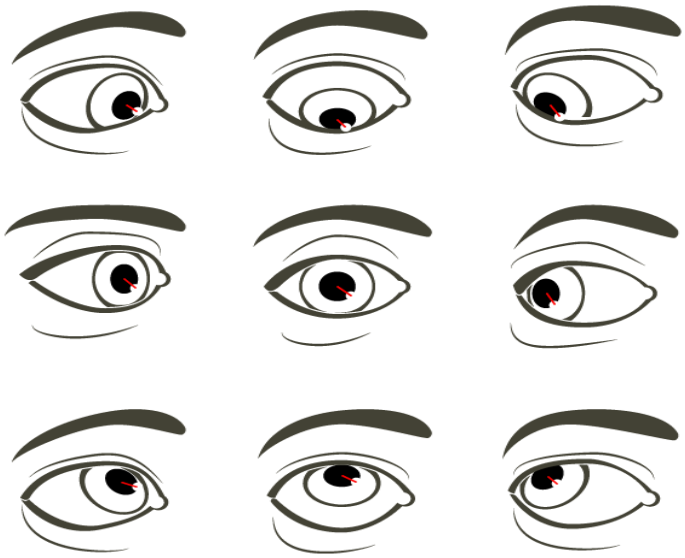


**Fig.3: Pupil Deformations according to gaze direction**

So **Fig3** and **Table2** summarizes many observations on pupil and cornea characteristics that could give the gaze direction. So, a threshold deformity could be set to state that the gaze is still onto the screen. But beyond that threshold, the deformity indicates that the gaze direction is *so extreme that it is out of screen.*

The other possible approach could be to analyse the gaze directions with virtual rays tracing the gaze path and then analysing whether the virtual gaze path strikes the screen or not. But we choose threshold deformity method as the general feature extraction method we use is a **PROJECTION METHOD** which works on calculating the deformity in horizontal and vertical direction and extract the exact eye state and features. So additionally, doing geometric computations for rays tracing eye gaze and locating it relative to the screen would be much extra overhead. And the threshold deformity approach is practically accurate and fast. The input signal generation is very much similar here to the above **eye open and eye closed** case. In this case too similar to Table1, if the gaze is out of the screen and the previous gaze direction was on the screen, we PAUSE the video. And if the user gaze is back on the screen, we play the video. Similarly, all decisions about what input to be transmitted in the system could be decide on the basis of previous gaze *in or out* and new gaze *in or out*.

*"Important":* The horizontal and vertical directions talked here id the line joining the centre of both eyes is assumed to be horizontal and the perpendicular to it is vertical.

- **Eye gaze point on the screen:**

The point on the screen on which user's gaze is focussed could be used for generating an input signal to the system which may enhance the usability of the computer system for the user and could increase the levels of his comfort and reduce his effort. The point on the screen currently having user's attention is a complete discipline in itself to increase the usability of websites. But here in our HCI (Human Computer Interaction) model we following our trend of simple eye feature detection and having a corresponding input signal to be generated, we use this eye feature for **SCROLLING** proposes.

The system works in the manner that if the gaze location of user is bottom of the screen then we inject the input signal to the computer system for SCROLL DOWN. Similarly, if the gaze location is the top of the screen, it SCROLLS UP. But the trick is that both the implementations are not similar.

User gaze could arrive to the bottom of the screen if contents are available after scrolling down in one way that is relatively much slower, in the sense that it would reach to the bottom inspecting the contents from the top and now wants to go to the information further down. Other possibility reaching bottom of the screen is in a quick arrival to bottom skipping many contents. The first motion of reading the contents from top and reaching bottom is much easier to detect as it is in ORDERLY and INCREMENTAL order, so the previous frames could easily predict it. So, in such a motion the SCROLL DOWN input transmitted to the system corresponding to user's gaze at bottom of the screen is easy to correspond from the eye state and movement analysis.

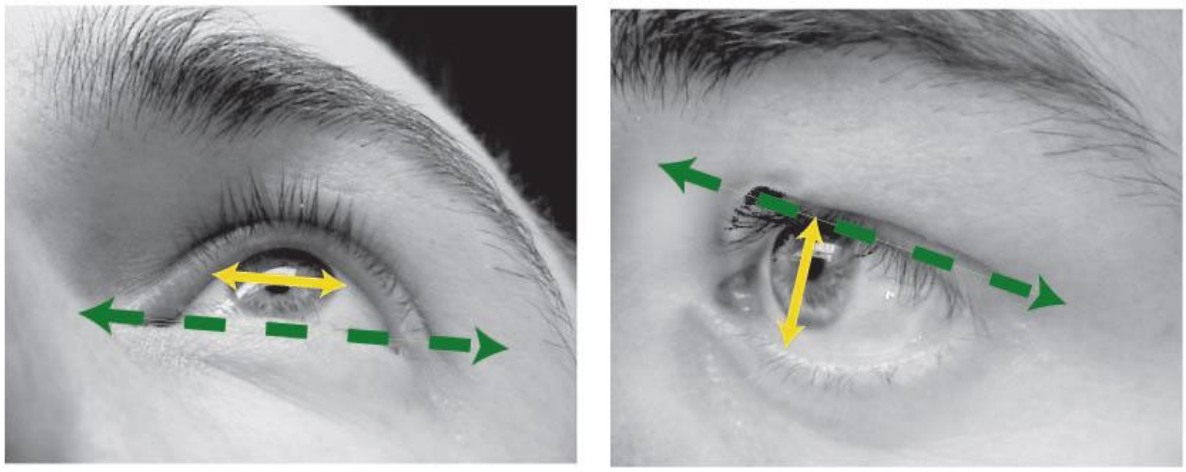
But it is the second type of motion that is difficult to have correspondence to SCROLL DOWN input signal generation. The reason being this to be a quick motion and is very much similar to noise data. Noise data means the quick irregular motion that the human user tends to make without some definite intention. These motions need not convey any information to the computer system but are just unintentional and unimportant. So quickly reaching the bottom without having attention to contents in between just skipping them is very much a candidate for potential noise movement. But we can recognize it successfully and generate the SCROLL DOWN input corresponding to it by detecting that the user gaze is fixed to the bottom after the random motion suggesting that SCROLL ACTION is required. So, detecting the gaze of user to be stationary at the bottom of screen corresponds to SCROLL INPUT

transmitted to the computer system. So, once again we restrict ourselves to detecting simple and much more explicit gestures that have a very definite corresponding input signal too it. Similarly, SCROLL UP is also initiated when the user gaze becomes stationary to the top of the screen after a random motion. The thing to note in the case of SCROLL UP input generation is that user gaze reaching the top of the screen is always a random noise like motion skipping major parts of the screen. This is because while going UP, we have already encountered the contents and the motion tends to be fast and irregular. But again, the fact that the eye gaze remains stationary there before again moving makes it correspond to SCROL UP much directly and easily.

"The prime reason that this approach of relating eye being stationary for some time at the top or at the bottom to SCROLL input is that the retention and the processing capability of brain is in such a way that whenever it sees end after a quick skip over information makes it to stop at the end point and mark it as a checkpoint before proceeding further."

So, here the limit of the screen is the top or the bottom of the screen and AS REACHING HERE IS FOLLOWED BY A QUICK MOTION, the brain works in the way that seeing the end it tends to be stationary for an instant before making another movement. This could be easily visualized as the "CLASSICAL STONE THROWING IN VERTICALL DIRECTION" problem. As the stone is at the end of its journey UP, it tends to be stationary at the topmost point and only then it makes next movement, which is to come down. So, analogously is the working of the brain and as it has skipped over some information and gone quickly, it tends to stop when the end is encountered.
So, apart from these two difficult motions, we do not try to detect any motion that is not simple. Like checking whether the eye is open or closed or gaze is to the screen or away, are some very simple eye features to be extracted. And we treat all other motions to be noise and tend not to react or respond to that. This assumption is not true but basic functionalities are achieved and no need for finding sense in the user movements or decoding subtle human features is to be done. But be beware that actually eye has a potential to convey much more information, only if we can make the computer to understand them.

"*Important":* As detecting eyes at the bottom or top of the screen is the limiting point, and after which the gaze is out of screen. So, the threshold value of the pupil deformity is actually the state of pupil at the screen end points.



**(a)** **(b)**

**Fig.4 : Illustration of the eye gaze to be at the extreme screen points**

(a) the state of eye at the top position of a screen. (b) the state of the eye at the rightmost position of the screen.

- **Blink: *SCREENSHOT***

Blink or more correctly, the voluntary blink is an easy eye state to detect [30]. Blink in true sense is not a state but a combination of eye states and eye movements. So the input signal corresponding to the detection of blink is SCRENSHOT. As most modern computers have a SCREENSHOT input signal, so it could be easily implemented as other trivial input signals like "SPACE BUTTON". The only peculiarity that arises here is to distinguish from a sequence "eye close" and an "eye open" state. The solution adopted is to have three categories for this "eye close" and subsequent "eye open" movement. This is because the same sequence happens but with varied time lag in three different types of movements. The first being, INVOLUNTARY BLINK, secondly, VOLUNTARY BLINK and thirdly, EYE CLOSE AND OPEN. The basic constituent action for all of these is just same but the difference arises from the time gap between occurrence of the constituent actions.

In involuntary blink the "eye close" and "eye open" movement is the fastest and also involuntary blinks form a coherent repetitive pattern, such that it could be easily segregated.

In voluntary blink the constituent actions of "eye close" and "eye open" is relatively slower but has a threshold maximum time to complete the blink.

And lastly, the "eye close" and "eye open" is that sequence when the maximum allocated time window is expired to perform a blink.

The other difference being the actions of eye lids are much more suppressed and subtler in involuntary, whereas it being most deliberate in "eye open" and "eye close "sequence.

How a webcam detects a user blink is shown in Fig.6, where the eye state of "eye closed" is detected in the Fig.6(a) and the eye state "eye closed" is detected in Fig.6(b). So, only after the whole sequence occurs, we testify the occurrence of blink.

In Fig.5, the characteristic cycle of involuntary blinks is shown. After the continued recording of the data, the tracking of the involuntary blinks could be said to follow a characteristic pattern and so involuntary blinks could be identified by this signature wavelet.

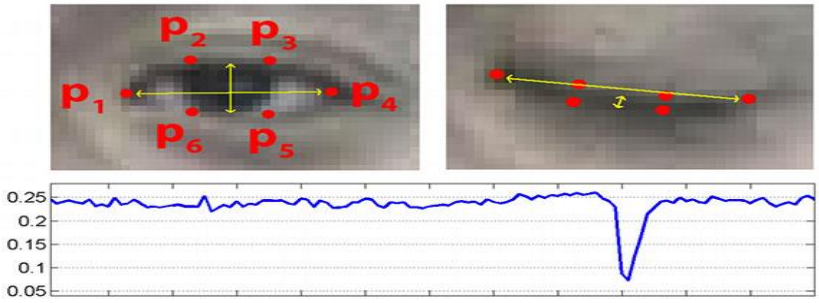
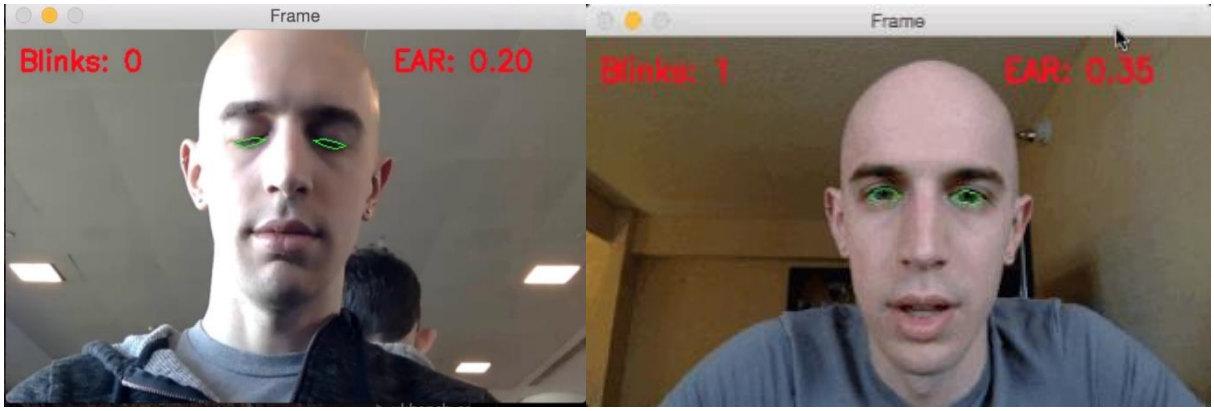



**Fig.5: The characteristic wave cycle for involuntary blinking**




**(a)** **(b)**

**Fig.6: Detection of a blink by the webcam. (a) "eye close" and (b) "eye open"**

- **NO EYE IN THE FRAME:*NO ACTION***

  The state of no eye detected in the webcam stream corresponds to having to do nothing for input signal except if video is being played, it has to be paused. So, this eye state or more correctly lack of any is a detection state. The prime significance of this state is our algorithm knows to stop looking for eye if there is none. This means there should be a possibility of having no faces in the webcam stream and we duly acknowledge it. Not only it saves CPU resources in futile computations but also lack of any eye or face is a state. Generally, lack of any eye is concluded by lack of any face, and it could be assumed for all cases without any major logical fallacy. Our model works in such a manner to detect the face first and then locate the eyes based on the position of face. So, if a face is successfully detected, eye detection is guaranteed in our model. The special cases like webcam facing back of the face or only that part of face is in the field of view of webcam that do no contain the eyes, etc are not possible in our model. As the situation when back of the face is facing the webcam would never be recognized as face in the first place because we train the algorithm to detect faces based on examples of faces in frontal view and so the model recognizes features of face from frontal view. So, back of face is actually not a face. Similarly, if eyes are not visible in webcam, then we cannot claim the remaining part of face that is visible to be a face because eyes are an important feature of the face and without this significant feature the webcam stream would be just marked as without a face, even though it contains part of face. So, a video stream without eyes would be discarded to be one having a face. The working of our model is in such a way that it marks areas of the video frame containing eye like feature as positive candidate and those without eye features as negative and discard them.

So, we have seen that the eye features or the state of eye that we are concerned here to detect are very simple and their corresponding input signals to be generated be straightforward match. So, it actually does huge injustice by undermining or ignoring all the complex a subtle eye features and gestures that are capable of conveying a large amount of information. But doing that would incorrectly take us to the realm of human psychology and behaviour analysis. But as being a proposal for new input system the focus should be on the technical possibility, accuracy, speed and benefit of the model. It at once does not claim to completely replace the existing input technology, no doubt that is the ultimate aim. But for now, it has to be started as being a complement to already existing input infrastructure. Progressively as it matures, many possibilities lie in the future. Integrating it with existing input methods is very easy and is a major strength of this model because webcam devices are too common in computing systems now-a-days. And after webcam the only requirement is to run the software module on the system and let it transmit input signals to the computer system. The input transmitted by other input devices like keyboard and mouse is just similar to the inputs generated by our model. The difference just being the cause of generation of the input signal. In keyboards, it is typically through a key press. In mouse, it is through the displacement of the mouse. And in our framework the origin of the input signals is some eye features of the user which is detected, like eyes open or close, gaze towards screen or not.

## 1.8 Need for new type of Input Interface:

The webcam-based input interface is the filler that could bridge the gap between ***user wants and intentions*** and ***conveying them to the computer system***. It is an established and most fundamental fact in computer science that *the "CPU execution speed is very fast but the rate at which user inputs data is comparatively very slow and that is the limiting factor in execution speed of programs or running of applications."* While this is always assumed to be the go-to assumption and most fundamental hypothesis, there is a fault in this statement. The trick is that our intentions are very slow to reach the programs than the execution speed of the program. If our intensions and gestures are detected as quickly and as smoothly as they originate in our mental system, then the input action would not be treated to be so slow. But unfortunately, we think, perceive and give intents at a much faster rate but WE HAVE TO CONCIOUSLY PERFORM AN ACTION ON INPUT DEVICE TO INFORM THE COMPUTER SYSTEM of our intent and wants.

So, this intermediate stage of performing an adequate action on input device corresponding to what we want to inform the system is what is termed as the slow process. This process is the slowest process in whole sequence of events and is the main culprit. So, nor the input devices are slow neither our thought process to think of what input t ogive is slow. But the fact that after thinking what we want to input to the system, we have to transfer this knowledge to computer system by performing an action on input device, makes input process to slow down considerably.

As established in the initial parts of this report that the output speed like on monitor and such devices are limited by our speed of perception. So, the limiting factor in output mode is human processing capability, and as such there is not much opportunity here to speed up this process. But the thing that is making the input of information to the system slow is "User's deliberate action for conveying input message".

We in this report, try to directly detect the input message that the user wants to convey to the system through the state of his eyes.

*"Eyes are door to the mind".*

Eyes have much information to convey and if analysed properly, a great knowledge about the intention, needs and wants of the User could be guessed. So, to directly interpret the user intentions and state of the mind we analyse eyes for information [35].

The traditional input models like that through a mouse or a keyboard were used in the 1980s. The computer systems when mouse and keyboards became popular input device as changed so much. The components like FLOPPY DISKS and TAPE DRIVES were the main parts of computer systems have come to a state when most millennials have never seen a FLOPPY or TAPE DISKS in whole their life. But the state of inputting the data is keyboard and mouse only. We have face recognition systems in our phones and VIRTUAL ASSISTANTS for the comfort and ease of the user but the dilemma is that "EVEN THE TOUCH SCREEN HAS TO HAVE AN ONSCREEN KEYBOARD TO FEED INPUT" [37]. So, these all observations are suggesting to one peculiar property that, "IT IS DEEP ROOTED IN OUR MINDS THAT INPUTTING DATA IS A SLOW PROCESS, BECAUSE USERS FEED INPUTS SLOWLY". But the actual issue is "OUR EXISTING DEVICES ARE SLOW TO GET INFORMATION FROM USER", if the way in which user inputs the data could change, then maybe the transfer of knowledge or information from human users to computing devices be fast. Pressing keys of keyboard, either it is physical hardware keyboard or it be ON-SCREEN Touch keyboards, has an upper limit of speed because of the action of pressing particular key for every corresponding input. If this action of having to press a key is substituted by any other action to convey input that is faster than "HAVING TO GO TO A PARTICULAR KEY FROM THE ARRAY OF KEYS ON THE KEYBOARD AND APPLY MECHANICAL FORCE", then the speed of inputting data will increase. And as a matter of fact, this report is not undermining the usability or contribution of keyboard in computing world. But while giving it its due respect, we have to acknowledge that WE HAVE TO EVOLVE [34].

## 1.9 Why its right time for new input modes:

Now, after establishing as a matter of fact that there is a need and scope of a new input methodology. It is essential to prove that technology is in such a state that it can provide alternatives and support new models.

The input interface that is proposed in this report is one, in which the webcam takes the videography of the user. This video is analysed in real-time to DETECT THE STATE OF USER'S EYES and then the state and features of the eyes are used to PREDICT or RELATE TO WHAT USER WANTS TO INPUT. This correspondence between the detection of user conditions and input action is the new "PRESSING PARTICULAR KEY ON THE KEYBOARD". The exact input is fed to the computer after the user's actions make this inputting task obvious. Thus, as soon as the user's mind decides for an input or wants to INFROM THE COMPUTER SYSTEM, the system which is constantly analysing the actions and gestures of the user TRIES TO GENERATE THE INPUT according the STATE OF MIND of the user. So, the USER when realizes a ***need*** to INPUT a data, he/she has now "NOT TO PRESS THE CORRESPONDING KEY COMBINATION TO FEED IT INTO TH ECOMPUTER SYSTEM" but ***now*** "As soon as the facial features or the eyes of the user reflect this need, the system generates the corresponding INPUT".

So, the ORIGIN TASK as well as the GOAL TASK are both same in traditional input systems and the proposed input interface systems. And surely the INITIAL/ORIGIN TASK and the GOAL/FINAL TASK have to be same for any input interface that could be designed. This means the **ORIGIN/INITIAL TASK** that the user realizes a need to input data to the system and the **GOAL/FINAL TASK** of that particular input to be received by the computer system. But the path taken or the process in between that effects this transfer of information limits the speed of input. We are here proposing a system that achieves this transfer of information as soon as it originates. The traditional input methods have an intermediate step" of pressing some key combination corresponding to the information that the user wants to convey", for instance. This step is the limiting step or the bottle-neck that takes most of the time. THE USER HAS TO THINK WHAT HE WANTS TO INPUT, this action could not be speedup. But the other thing that the user has to do is to THINK THE PARTICULAR ACTIONS TO BE PERFORMED ON THE KEYBOARD for inputting that data.

**Here lies the misconception**. The USER is really SLOW in performing this task of THINKING what pattern of keys that he has to press for the input that he has in his/her mind and then actually have to make many physical actions for pressing those keys. Then the system is fed with the input data that initially the user wanted to convey. THIS ACTION IS REALLY SLOW AND IT IS BOUND TO TAKE TIME. Not only user has to map his input words to some locations on the keyboards but has to make actions like moving and pressing that WILL TAKE TIME. There is no way time for this is much minimized. The most **POPULAR** SOLUTION till date of this ISSUE IS "PRACTISING TYPING" and increasing TYPING SPEED. Thus yes, time taken for this intermediate sequence of action gets reduced but the gain in minuscule. This solution is anyway not a solution, not until humans have innovation. This slow and fully user dependent intermediate step is what gives us ILLUSION that the speed of input is limited to the rate at which user can feed the inputs. But the speed of inputting data to computer system is limited by the methods or the mode used to extract data from the user. If this mode of pressing correct keys is adopted, this does not mean that in other modes too the capability of the user to make some conscious actions based on what he/she wants to input is necessary. Yes, till the point some conscious effort from the user for some input feeding is required, it would be limited by the physical capability of humans to perform that action. As it is a physical movement by the human, it will be slow. BUT THIS IS NOT THE LIMITATION OF HUMAN USERS TO FEED INPUT. To simply put this argument, inputting data to computer system is similar to giving some information to someone. Now the rate at which we feed input information to computers is just incomparable to our rate of giving inputs to fellow humans.

Now we compare and try to analyse what is the most natural way for humans to provide inputs.

- **Two- way involvement and effort**

  This is primarily because of a two-way involvement in our action of transfer of information to other humans. When we are giving some inputs to fellow humans, he/she is constantly though sub-consciously analysing our features like EYE-CONTACT, FACIAL FEATURES, HAND GESTURES, BODY LANGUAGE, EXPRESSIONS, EMOTIONS, TONE OF TALKING, STATE OF MIND, etc. For a simple instance, the action of expanding the size of eye pupils is a straight sign of surprise. And this gesture is not without any logic. The person when being surprised, wants to fully analyse the situation as the situations is out of normalcy and needs more attention and observation. So, the state of being surprised is followed by the eye pupils to be expanded.
- **Less effort and more subtle way to transfer information**

  This is second most important thing, that the transfer of knowledge or the input of information to other user with done much less effort than we make in inputting data to computer systems. We generally input data to other humans as voice signals or simply put oral interaction with some gestures and movements. Now if we increase the effort in conveying information, the speed will definitely reduce. For example,

  - <u>if the other person is unable to hear us and knows to read</u>: We are left with the option of conveying him information in writing. This takes more effort than just talking and the interaction is slower.
  - <u>If the other person is unable to hear and read but knows sign language:</u>

Then the mode of communication would be the sign language. Any input information that you want to convey the other person is you conversing only in sign language. Thus sign language is not the natural mode of communication for us, we first remember the corresponding sign language gesture and then with action of our hand we show particular gesture signal. Something similar is the case with ***keyboards and mouse***, we have to first think the corresponding intermediate code for the data that is in our mind and that we want to input. And after this when we remember the particular intermediate code for input data, like sign language symbol or particular key combinations, we make a conscious effort to do some actions. The action in sign language example is the hand gestures made for each alphabet, and in keyboard input system, is finding the keys and pressing them. So, communication with sign language is much slower and involves much effort.

- <u>If the other person is unable to hear, read and also do not know sign language:</u>
  The effort in conveying the information that we want to give is simply increased manifolds than the initial case. We have to make general gestures appealing to common sense and wisdom of another person to understand what we want to say. In this process, we first try to think of appropriate gestures that have the same meaning to what the input information is. Then after this mapping step we have to perform these gestures in correct manner such that the input information is conveyed. Surely, this all increase the effort and the time for the communication.
- <u>If the other person is unable to hear and see:</u>
  In this case tremendous effort is required to convey input information to the other human. The time required is also manifolds. This is in no way easy for the input providing party. The other human can in no way involve efficiently and have to patiently keep trying.

So, as the involvement of the second party in the procedure of providing input from one other decreases, effort of one has to increase definitely. As earlier in interdependent manner, one human not only gives input information but the other is extracting the input from the one giving the input. So, giving inputs was easy and fast. Same is with the computer systems.

So, we discussed that how effort in inputting data and so naturally the time taken to input the data is particularly larger than the time user takes to think of next input data. Our aim should be to achieve that rate. In this report we try to feed user input instantly as the user becomes aware of it or wants it.

So, the main fundamental idea behind this proposed model is to develop a system which is capable of extracting data from the user thus establishing a two-way interaction. And the continuous detection of the state of user's features are used to generate input data, thus that middle layer that is essential in the traditional input devices is not in the proposed model. Though, we do not claim full absence of any conscious effort by the user in passing input information to the system but we are endorsing as minimal conscious effort and actions to be performed by the user as necessary for the system to map the user state to accurate input that is conveyed by that state of user. To make it clear, some conscious

efforts and action like blink and wink or hand gesture signals are used. But these user actions are requiring much less effort and are much faster to perform. The hand gestures for bringing next image on the screen could be a simple swipe action. But if we are using the mouse, we have to take the cursor to the next button or forward sign and click it. The conscious efforts are both fast and easy.

The arguments that the proposed system would speed up the input speed is one major goal in itself. But in no sense, it is the only argument in favour of it.

## 1.10 Benefits of the proposed input interface system:

- **Hands-free:**
  The proposed system is based on the fact that the input action user performs that includes his/her hands are slow. These actions are "pressing keys" or "moving the mouse", etc. So, in making the system for input interface faster, we are compelled to have a system that is hands-free.
- **Natural:**
  As the model is inspired by the natural way the humans communicate information or pass on data. So, the model has a central feature to it, that it mimics the natural transfer of information operation.
- **Comfortable:**
  The comfort and user ease is prime goal of the system because "THE LESSER THE EFFORT USER HAS TO MAKE, THE FASTER IS THE INPUT PROCESS". So, to increase the speed of inputting data we have to extract information from the user in subtle and very observant way, such that not much of the effort the user has to make.
- **Intuitive:**
  As the computer is driven by observing the user, his actions, gestures and present state, the working of the system would be very intuitive. And as it tries to decode the state of mind of the user by observing him and analysing him, it will perform exactly on the basis of what the user is thinking and what input data it wants to feed to the computer system.
- **Easy and wide use:**
  Apart from speed benefits, there are many major benefits those are in themselves worth working for. The ease of system use would be much more than the current methods.
    - ***Helpful for People with some disability:***
      This could be stated by the fact that most of the HCI (Human Computer Interaction) and eye or face based systems have been developed for people with disabilities. A person who do not have the gift of two hands could also operate and command the computer as any normal human user would do, not only makes them confident but only minimizes the disadvantages of disability. Similarly, a whole lot of people who are until now, facing some degree of discomfort using a computer system would be on par with others. Much researches and products are developed for specially abled users using eye and face observation. But the unfortunate thing, it is not developed for general computers, the technologies for disabled people using these natural input interfaces are implemented for special computers and are not mainstream. While it is valid that apart from this help, if there is some help from hardware then the usability increases for them. But it comes at a cost. This further help to them isolate them from using normal computers as they are so used to their different computers. This type of natural HCI if is adopted for general computing system. Then not only it would be a help and reduce in effort for general users. But it would present opportunity to specially abled to lead a normal life [29].

- ***The next BILLION:***
  There is a far-reaching, important and influential project ongoing in computing world, its named **"The Next Billion"**. As this project influences many developments and activities in computing world, there are ways being found out to help and contribute in this endeavour. The project simply aims to connect next 1 Billion people to computers and internet, who currently do not have. Apart from providing architecture and services in places where there was none, we are improving the resource availability so that new people could get these facilities. But in this movement, this report could contribute very much. When we would evolve the computer systems in a way that inputting data to the system is easy and natural, then more and more new people could easily learn to operate computer systems. For example, for "pausing" a video that is being played on the screen because he has to go for some other task, a new user has to struggle to remember the correct button or action to perform to do this work. We usually see new users to remember the location of keys on the keyboard. They scan many buttons before finding the one they were looking for. Now if the system which his continuously analysing user's eyes could easily detect when user is not looking or the user is not present, then the computer system automatically pauses it. Thus, in this scenario, user has not to worry about remembering to pause the video before leaving and neither the exact button for "pause". The operation of computer system if is this easy, new user will also be benefitted much. The other action that new user struggles too much is scrolling, they have no practise as how much pressure is to be applied to mouse and how much movement is to be given to the mouse. They generally apply more force or less force, before having a habit of using it. Thus, if the HCI developed performs the action of scrolling, like when the continuous detector and feature extraction algorithm detects the eye gaze to be at the bottom of the page, it automatically performs SCROLL action there. Thus, the ease to use the system is also a goal of the model proposed in this report. The other benefit here is that even an ***illiterate*** person could operate the system. Now, obviously initially he/she needs to perform some basic operations. Thus, if NLP (Natural Language Processing) is used along with HCI, then the illiterate person would be able to make much use of computers and also be able to operate at the first place. The NLP could process information and present him in form of natural speech, which could be well absorbed by him. And using the computer like navigating, scrolling, opening and closing, playing and stopping, all these would be if performed according to the natural behaviour of the user and after observing his actions. These responses would be natural and no special knowledge would be required to operate a general computer [36].
- ***No Technical knowledge would be required:***
  Some technical knowledge is required for operating the computer systems. But if the computer systems would be as interactive as other human, we barely need any of the computer specific knowledge and then too operate computer systems flawlessly [33].

- **Reduce the size of computer systems:**
  Input devices are an integral part of a computer system. They are the only means through which we can input an information in the computer system. And inputting data to the computer system is a very basic requirement. So, each computer comes attached with input devices (mobiles and tablets have their screen as input device, and they have all the circuitry to have the screen function as screen i.e. output mode and touch as input mode). So, as each computer has input device if we no longer need these large input devices and

the small camera at the top of the screen and microphone installation is sufficient, the size of the computer systems would reduce. For example, the general laptop structure is it is comprising of two surfaces, one is screen and the other is keyboard and track pad. Now both these surfaces are hinged at one edge to one other and generally 120-degree separation between the two surfaces is possible. The general use case is that the surface containing keyboard and track pad is treated as horizontal surface and that rests on some base and the screen surface is approximately in vertical plane. Thus, if we do not need the keyboard and mouse then the horizontal surface is not required and the laptop would be half of its present weight and width also half of the present value. Similarly, it could be argued for PC systems, where a keyboard and a mouse are generally connected to CPU cabinet. Now, if keyboard and mouse would be irrelevant, then a lop of desk space would be saved.

- **Variation in user skin colour and eye colour**
- **Variation in illumination**
- **Variation in background**

## 1.11 Difficulties in using proposed system:

**Irregularity of movement of natural organs of user than explicitly commanded input interface:**

The user if is in control of a input interface that is not natural and needs conscious explicit movement like mouse, keyboard, etc, then the irregularities are "MUCH RARE and IF OCCUR is easy to be CATEGORIZED as one" but the problem with natural movement tracking and unconscious input mode is to distinguish an irregular movement from desired movements.

**Solution:** The exact same problems have to be faced by humans when interpreting other's movements and there are a number of situations where natural things and general perception is very complicate for computers to understand but humans have a definitively correct answer. That was and is a problem but "ARTIFICIAL NEURAL NETWORKS" and deeplearning systems have an answer for it to some extent. They are able to duplicate the thought process of humans processing enormous possibilities and taking decisions in intricate and complex matters which was away from the reach of computers before these technologies.

## 1.12 USE CASES OF PROPOSED SYSTEM:

1. Autonomous cars
2. Driver drowsiness tracker
3. Webcam mouse
4. Intelligent camera adjustment
5. Face recognition
6. Natural HCI through mobile devices
7. Website User Heat Map
8. Eye trackers

## 1.13 Webcam input Interface Vs Traditional Input Modes:

The working methodology that our input model follows is the general template that any input device follows.

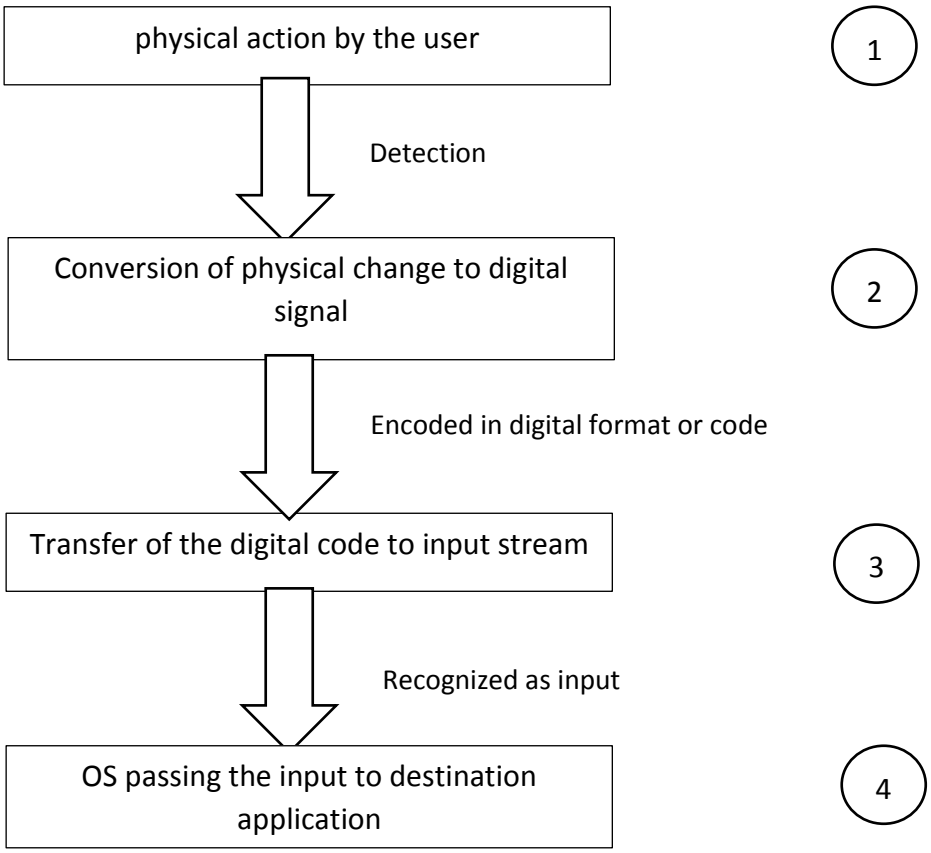



**Fig. 7: How the user action on input device is an input to programs**

So, the general flow chart that is followed by input devices is what our webcam-based input interface too follows. But it differs in the first stage of the Flow Chart in Fig. 7, as the later stages in the procedure is what any input methodology proposed have to follow. So, to qualify as an input interface any input model has to have its general layout corresponding to Fig. 7. The difference in various input interfaces is only in the implementation of the first layer. This is what characterizes as well as differentiates an input device from other. For the sake of illustration what

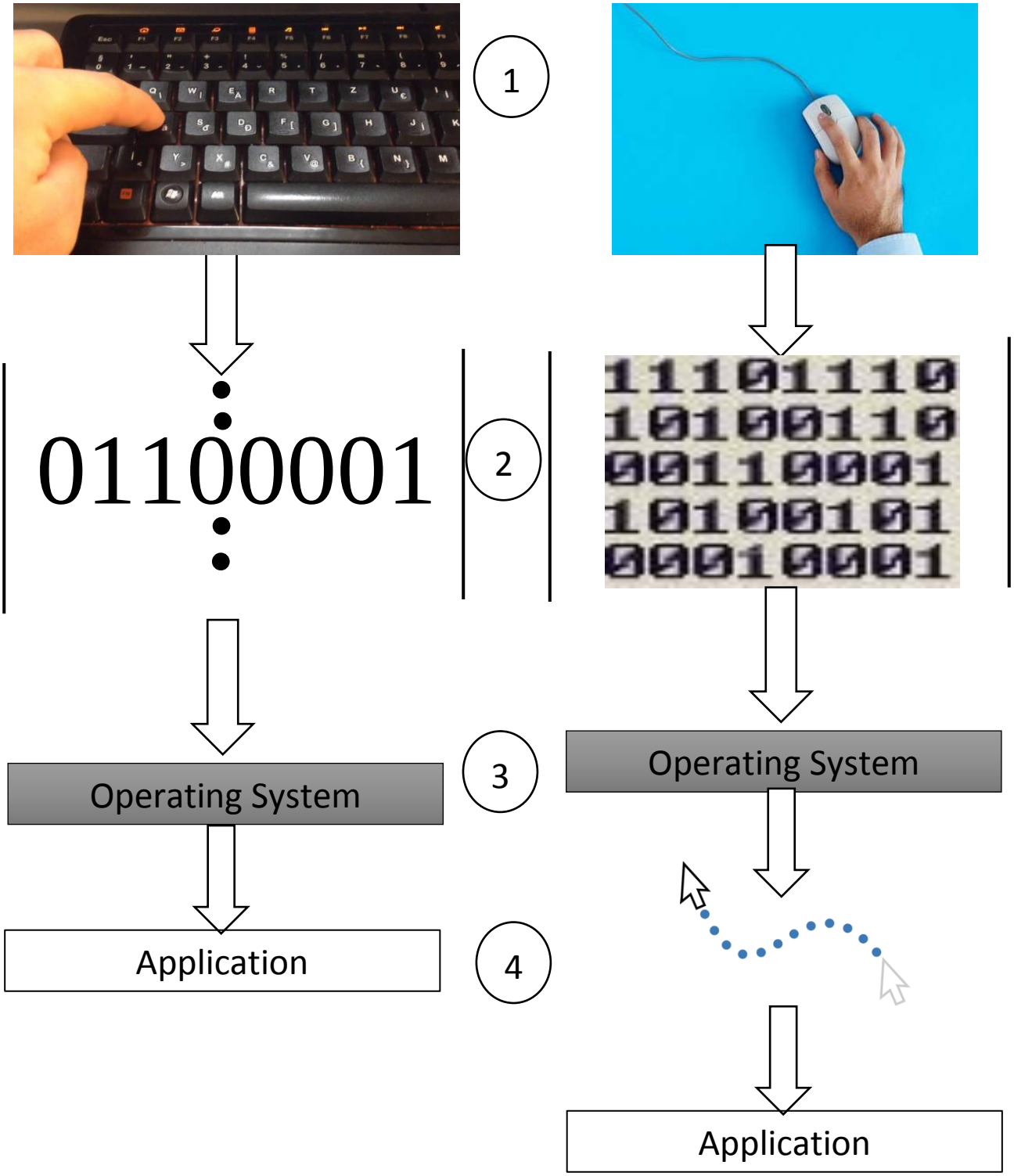



**Fig. 8: Interacting to applications through actions on input devices: keyboard and mouse**

Above in **Fig. 8** is the exact happening of events for the general input device pipeline that we have given in **Fig. 7**. The general template of an input device having 4 steps or sequential stages as shown in **Fig. 7** is mapped to the exact actions happening in **Fig. 8**.

The exact details of events occurring in each of these 4 steps in **Fig. 8** are:

**Step 1: MECHANICAL ACTION**

The most time consuming and slowest step of the pipeline is step 1, both in case of keyboard as well as for mouse. This step effectively is functioning as a **transducer**.

**"IT CONVERTS MECHANICAL MOVEMENTS TO DIGITAL CODES."**

This action is time consuming because the user has to make a mechanical effort and this mechanical effort is recognized by the computer system as it is converted to digital code. In our proposed system, this bottle-neck step is very fast as our system do not require any time-consuming effort or actions by the user but his state of mind is analysed by the system through observing them.

Keyboard: The mechanical effort is pressing of a key. A key on the keyboard is pressed to convey to the system the input that the user needs to feed.

Mouse: Physical displacement of mouse is the effort that the user needs to perform in order to inform the exact input information it wants to convey.

These actions are to be consciously made by the user and user has to decide the exact correct action that he has to perform corresponding to the input data he/she has. Thus, very little work is on the part of the input interface system. *The user has to map the input information to corresponding sequence of actions to be performed on the input device.* This sequence of actions is directly having a one-to-one mapping with the corresponding input signal to be generated. As for the input device, it is like a table scanning task. Where the user's sequence of actions be one entry in the table and corresponding input signal code be the other entry. For example, pressing a key on the keyboard, the user finds the exact locations and combinations of keys that should be entered and there is with the keyboard interface driver exact code for each key. Any sequence of keys which are pressed could easily be matched to binary input values to be given to input stream.

**Step 2: EQUIVALENT BINARY VALUES**

The mouse driver has the task of analysing the physical movement of the mouse and generate corresponding sequence of 0s and 1s to the input stream. Similar is the case with keyboard input interface driver, though its task is much simpler than the mouse driver. In fig. 8, the key that is being pressed on the keyboard is 'a'. So, the corresponding ASCII code for 'a': 01100001 is generated. Thus, this correspondence of physical actions to binary values is the work of step 2.

Once, **THE INPUT INFORMATION TAKES A BINARY FORM IT IS USABLE BY A COMPUTER SYSTEM.**

So, now the computer system is able to have an input in a format that could be easily decoded and understood by it.

**Step 3: OPERATING SYSTEM INPUT STREAM**

The corresponding drivers of keyboard and mouse are responsible to pass the binary input to the operating system. The drivers are generally under the control of operating systems and tasks like input and output generally needs the intervention of the operating system and user programs could generally not access them directly. Now, a days there is no infrastructure that has operating system directly interacting with the input interface of keyboard and mouse. But they interact with the corresponding drivers which in turn are connected to the input interface of physical devices.

**Step 4: OS PASSING INPUT TO DESTINATION APPLICATIONS**

The journey of an input data to the computer system terminates at the particular application that needs the input data. After that this data is no longer an input data but becomes stored data. This step is

usually very fast and depends on the execution speed of CPU. The CPU execution speed is usually fastest step in this input pipeline and so this step depends on the previous slower steps.

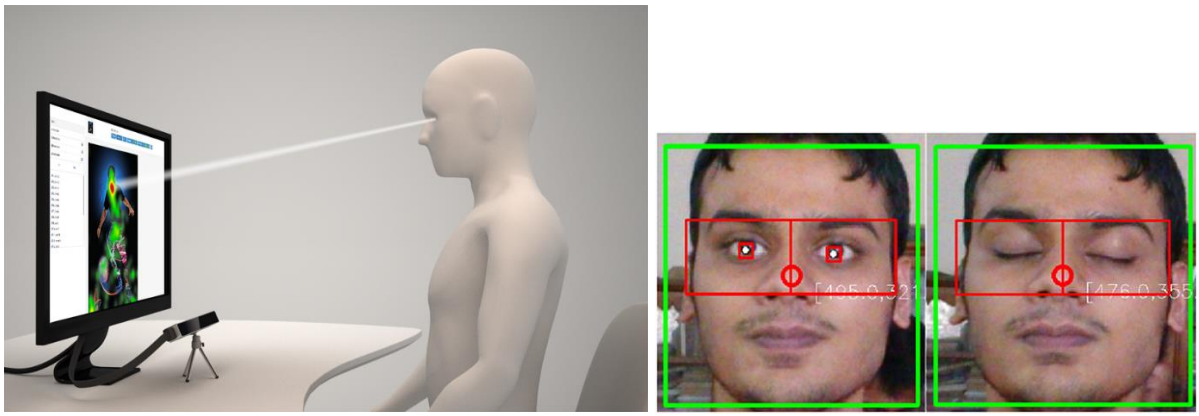


**Fig. 9: The physical actions that are recognized by our proposed input interface system**

The paper is divided into four main steps of: (1) detecting eyes as fast as possible by "harr-cascade classifier" (which is the most popular and most widely used algorithm to detect objects in real time), (2) and after detection we use "Local Feature Matching" to have track of eyes, (3) then we extract the eye pupil features based on projection method (simple and fast) and at last (4) we stimulate desired input that the eye is transferring into the operating system input stream. So, in all these 4 steps read the eye behaviour and the information abstracted in it to transfer input to the input stream and interact with applications and programs. This transfer of information to computer from eyes has to be in real time and the computations performed have to be bare minimum. The proposed system for the step one uses Harr-Cascade. Anytime as the frame from the video stream is chosen for evaluation, it is pre-processed into a new image representation called the "Integral Image" [1] which makes all the further computations fast. Then the detection step is modified AdaBoost learning algorithm. It is a degenerate decision tree like classifier that starts on the pre-processed frame image and in successive steps finds interesting areas in the image to focus on [1]. Each iteration of classification represents some features that it has learned from training examples and can go to 38 steps, each one more aggressive than the previous. This is because in training examples the successive stage classifier works on the images that the previous stage classifiers passed and it has to filter among them to the next level. So, after training the AdaBoost algorithm stores in it feature classifiers (many successive stages) and each classifier looking for some feature that it has learned. Increasingly more classifiers are "cascaded" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The step two is tracking which is done by continuous image analysis. In this step we at fixed interval of time analyse the live video to get the state of the eyes. The present state is then compared with the previous states and this kind of adaptive algorithm that also mainly focuses on only the important aspects of the image is very computationally efficient. Then in third step, we use detailed projection analysis on the detected eye region to get full information of gaze, open or close state, etc. We can get data for detailed analysis for drowsiness, attention level, emotions displayed through eyes. These inputs may be used for some advanced interaction system. The last step, fourth, is the bridge between the extraction of exact state of eyes in the video frame and informing the computer system by sending appropriate input signals in the stream. The least computationally intense and quickest step is this which could be a simple and fast program code that maps the eye states to accurate inputs for simple program requirements. Like having gaze not in direction of the screen may introduce command for pause if video is playing. This approach maps

continuous input class to binary or multi class outputs using a “parameter” value for mapping. For example, pupil size in the detailed analysis is if less than the “parameter”, then eye is considered closed, and thus a continuous input class is mapped to discrete output cases. These algorithms have the potential to revolutionize the future computer-man interaction. Eye and lips are two of the most interactive organs to communicate many things and if we successfully make a bridge for computer to interpret inputs directly, then many new and unheard-of areas will be unlocked.

## 2. Literature Review:

The research in the field of eye tracking till now could be categorized in two basic types:

1. Special Equipment setup for tracking
2. Tracking model on existing hardware and software resources

The striking difference between the researches in both these types is that in the one "which requires special equipments" are more industry oriented and some are the development of special products. But almost whole of the research in the without any "equipment tracking system" are all done by academia or by individual researchers. So, the content of both the researches gets influenced by this fact about their nature.

Now these two categories are further divide into several approaches like.

**1) Special Equipment setup for tracking**
   a) Infrared wave usage
   b) Electrode immersion in head [31] [32]
   c) Special set up of cameras

**2) Tracking model on existing hardware and software resources**
   a) Geometric analysis
   b) Convolution Neural Networks
   c) Classifier Algorithms
   d) Color analysis methods
   e) Pixel coding methods

### 2.1 Special Equipment setup for tracking:

Broadly all the approaches in this category can be classified into two basic categories:

- **Remote setup**
- **Head mounted or equipments as gears**

#### 2.1.1 Remote Setup:

This is an approach in which the working setup required is at a distance from the users and users have no physical attachment to any part of the experimental setup. The remote setup is stationary and analyses the user characteristics, it is assigned to watch. The connections of the equipments analysing the user to the computer system is usually fixed and stable. The most common equipments used in this approach are two:

- cameras mounted at various suitable angle to get full information of user:
  In this approach usually two stereo cameras are mounted or in some cases only one stereo camera is mounted but at an angle and place that is special for analysis.
  - Two stereo camera system: Two camera takes their own video input and the user is analysed based on the combined output of both the cameras.
  - One camera at special location: Usually the centre of keyboard is the location for the camera, as the user stream captured at that location is much helpful in detection of specific features that would otherwise be not so striking.
- special purpose cameras that can watch special features of the user:
  This approach requires specially designed cameras having the functionality to detect special features of the user.

#### *2.1.1.1 Thus, the benefits of this approach are:*

i) The user feels no discomfort:
One of the prime characteristics of remote equipment setup is that it is in no way attached to the user. There are no wires and connection to the user body and hence, to use the equipment requires no extra effort. The user does his work and in parallel the equipment performs its tasks. So, no direct physical dependency of the equipment on the user is a very powerful argument in favor of this type of systems and make them much more lucrative than attached gears setup. User comfort is prime goal of the computing system, every thing in computer is just happening with one goal only, to give comfort to the user. So, the systems that are not remote generally tends to be much more accurate and faster but many a times comfort could be the deciding factor in deciding the winner.
Hence, making the system remote i.e. at a distance from the user introduces a number of difficulties that were not there in wearable equipments but we accept its limitations as it is comfortable to the user.

ii) Free movement of the user:
Attachments do not restrict the motions or movements of the user. Generally wearable equipments restrict the full set of motions that a user could perform. So, restricting the movement set of the user is a great demerit and it is solved by remote equipment setup.

iii) No harmful side effects of using the system:
Generally wearable equipments are the equipments that are attached to the user body are harmful for the human body. Two central ways in which they risk human health are:

- Infrared Radiations: The most popular wearable gears are all using IR. There is a profound benefit of Infrared radiation in eye detection mechanisms, as the reflection characteristic of Infrared Light from the pupil and cornea of the eyes are strikingly different. Due to the broad difference of the reflection prints from these two areas of the human eye, it become very easy to locate the pupil exactly. The computations needed are also very small in Infrared Light methodology as we have just to identify two segments of reflected image that are having particular characteristic and these characteristics are very prominent in these reflected images. But blasting these rays on eyes are not healthy and some cheap Infrared Light gadgets emitting light on eyes are dangerous for prolonged use. Thus, however cheap the product could be if we use Infrared Light and however accurate and computationally light it is, we have to compromise our eye health. So, remote equipment is much needed.
- Strict pose of the body:
  Many an attached gear system restrict the body state to be one out of few options available. Like some head gears require face to **straighten** up and be in this position for prolonged hours. This is not only uncomfortable in the first place but is bad for our health too. It may lead to back pain, neck pain and other body malfunctions. Thus, remote equipment setup solves a major problem of adverse effect on health and their utilities are much increased due to this factor.

iv) Much more natural and free state of mind for the user:
The user void of any attached equipments show motions that are natural and unrestricted. Many a times motion and movement detection in wearable equipments case suffers from the fact that if the user is conscious of the gadget and its working, then the user tends to suppress his normal body flow and act in strange manner. Thus, having a knowledge that the equipment is tracking the motion leads to controlled motion display by the user. This

leads to irregular working of the equipment, and at the same time limiting the action set of the user is not a successful approach.

v) Less focus on user comfort wearing it:
Major amount of attention of wearable or attached equipments design goes on the area that it should be comfortable for the user to wear it and use it. So, automatically the attention to the computer science area suffers and many a times trade-offs have to be made based on user comfort wearing it and some functionality or algorithmic implementation. For example, if a wire of more width is required from the head gear to the computer system to increase the bandwidth for sending parallel data may not be adopted, because a heavy wire is unsuitable to be installed in head gear for user comfort. These all trade-offs and design peculiarities are never an issue in remote equipment mechanism as they are situated at a distance from the user.

But despite of these many advantages, the disadvantages exists of remote equipment system, mainly due to the fact that it is not as near to the user as head mounted systems are.

***2.1.1.2 Disadvantages includes:***

i) Cost of equipment:
The overall cost of equipment of remote setup system is greater than head mounted systems. The prime reason is as the head mounted equipments are very close to the eyes and they do not have to search for eyes in a large search space. And also, as they are so close to the eyes of the user, the amount of information that they extract makes them very easy to compute what they want. Remote setup system takes a lot of time to compute as the user eyes are too far from the equipments and the search and feature extraction is not so straightforward nor so easy. So as the detection requires heavy software's to predict accurate results, they are relatively costlier than the head mounted systems whose major problems are solved because of them being so near to user.

ii) Increased amount of computation:

The computation that the remote setup system has to do is the cost it pays for all the user comfort and freedom it gives. IT is sure that remote tracking mechanisms are at a distance from the user, allowing him/her to have much freedom in the movements that he/she wants to make. But because these systems are at a distance, to predict the position and movements of eyes it has very much more work to do than the head mounted systems. The amount of computation also means that the equipments that execute these computations have to be more powerful than those required in head attached mechanisms. As these computations have to be executed very fast or at least so fast that real time implementation be possible, power of computational equipments are increased which in turn raise the cost. The detection of eyes from far distance also requires complex algorithms, contrary to small modules for head mounted systems.

iii) Less accuracy:

The accuracy is lesser when we compare remote system a t a distance from the users as the input it gets has a lot of **noisy data** in it. The major remote mechanism has architectures comprising of camera systems and the input to these cameras is whatever is in front of the camera. This is in stark contradiction to the head mounted systems as the eye is approximately the only thing that is in front of them. As the eyes are directly in front of the head-based equipment systems, the work to be done for extracting the eye information is very much less. And comparing the remote equipment setup, any general object could be in front of it and even eyes are too far from them, so detection has to be

much more advanced than used in head-based mechanism. And so, this generality of input data makes this system less accurate and not much reliable.

iv) Limited usability and functionality:

The head mounted systems are so close to the eyes that they can detect the state of all intricate parts of the eye. And thus, the analysis and evaluations that they can do are on much advanced level. They can

- exactly pin-point the position of the pupil
- evaluate the size of pupil
- the position of eyelids
- cornea status

This amount of data is very much less than what the fixed, head mounted systems provide.

vi) Constraints on background, lighting, skin colour of user:

The factors like the background and its features as well as face of the user and all its features are noise for the eye detection system. The head mounted wearable systems are just fixed to look for eyes and they do not input the background data at all. They also do not take user's facial features or other data, if eye detection is the concern. But the remote systems are there at a distance from the users and do EXCLUSIVELY not focuses on eyes but they take the video of whatever is in front of them. Thus, extracting the state of eyes when the only data to the system is of eyes is much simpler than, when we have a whole lot of information and eye is a tiny part of it. So, we have to first locate the eyes in remote installed systems and then only analyse the state of eyes. It is not at all as straightforward in the case of remote systems compared to how much easy it is in head mounted systems.

vi) Mostly calibration is required: Many a system require for calibration. The head mounted wearable products are mostly sponsored by industries, they are ready to use from the start. But the remote systems are mostly not industry products but academia projects, they are not to be directly used from the start. They require some user expertise or familiarity when being used.

### 2.1.2 Head mounted or equipments as gears:

The google glasses or other wearable devices mounted on head are either using a camera setup very near to the eyes of the user to collect eye information or Infrared Blasting is used. As the circuitry is so much close to the eyes of the user, the information tends to be much more accurate and much less computations are needed. Many advanced analyses on the user eyes is possible due to the setup of product. But this comes up with the disadvantage that it is directly attached to the user and may not be comfortable all the time and could restrict certain movements that the user wants to make.

## 2.2 No-special Equipments required:

This approach is much more complex to develop as no help from the equipment setup is present. We have to use the general devices that comes with the computer systems and build our model on top of it. When compared with the special equipment setup system, this system is particularly weak. But some very important advantages of these systems, call for their popularity like:

- **Portable:**
  Since, the model has no strict requirements and needs, it has maximum portability, and could be implemented to many different systems. As no specific system need is there, so no restriction in applying on any particular device.
- **Integrated to maximum devices:**
  Absolutely no help from the hardware or software is needed, so could easily extend to many devices. Once many devices start having this feature, it could be vouched for in the devices that do not implement this and would be forced to implement it.
- **Cost free:**
  No additional cost to the end-user of the system manufacturer.
- **Could take a place as general model in computing:**
  A model that works on a special environment and is not general and robust is generally destined to be serving to special user groups only. But general models that are acquired with no EXTRA EFFORT, no EXTRA COST and is EASILY AVAILABLE could make the model a generality in all computing devices if it is effective and useful.

## 2.3 Detection algorithms:

The popular face and eye detection algorithms are of the following types:

1. Colour based models
2. Geometric approaches
3. Feature based approaches
4. Convolution neural networks

### *2.3.1 Geometric Approaches:*

The detailed analysis of pupil is done graphically and thus the only computations are of distances and shapes for knowing the exact state of the pupil for gaze direction and pupil size. Using graphical calculations focus on the black spot of pupil surrounded by the white cornea. Some fast algorithms of computer graphics like flood fill algorithm (or boundary fill), and also as the computation at this stage is only black and white, the boundary of the pupil could be constructed in the detected eye region. The pupil boundary could then be analysed for its shape and size. The shape of the pupil is given by the averaged maximum and minimum points in direction parallel to the line connecting both the eyes and the averaged maximum and minimum points in the direction perpendicular to it. These calculations could give the estimate of the pupil shape. The size could be calculated as the ratio of the pupil averaged area to the eye averaged area. The ratio if less than the threshold would signal eyes closed and if ratio is more than threshold, indicates eyes open.

The key innovation of this model is to discover a fact that was hiding in plain sight “the webcam is kept stationary and the movement of head is relative to it” [16]. Then to combat the pioneer obstruction in the field of webcam captured video analysis is differential lightning, illumination and other regularities. The novel concept of this research was to use adaptive dynamic thresholding to binarize images [18]. The steps of action to convert an image from webcam to a totally marked and accurately detected image is:

1. Binarization: Convert the image to binary using dynamic thresholding.
2. Eye Image: From binary image, the geometry structures of eye image are selected out.
3. Eye Corners: The positions of both the corners of eyes are detected using "estimation-based model" based on geometry features of the eyes.
4. Center of iris: Matching of an "Iris-Boundary Model" as well as "Image-Contours"

2.3.1.1 Limitations:

1. It worked on images and not live videos.
2. It had no real time needs. It was just to analyze features and the use of extracted data was not in any real time application, so reducing the calculations and making it quicker was never the aim. In fact, for increased accuracy, it compensates calculation speed.

The Geometric approaches like the rank order filter algorithm by Ren and Jiang [12] has a definite accuracy feature [42]. It combines the biological feature of eye and tries to investigate these features through geometric calculations [26].

### *2.3.2 Color based algorithms:*

The colour-based algorithms like the one by Nasiri [13] we a different colour space called $YC_bC_r$. The normal image obtained by webcam could be encoded in this format and then geometric tests are applied to locate the eyes, using the geometric information of the eyes being a certain position on face and both eyes are replica of each other [24], etc.

### *2.3.3 Detection algorithms:*

The haar cascade algorithm is a classification algorithm [21]. It uses features in an image to detect an object. Detection is the first task in our model. The eye detection algorithms that are based on Convolution Neural Networks (CNN) are generally most accurate but not fastest. The geometric detection algorithms are compute intense and we cannot put so much load on CPU for input interface. The colour-based detection algorithms are generally carried out in two to three steps. In which the original image is encoded for each pixel based on some colour property. Thus, in subsequent steps we get the binary representation of the image such that some features are very obvious and these features are due to the colour pattern of the image. So, colour detection algorithms are though limited by the colour pattern of training examples. That is if all the training examples are of people of some particular ethnicity, then the algorithm automatically assumes that face is of this particular colour only. And if any face is given to the algorithm that is of different ethnicity, the algorithm incorrectly dismisses this colour **face as not a face**. The solution is to train with many many types of face colours and ethnicity, but it makes the algorithm slow and more compute intense as now it has to match from a very large feature set. This algorithm suffers particularly from the varied illumination, background and camera angle. One such instance is this type of algorithm finds very hard to separate faces with colours similar to the background. So, it incorrectly detects it. The proposed solutions to these problems of colour-based detection algorithms is to pre-process the image in such a way that the effects of lightning, illumination and background is minimized. Also, foreground is made more visible and the background is somewhat blurred, having the argument that foreground is most often the face region and background is the uninteresting area.

Out of many detection algorithms proposed, LBP and Haar cascade are two suitable candidate detection algorithms for our project, meeting the requirements of real time, fast and least compute intensive.

The LBP is a relatively very light on computations. It creates a binary map for each pixel in the image. The binary map is computed by LBP algorithm according to the formula given in [10]. The main contribution of this paper is that the size and rotation of the target is included

in the algorithm and thus, the algorithm generally performs better when the object to be detected and tracked is deformed. This is very much useful point for face recognition system because, the users tend to come nearer the screen to look something closely and this changes the scale of face in the image. And also, the other most important movement of the face is rotation due to the fact that face is pivoted to the neck and rotation is the most general movement our heads make in front of screen. As rotation distorts the face feature, detected a rotated face is tricky but this algorithm does this task pretty well. Translation is another movement but it does not generally involve any deformity, it being the simplest of all the movements could be detected successfully by all the algorithms talked here. As claimed in the paper, **SINGLE FRAME IMAGE PROCESSING TIME** of the proposed algorithm is 0.0853s compared to Harr-cascade's 0.0973, the paper gives some promising grounds to explore. The accuracy of Harr-cascade to get a value 2.5 compared to the value of the proposed algorithm is what could be achieved if colour correction operation be applied to the existing algorithm.

The one of the most influential work in the object detection field of computer vision is the work by Viola and Jones [1], working on AdaBoost process that gives classifiers, selecting the most significant features of objects as the basis of classifiers. As the classification algorithm if classifies a part of image as an eye, then the eye is said to be detected. So, the classifier that AdaBoost constructs stores those features for comparison that are most critical in other word the minimum number of features are included but the classifier is much much strong. This makes the classifier ideal for real-time tasks. But the real trick pulled out in the paper is to have a cascade of classifiers. This means that AdaBoost is used for generating classifiers that are most-efficient but we use a series of classifiers and the paper puts forward the idea in which these many classifiers are combined that results in fastest detection. The general idea is to have weak classifiers work first and discard the regions that are most obviously not the target. And thus, at each step we are left with a region of interest smaller than the previous which can now be worked on by stronger classifiers and further narrow down the region of interest. The benefit we get through such mechanism is that strong classifiers needs more time to compute than weak classifiers, as the amount of knowledge with weak classifier is less and it can analyse the region with that knowledge faster. The strong classifier has a more comprehensive knowledge stored and they take more time to work on. So, weak classifiers work first and discard uninteresting part of image, such that the more time taken by strong classifiers are well spent on important region. So, all in all we do as little computations as possible and yet detect the object. How these different classifiers work is the proposal in viola's paper, such that the detection is very fast. And really Haar Cascade is one of the fastest algorithms to detect objects. The proposal for a special pre-processing to be applied to the image before analysing it, to much reduce the computations in further steps.

The immediate response to the Haar-classifier for further improvement in execution, was to use more than one detector in parallel and combine the output of all of them to give a detector. But, the amount of improvement in accuracy was very slight and obviously the amount of computations increased much. So, as per our requirements here, the one classifier that is trained perfectly is better.

The strong detectors in the cascaded detector system can detect some scaling and translations of the features with respect to the base features. This could be very useful in the fact that if the face is moving rarely, then those slight movements could make no difference and the next frame could be easily matched to the previous one. The light-weighted computation nature of this algorithm is shown by the fact that sixty machine-instructions are used by a 2-feature classifier.

Then talking of Neural network-based algorithms [23], it is important to note that the first major paper in face detection using neural networks was by Rowley [2]. The differences in

the face detection from 1996 to 2019 is adequately shown by the ANN put forward by Rowley and the CNN that are used for face detection now. Rowley proposed a system of detection combining 2-detection networks. The idea somewhat resembles Haar-cascade, in the sense that one of these classification network is weak and other one is strong. And they are cascaded efficiently that the time taking computations that are done by the strong classifier is only on the small important region of interest. The key ideas from this paper is first of correctly pre-processing the image. Histogram equalization is done on the images after being gray scaled to improve the contrast, which further helps in giving features that are prominent and thus leads to increase in accuracy and speed. The ideas for eliminating the effect of illuminations and lightnings in a rather much simple way is to obtain the approximate region of face in quick steps and then to further analysing it accurately and with speed we correct the lightning. We first calculate the average brightness of all the points of face from the grayscale format image. Then we apply a linear transform to all the points to have a constant lightning over whole face. By such procedures, no feature information is affected by the particular lightning and illumination conditions but only the face's actual features are stored as features in the knowledge database. It also improves the searching and matching of features of the input face to the stored database, as now the search is only guided by actual facial features and not any noisy features that could creep due to unequal or special lightning conditions on the face.

### *2.3.4 Eye tracking algorithms:*

Tracking algorithms are dominated by optical flow algorithm which was first proposed by Lucas and Kanade [3] in 1981. The optical flow is the movement of the region of interest in the search space. This region of interest is the object that we are tracking. Thus, the change in the object in two frames is calculated by optical flow. The change is calculated by some features of each pixel. The primary strength of this algorithm is that it need not compute the pixel by pixel difference between both images to spot the difference. But what it does is it starts searching the object in the next frame from an assumed value or a location it deems most likely to be found. Thus, the search starts from a candidate space and if that is not the object to be detected then we go to other locations to search. So, a major fact that should be known is, that the strength of optimal flow is that it already starts from a candidate solution and has to search less as do not have to search from start. But this introduces the problem that optical flow can measure small movements only.
So, optical flow is a great improvement over exhaustive search but recent methods are also fast and one thing for sure is this algorithm aged really well and still finds its place in one of the most prominent algorithms of object tracking.
The next approach is a hill climbing inspired approach, in which the search for the candidate object starts from the location of the object in the previous frame and it searches candidate frames based on the degree of matching the candidate object to the object that it is tracking. The search is guided in this manner and the appropriate object is found in the image through a series of wrong guesses but each guess in turn tell us what should be the most appropriate guess this time. But the hill climbing suffers from the usual disadvantages of hill climbing algorithms like:

- Local maxima
- Plateau
- Ridge

Meanshift between two images in a video is traditionally applied to get the difference between the two images, which gives the movement undergone by the object and we track it based on this movement. Meanshift is fast but it is not robust, i.e. it do not work well in case

of deformations in the object, rotations and other such non-linear transformations. So, the paper by Liu et. Al [10] shows a way to use LBP Operator and in a way that the LBP map for an image is when generated, it stores the prominent features as well as the rotation information for the features. Bhattacharyya coefficients are used to match the pre-processed and encoded images. These coefficients measure the extent of similarity of features between the two images. When the value of these constants are the features in the images that are being compared are matching more and the large values of the Bhattacharyya coefficients means the matching between the two images is not much. So, to give the non-linear nature to the normal Meanshift tracking algorithm, these Bhattacharyya coefficients are minimized. The experiments further in the research signifies that if the colour feature information is also used in describing and encoding the image features, then the non-linear Meanshift algorithm we get is having both speed and accuracy to be used in real-time systems.

The other fast detection algorithms use the concept of identifying moving objects in the image frame. If the object to be detected is the only moving object and the other objects are static or very slow to move. The background is usually static in personal computer systems and the laptops sometimes. As this assumption is somewhat dangerous to apply to all the computing systems including mobile phones and tablets, we in our model do not use it. But this assumption is very much valid in specific computing devices like the PCs and laptops. So, the idea put by Li [11] is to encode the images in a manner to highlight the part of image that are moving. The algorithm used in that paper could also be utilized to code videos and develop a new video coding format. Three frames are required to analyse the moving object. The working is such that the unchanged features from the images is subtracted out, leading to only those features that are changing positions. Thus, once the changing features are detected, with the other information, the exact moving object is obtained. Thus, this continued detection is said to be the tracking system. The system could be very fast and efficient if the background is not moving or have a minimum movement like the clouds or wind on grass [40].

Local Feature Matching is a tracking algorithm proposed by Liu [6] which matches features based on the information of edges and then getting the SHAPE CONTEXT from this edge detection. Then from the SHAPE CONTEXT the shape is extracted and assumed to be the detected objects, which by the FEATURE MATCHING calculations is tested for. The FEATURE MATCHING computation then gives the value of difference between the features of the candidate object and the feature information that we have about the object that we have to detect. This value only is further used to choose other candidate locations in the image and hence drive the search algorithm to eventually reach to the target. The feature difference value is used to give the new location to look for the object by using a voting system. The various region of interest is analysed for being the candidate region and these values vote for the next region to be checked, the region out of all the proposed regions that gets the maximum voting in favour is the next region to be analysed. Thus, on the basis of similarity of this region and the object we are searching to match gives values for next voting and prediction and it iterates till we reach to the destination. In some sense it resembles hill climbing in working [38].

### *2.3.5Feature Extraction algorithms:*

The tracking of faces through webcam is intensely studied in the paper by Zheng and Usagawa [4]. Their main focus was not to develop or improve the algorithms that are applied in the field of eye and face tracking, but they focused mainly on what are the requirements of the detection mechanism through webcam. What is the characteristic of the video stream taken by the webcam and how to apply the existing algorithms properly to raise efficiency? What are the characteristic problems when working on images by webcam and their solutions? To do this it does through study on the movement of the pattern and the eye movement in different types of motions like abrupt, focused. It measures the threshold value

of movement that sets how many frames should be analysed in a second for optimal performance based on the movement capabilities and limits of the eye. They found that from a smooth motion to an abrupt movement approximately the eyeball speed is “30 degrees per second” to “400~600 degrees per second”. Thus, the crux of the matter is at least 4.5 frames should be analysed per second.

Adding further to the findings, it should be noted that the value advised in this paper is 5 frames per second. Though an implementation that analyses 4 frames per second performed sufficiently for the basic action detection that is required in our system. But for advanced eye feature detection more analysis is required.

The feature extraction works in a manner that the movement of eyeball is recorded and based on the movement action it is proposed where the gaze position is on the screen. Further, for simplifying calculations the 3-D circular motion of eyeball is approximated to linear motion. The results in this paper shows for the basic action detections as required in our project this simplification is valid [4]. The method claims to have an accuracy of 94% and having a speed of 8.2 frames. These metrics are signifying that the real time application is very much achievable.

The exact state of the eyes could be detected by numerous proposed GEOMETRIC, CNN and classification methods. As the accurate analysis of the specific state that the detected eye has needs large computations. Geometric methods as well as CNN methods [28] are much costly in sense of CPU need. Already we had set the “detection” and “tracking” algorithms used in our model to be the most time taking, so we were in search of a light algorithm that detects the eye states exactly but is light on computations. Also, since at least the detection of the eye state needed in our application in this report could be fulfilled by this Projection method proposed by Liu and JIA [7]. In this method we first have a pre-process stage to make the eye region that is detected to be just containing eyeball and cornea feature information. Now, this coded image is analysed for horizontal and vertical projections of the eyes. The algorithm estimates the size of eye ball in horizontal direction and the size in vertical direction, these parameters are used to detect the exact state of the eye. The filter used is the median filter such that in analysing the eyeball horizontal and vertical width, only focusses on the eyeball black part surrounded by cornea and not affected by noises. The horizontal projection and the vertical projection are combined to say about the state of the eye. The basic states like:

- Open or closed eye
- Gaze in or out of screen

are easily detected with much confidence value.

### *2.3.6 Correct input signal generation:*

To get information of the input stream of different operating systems, the internal details of two most popular operating system has been analysed, Windows and Linux. The WINDOWS INTERNALS (2009) [8] is a great reference and could be termed as the only authority to cover Windows Operating system. As the Window’s source code is not publicly available, something like input interface could be only integrated to the operating if its inner details are known, and surely this book is sufficient to add an input interface to the windows operating system.

For Linux Lions’ [9] running commentary on whole of the UNIX source code is very much informative, insightful and sufficient for implementing an input interface along with the existing input interfaces. His book is a definitive guide on UNIX and is through as each and every line of Linux source code is fully explained.

## 3. ***Methodology:***

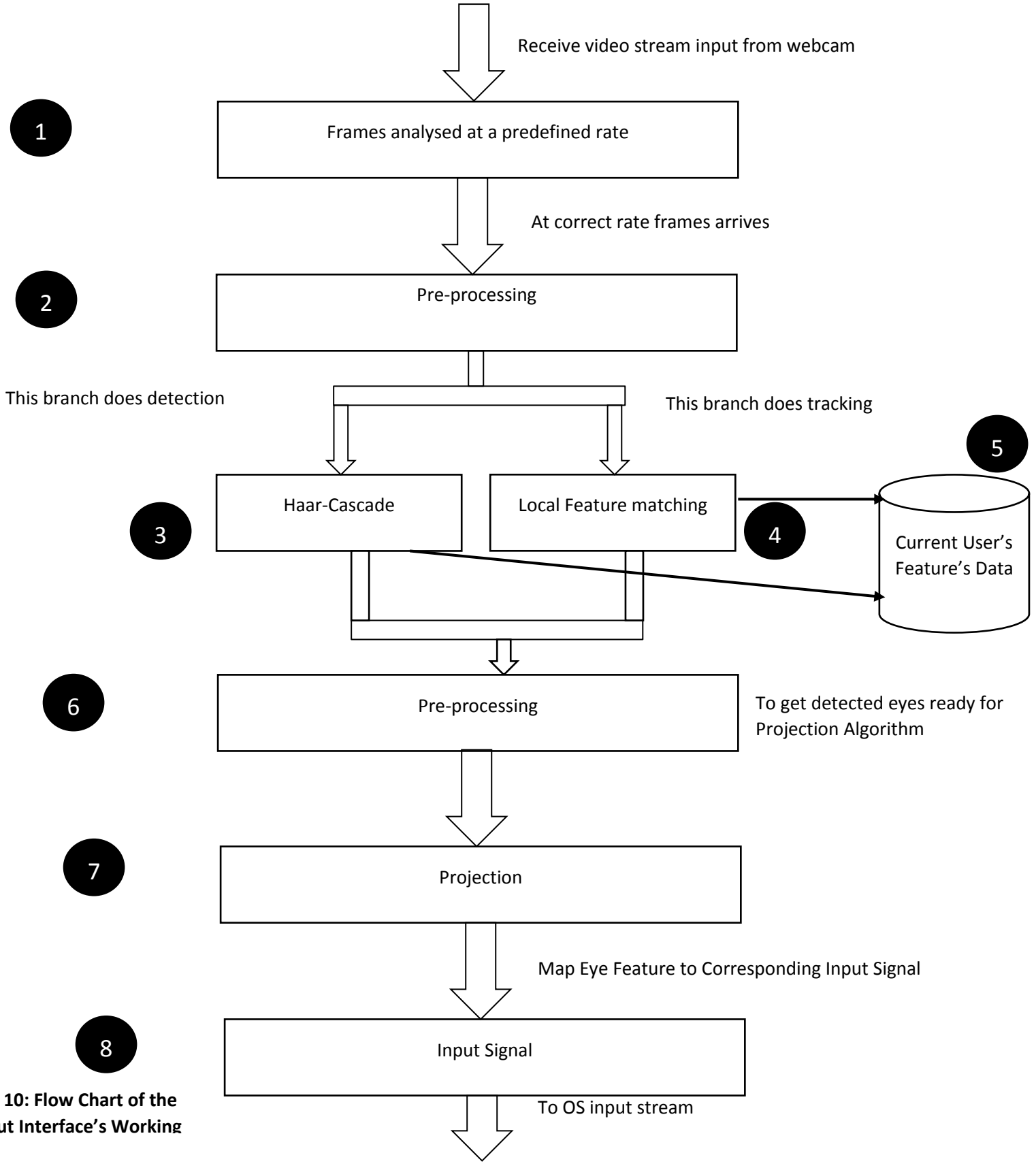



**Fig. 10: Flow Chart of the input Interface's Working**

**Fig.10** is the flow chart having eight elements numbered as shown and represent the functional flow of the model that we propose. The key to note here is that out of these eight elements, we have six stages of the flow diagram. These six stages are such that, elements 3, 4 and 5 are in one stage. So, the most time-consuming steps are 3 and 4 because, 3 represents the detection algorithm and 4 represents the tracking algorithm. Then the step 7 is meant to take rest of the time. **All the other steps are constant time quick procedures.** Here is the expansion of the most important part of the flow diagram:

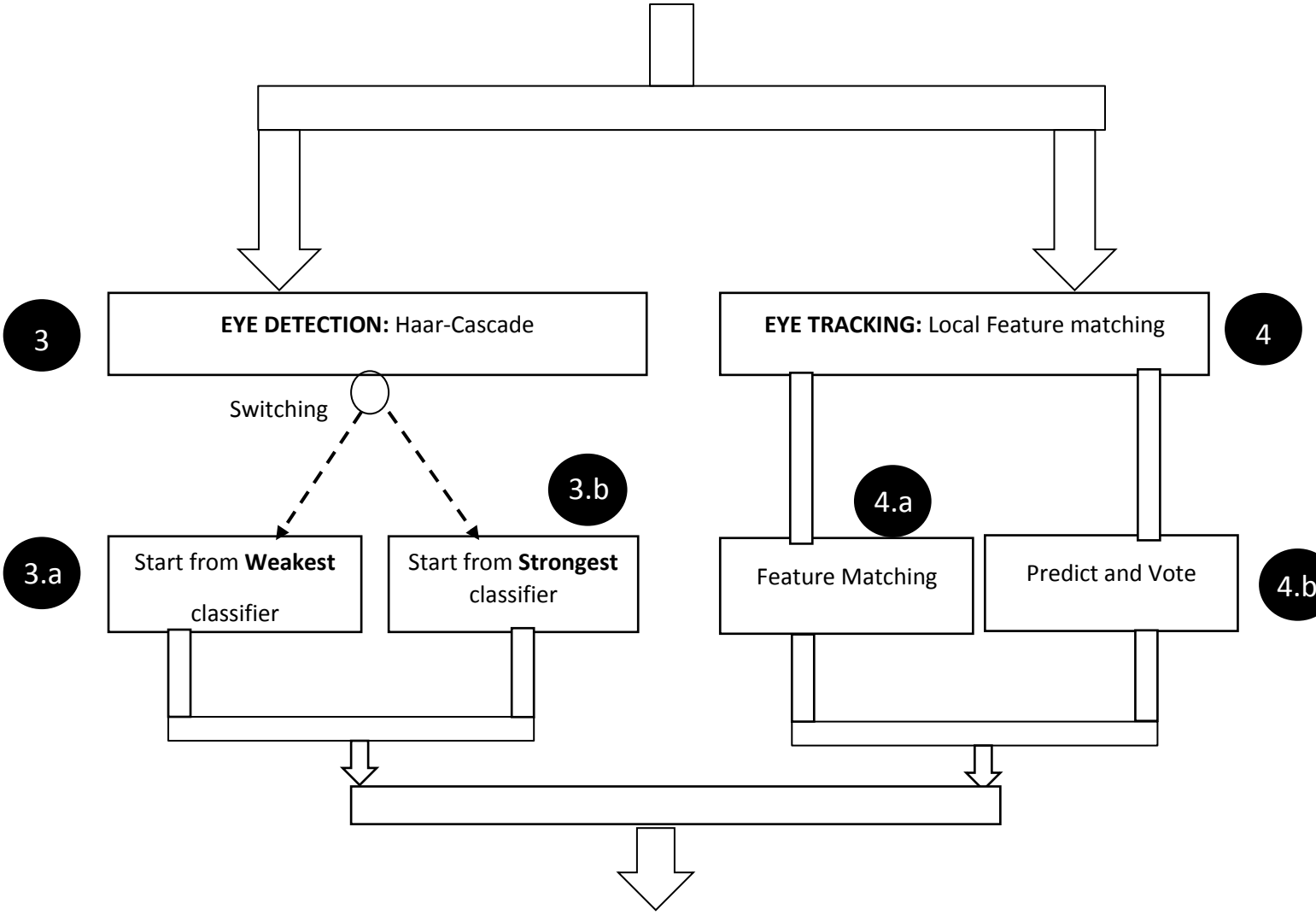



**Fig. 11: The Sub-parts in Detection and Tracking steps**

Now, these are all the steps that are taken by the algorithm to act as input interface for modern day computer systems. Through these stages, the journey of a chosen frame of webcam video input is converted as an input to operating system. Thus, webcam acts as an input device and could be used to interact with programs and the operating system just like what we do through mouse and keyboard.

The steps are elaborated in the following section:

## 3.1 Step 1: Frames analysed at a predefined rate

The input to our input procedure is the video frames from the webcam. The frames captured by the webcam is at least 15 frames per second. But we do not have to analyse all the frames that the webcam produces. Consecutive frames are mostly repetitive or a slight change and computation of such large number of frames per second is a very large overhead. Also, it is experimentally shown that for simple actions like video play and pause, and scrolling, analysing minimum 5 frames per second could also make the illusion of real-time response. So, we use this parameter of 5 in the working of our model.

**Analysing video stream 5 times in a second:**

The implementation includes the experimentally verified assumption that the time frame of 1/5 in a second is the trade-off for continuously tracking the eye in the video stream.

**What if due to worse case input, we do not detect a face in 0.20 seconds (5 frames a second speed):**

We move forward by giving it time to continue its computations and delay next frame detection. As some part of time of next frame could also be used in detecting the face in previous frame and only after that the computations of the actual frame that was scheduled starts, there arises the problem of congestion in the pipeline. So, we solve this problem by informing CPU to dedicate more computation resources to this input mechanism. As when an irregular abrupt movement is made, it is generally the chance that eventually the user will settle down to a pose and after that the movements would be slow. But till the user settles down, we have to track the face of user in a situation when he is moving too much. As we have already requested CPU of more execution time to our input interface, we can increase the frame rate for the time being. This compute intense phase is essential to track the use in his large head movements. Thus, when the user movement slows down, which could be indicated by very quick detection of the face in frames, we resume normal working. Thus, this adaptive approach allows us to never miss the track of user and also release the extra CPU resources when we do not need them. This phase of compute intense cycle is feasible as it allows us to have a constant track of user and never lose his track in the frame. Thus, had we not allocated the upscaling of CPU resources for input interface, then user track would be lost. And in the next frame we would have to detect the user face from scratch without a clue. Detecting a user face from scratch is much more computationally expensive than tracking the user. As in tracking, we have the user face location of previous frame and the face in current frame is close to the previous frame, resulting in low computations.

Other approach that this paper proposes is the one without involving a need for compute intense phases is by prediction and verification method. for failure of the DETECTION of EYE due to abruptly large motion, we move forward by assuming eye gaze is not towards the screen or the motion is a noisy movement and we have to ignore it, and verifying it instantly by “Face-Detection”. In this framework, we use face-detection for quick and assumed response in the case of swift motion, we could use eye-detection in the frame to localize the eye after the rash motion. But using face-detection at this step has several benefits: (1) after failure of our EYE TRACKING, we move forward assuming eye gaze is not directed to screen and so some conclusion should be as fast as possible, and face detection is faster than eye, (2) in case of large abrupt movement (as our methods fails) there is a great probability that there is no face in the video frame and (3) if a face is successfully detected, eye position prediction is made quickly according to the state and deformity in face status. Then once when we localize the eyes exactly, we see if what we assumed the eye state was correct, we move forward. And if we were wrong, we correct the things. This approach does not need more CPU resources at any instant and is pretty much accurate according to the time it needs to react. But less computations needed is at the expanse of some loss in accuracy of the system.

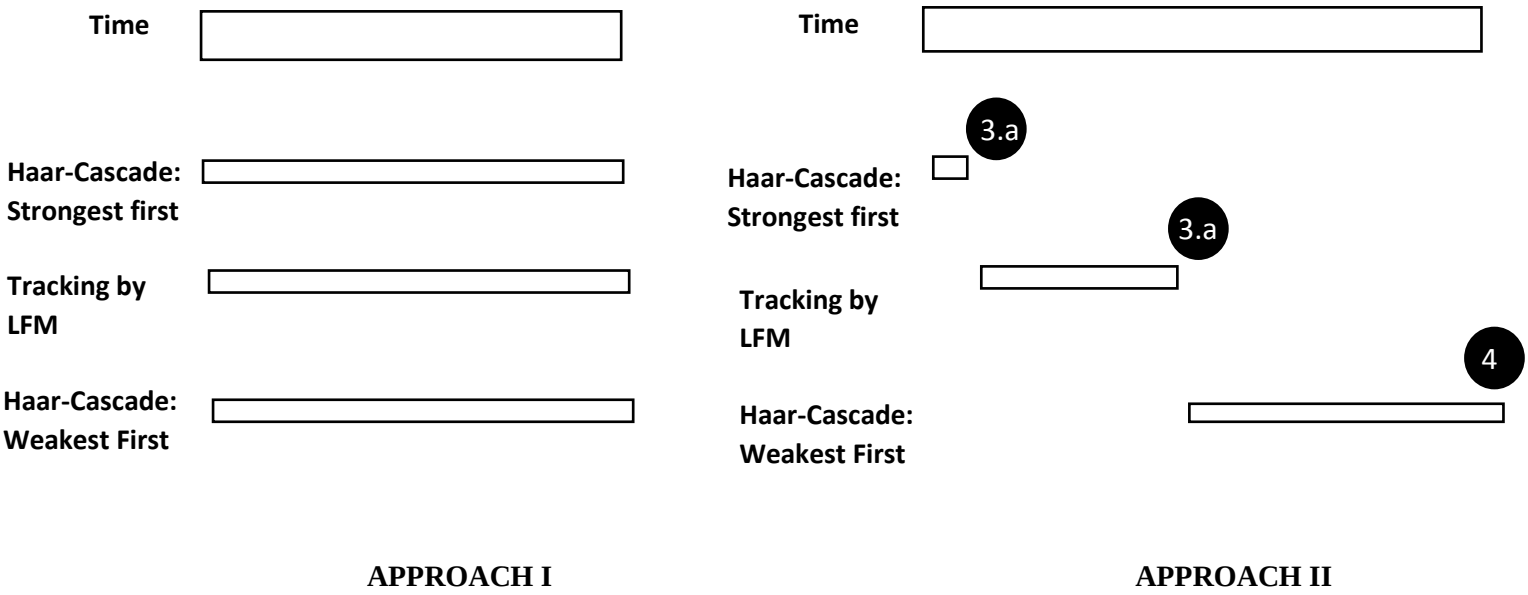
## 3.2 Step 2: Pre-processing

This step is a linear time algorithm, courtesy to Voila [1], and is a way to generate binary map for the video frame. The pre-processing phase gives output as binary image map which has greyscale pixels only. And the binary map for the image is constructed as the method proposed by Voila and Jones [1], called the "Integral Image". So, our coloured image frames are converted into a **pre-processed greyscale image representation** [27]that is most suitable for the HAAR-CASCADE "detection algorithm" as well as an image representation on which the LOCAL FEATURE MATCHING "tracking" algorithm is very fast to work. This represents coloured images in a form from which *feature extraction and detection is very easy and fast*. The only thing to keep in mind while implementing this pre-processing format is the features are easy to detect and match only when we are analysing in rectangular areas. The "Integral Image" encoding is formed in such a way that it supports rectangular region of interests. And both our main algorithms, i.e., HAAR CASCADE for "detection" and LOCAL FEATURE MATCHING for "tracking" could be implemented to only look in areas that are rectangular in shape.

## 3.3 Step 3 and 4: DETECTION and TRACKING

The very first time only step 3 of the process flow is used and step 4 is not used, as they are parallel process [39], for the completion of the whole FLOW CHART processes it is sufficient. This is only for the very first time when the user face is detected for the first time. After the user face is detected once, it is registered that the user session has started. And after this registration, for the next iterations, STEP 3 as well as STEP 4, bot work in harmony and co-operation.

Step 3 and Step 4 are applied in parallel. This simply is the case because both works simultaneously and if any one of them finds the location of the eye in the pre-processed image-frame, both of them stops and the location of the eye is passed to the next frame. And these two steps are further divided into two branches, 3.a and 3.b, 4.a and 4.b. The two sub-branches of DETECTION BRANCH are independent classifiers, though both are same but one starts from eliminating unwanted regions but the other vigorously looks for interested region. The two sub-branches of TRACKING BRANCH are inter-dependent on each other and could be termed as one search activity combined. Thus, there are in parallel three searches going on, two by the HAAR CASCADE DETECTERS and one by LOCAL FEATURE MATCHING TRACKER. The three are in parallel because if any one of them finds the location of eye, the work is done. As making all these three runs in absolute parallelism is a waste of resources and is not efficient for our use case. Thus, we cascade these three search activities in a manner that on an average takes least time to find the eye region.




**APPROACH I** **APPROACH II**

**Fig. 12: Two Approaches to cascade three search branches**

The Approach one is a simple approach to find the object having three parallel search branches. Though it can be implemented in computers having only one processing unit as how usually the parallel programs are implemented in serial processor by allocating time burst to them and servicing in round-robin fashion.

But all in all, the Approach one could be modified to run these three searches in an intelligent cascaded search policy. In this we not only convert three parallel lines to one serial execution line but also the search policy is the fastest. So, less computations and greater speed. We have made trade-offs to achieve this effect in multiple places in this report. This aim is central to developing any real-time system and is used here too. So, the second Approach says that we allow the HAAR-CASCADE which used its strongest detector in the region where the eyes were in the previous frame. Then it checks in neighbouring areas with less vigorous detectors. And it is allowed to run for just that amount of time that the AREA where FACE WASIN PREVIOUS FRAME as well as SOME SURROUNDING AREAS are searched and then this search is stopped and switched to next search branch. The idea of such a small time run for strongest detectors is that if the face in current frame is approximately near to its place in previous frame then, it is quickly detected and no need for further searches. Most of the time this is sufficient, as researches and practical observations suggest that most of the time the user face makes a very small movement and the times when the user movements are large are much less in number. Thus, effectively, most of the time the search is completed by this step. Now, the next branch of search is scheduled if this first search branch is not able to detect an eye in its due time. The next branch is the TRACKING ALGORITHM by LOCAL FEATURE MATHCING, this search method proceeds in such a way that the features of the eyes in the pre-processed image map is aimed to be tracked. TRACKING involves FEATURE MATCHING and PREDICTION PROCESS, so the search of eye feature is driven by a prediction of most suitable region through a voting mechanism and that region is computed by MATCHING THE FEATURES of EYES that are stored in the algorithm as well as a database. The FEATURE MATCHING computations for a candidate region gives the prediction of next region that is most likely to contain the FEATURES. So, this search branch too is given its due time to run and if the face with eyes is found in this stage, the overall search stops as it is successful. TRACKING guarantees to find the eye if the face movement is greater than that could be processed by previous stage but is not very large or very irregular. So,

regular motion of face is detected by this algorithm and if the search by this algorithm fails too. We are in the worst-case scenario, i.e., the face has made a movement that is large and irregular. So, instead of using the face location of earlier frame that we were doing in the previous two search strategies, we just start a new search of face in the frame using HAAR-CASCADE classifier. This HAAR-CASCADE classifier is the usual cascade detector and not the inverted HAAR-CASCADE that we used in the FIRST stage of this three-stage search cascading. Thus, we assume that our search should not be based on the location of the face in the previous frame, as we have strong evidence (failing of previous stages search) that face location in the current frame is better calculated as an independent search. These ideas are summarized in TABLE 3.

**TABLE 3: The work of each search branch in Fig. 10, Approach II**

| Search Branch | Face Movement |
|---|---|
| 3.a: HAAR CASCADE DETECTOR (strongest first) | Approximately face position is same, small movement |
| 4: TRACKING (LFM) | Smooth, Medium range Movement |
| 3.b: HAAR CASCADE DETECTOR (weakest first) | Irregular, Large Movement |

## 3.4 Step 5: Database Interactivity

The other specialty of Stages 3 and 4, is that they are constantly interaction with a feature database. This database is not the knowledge database that each algorithm has, either it be detection or be it tracking. This is the database that stores the feature information of the current user, who is being detected and analysed. ***Locality of reference is the concept*** used here. It is thought that the user generally remains the same over a large period. And also, the saved feature information of user is analysed to get the frequently visited user. The frequently visited user information makes the detection of the user face in the video frame very fast because if we have the features of a user then detection is just the process of matching. In normal face and eye detection systems it may not be valid assumption, like detecting human faces in traffic [41]. So, human face detection in traffic could not use this finite user feature data due to the number of users and the variety of users passing. But for our use it increases the speed of detection as well as decreases the computations done. And this trade-off is most vital in our project, and like majority of decisions in the project, this decision is also made to balance this trade-off.

## 3.5 Step 6 and 7: Projection Method

This phase is used to extract the state of the user eyes after the eyes are detected by the preceding algorithms. The state of the user eyes we detect in this report are only two:

- Gaze direction
- Open and Close

For the fast and efficient working of the Projection method [15] we use an intermediate step of PREPROCESSING, which is STEP 6 in the FLOW GRAPH for our system in Fig. 10. This pre-processing is different from the "INTEGRAL IMAGE" PREPROCESSING that we have done at step 2. This pre-processing is "MEDIAN FILTER" which smooth out the eye region image for the detection of black and white regions in the area of eye. The eye image's clarity is downgraded so the PROJECTION STEP is fast enough. The Pre-processing and Projection method is elaborated in the works of LIU and JIA [5].

### 3.6 Step 8:  Input Signal

This step is very simple in our use case of basic eye state and basic applications like eye open or closed, or gaze to screen or not. We detect the state of the user eyes and we map the state of the user eyes to the corresponding input signal to provide the input stream of the operating system. The operating system then gives the input data to the corresponding programs and user interactivity is achieved.

## 4.1 Conclusion:

The proposed system has to work in real time due to $1/5^{th}$ of a second continuous checking, tracking and detection. Experiments show that major time, the procedure just verifies that the position of eye is nearly static, thus saves a large number of computations. And when computations are needed, we do minimum to get accurate results, like Harr-Cascade which takes constant time for each classification and local Feature Matching which takes as minimum computation as possible in related images. Further "integral-image" pre-processing before any algorithm makes computations fast. The projection calculations are the only one to make detailed analysis but the trade-off we adopted for accuracy in gaze detection and time of computation is to have the computation only approximate, and move further by quick evaluation which should be supported by successive frames. It should work well in real systems as any wrong approximation will be detected by next frames and the user is not bothered by 0.20 second lag in correction. So, averaging over most of the personal computers in the market, the system would take only **1/10** of the **CPU resource** in worst case, but could be easily optimized below it. So, good implementations could achieve 7-8.5% of CPU resources, which though is costly for an input interface. The accuracy of the model combining the **accuracy** of all the component algorithms in a realistic manner and through limited simulation tests is touching **90%** value. Though the more correct values could be stated only after some more use cases and only after considering it for general purpose uses. But considering the fact that this model is still in its infancy, these values of metrics are considerably good and paves many ways to future optimizations.

## 4.2 Future Scope:

1. Website Usability: If the technology would be up to the level to precisely pin point the location of gaze of users, it would provide much use in knowing exact patterns of user INTEREST and VIEW. The tracking and analysing the eye gaze of users can help to get great and accurate feedback on the usability of the site, the relativity of the content and the interest of the users. This has the potential in itself to revolutionize the web development industry. The hotspots of user interest could be identified by the duration and intensity of the gaze and based on that data websites could be improved for both user and the company. User gets benefitted by the increase in the contents of his preference and interest and the company, has more attention of the users and has a website that is much more user oriented.

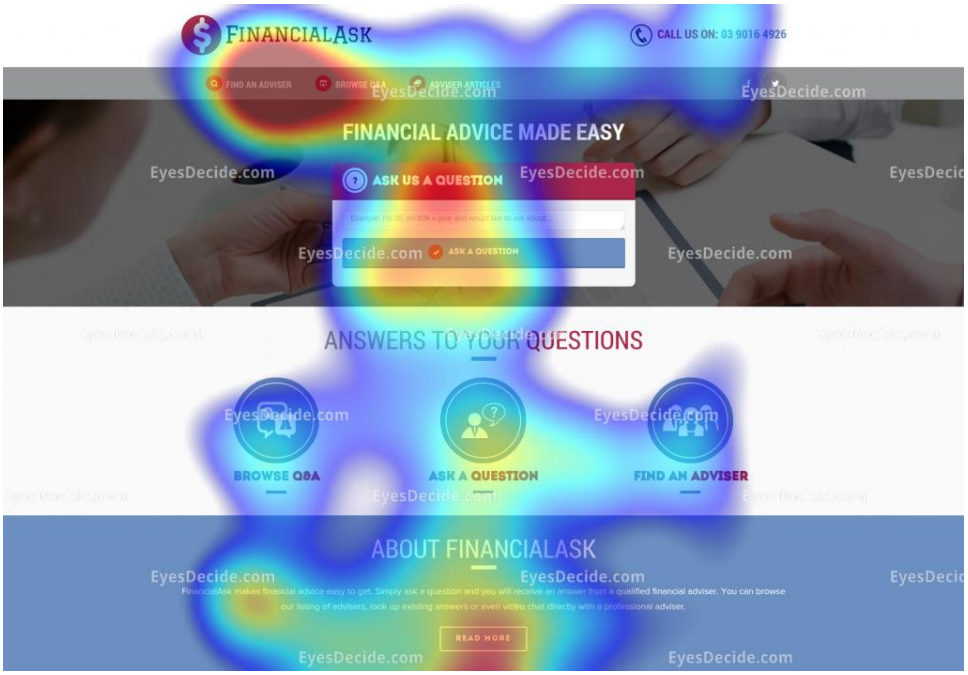



**Fig. 13: Eye Gaze heat map on a website**

2. AD Testing: Ads are annoying when they are irrelevant, have unorganized target and are random. But Ads are present in an extremely great magnitude and their presence is continuously becoming increasingly massive. Why? Because advertising online is easy and the amount of audience is exponential. The economy of many tech giants is standing on AD publishing. So, two things are clear, firstly, Ads are very powerful and have greater potential if they are oriented in the right path, secondly, the prime disadvantage of this whole Ads machinery is that is becomes evil if the AD matchmaking is inefficient. So, the solution is staring us in the eye (no pun intended), presenting Ads to correct target, by analysing the behaviour of the target, the kinds of contents he is spending more duration looking at, the kind of subjects he is searching for, his interest catching checkpoints, and then categorically present the correct Ads to the correct person. There can be many other ways in which it can be explored like what placement of Ads are more appealing, which are on the annoying side, how to capture the attention of the viewer, etc.


**Fig.14: Feedback on a new feature of website through auto gaze response tracker (**photo courtesy: xLabs Pty Ltd(AU), https://xlabsgaze.com)

In the pic: We are testing the user eye gaze response when one feature in the website is just substituted by another. And the User Eye heatmap easily and definitely points out massive change in behavioural patterns.

3. More natural mouse cursor flow and introduction of new input modes that are much more natural, convenient and easy for the user. The argument is simple, you cannot just carry along those modes of input that were started in 80s, when the terrain of computer-user interaction has changed so much that instead of "A discrete mechanical entity, having keys and plugs" to "A natural integrated device that can learn from user in so many new ways and interact much more SEAMLESSLY". We surely need new input modes that can "Adapt to the natural flow of humans" and complement as well as supplement it. We need systems intelligent enough to predict our intension by our patterns and take "Much less Explicit command and detect our orders in subtility". New inputs parameters like pupil diameter could tell attention level and thus adaptive GUI could be introduced.
4. Minimize the battery and resource use by this compute intensive process and one that always needs camera activity.
5. **Combining Eye analysis with hand gestures detection:**
   Various hand gestures could communicate additional signals and the interaction would be fluent, intuitive and more natural.

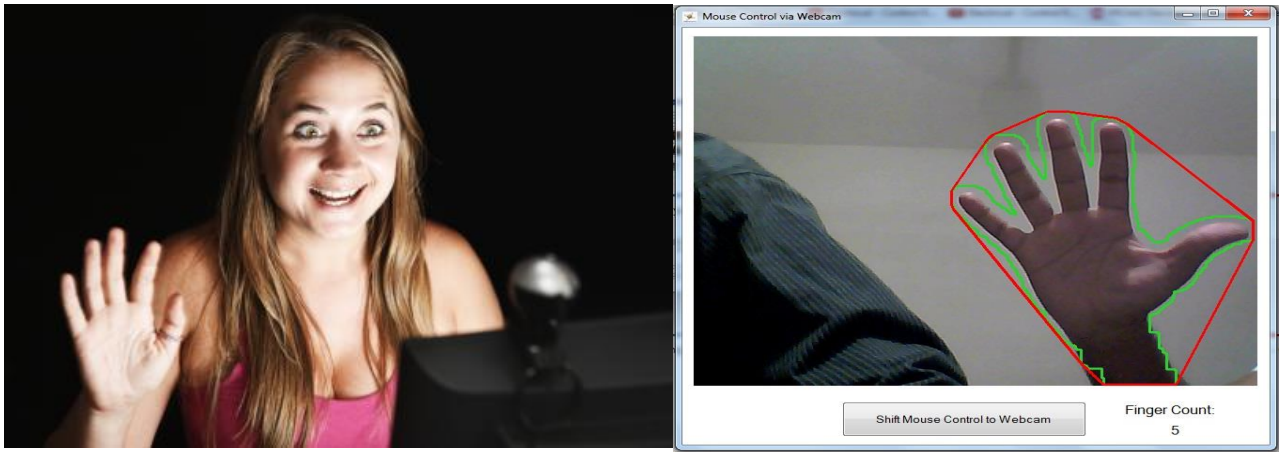



**Fig. 15: Using HAND GESTURES along with EYE INPUT INTERFACE SYSTEM**

6. **Combining Eye analysis with lip tracking:**
   Eyes and lips could convey a large amount of information which presently we do not analyse and if analysis could be extended to lips too, then the input to computer could revolutionize. Our access to computers would be much more fast, convenient and comfortable. The speed of input would increase drastically.

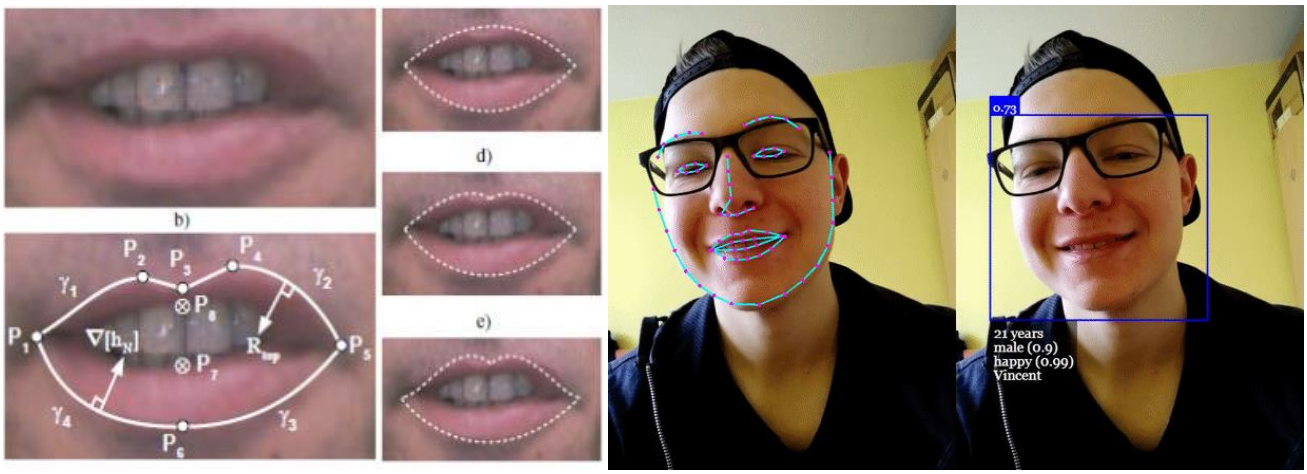



**Fig.16: Using LIP MOVEMENTS along with EYE INPUT INTERFACE SYSTEM**