ISSN 2229-5518

Iris Movement and Eye Gaze Tracking

Villuri Gnaneswar

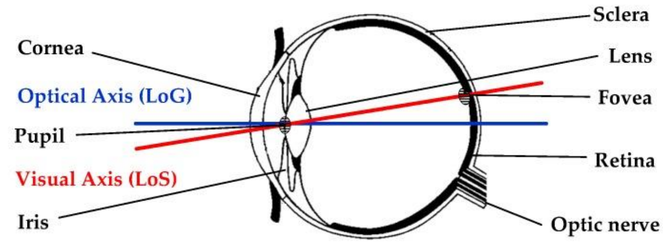
Abstract- Iris Movement and gaze tracking has been an active research field in the past years as it adds convenience to a variety of applications. It is considered a significant untraditional method of human-computer interaction. The goal of this paper is to present a study on the existing literature on Iris Movement and Gaze Tracking and Develop an Efficient Technique that can revolutionize the field of Computer Vision. With the uptrend of systems based on eye Tracking in many different areas of life in recent years, this subject has gained much more attention by in the academic and industrial area. A Motion analysis method is developed to track and detect Iris movement and gaze Tracking. They serve a wide range of severely disabled people who are left with minimal motor abilities. In this work, we want to put some light on the new system in which Using the eye as an interface to communicate with a system for people that are severely paralyzed or affected by diseases in which they are unable to move or control most of their body parts except for their eyes. This paper gives an overview of different techniques and describes the best possible methods of Iris movement and Gaze Tracking techniques. The main purpose of the system is the motion analysis method and finding Horizontal Ratio to Find the Direction of Vision i.e., Left, Right or Centre.

Keywords - Iris Movement, Gaze Tracking, Iris Tracking Applications, Computer Vision, Motion analysis, Horizontal Ratio.

I. INTRODUCTION

Eyes and their movements are important in expressing a person's desires, needs and emotional states. The significance of eye movements with regards to the perception of and attention to the visual world is certainly acknowledged since it is the means by which the information needed to identify the characteristics of the visual world is gathered for processing in the human brain. Hence, robust Iris Tracking and Gaze Tracking are considered to play a crucial role in the development of human-computer interaction, creating attentive user interfaces and analysing human affective states. Iris tracking is widely investigated as alternative interface methods. They are considered to be easier to use than other methods such as voice recognition or EEG/ECG signals. They also have achieved higher accuracy and performance. In addition, using eye tracking as alternative interface, control or communication methods is beneficial for a wide range of severely disabled people who are left with minimal ability to perform voluntary motion. There are many approaches introduced in literature focusing on eye tracking. They can be used as a base to develop an Iris tracking system which achieves the highest accuracy, best performance and lowest cost. There are many proposed approaches. Some approaches may be implemented using low computational hardware such as a microcontroller due to the simplicity of the used algorithm.

The typical eye structure used in Iris movement and eye gaze tracking applications is demonstrated in Figure 1.[34]

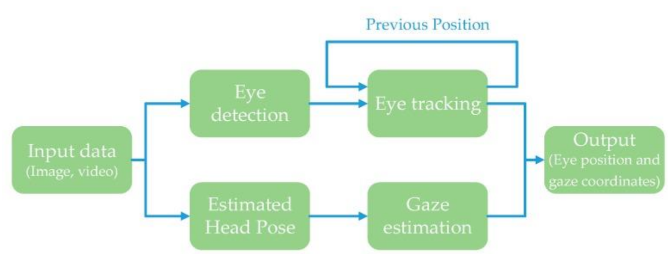


1.Structure of human eye.

During the last two to three decades, a revolutionary development was observed in eye tracking due to introduction of artificial intelligence techniques and portable electronics and head-mounted eye trackers.

Eye tracking and gaze estimation are essentially two areas of research. The process of eye tracking involves three main steps; viz., to discover the presence of eyes, a precise interpretation of eye positions, and frame to frame tracking of detected eyes.

A generic representation of Iris Movement and Gaze Tracking technique's is shown in Figure 2.[34]



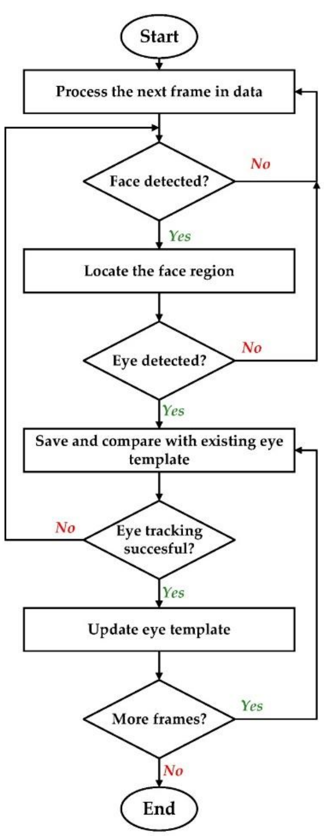
2.The process of tracking eye position and gaze coordinates.

The objective of gaze tracking process is to identify and track the observer's point of regard (PoR) or gaze direction. For this purpose, the important features of eye movements such as fixation, saccades, and smooth pursuit are utilized. Fixation represents the state when the observer's gaze rests for a minimum time (typically more than 80–100 ms) on a specific area within 2–5° of central vision. Saccades are quick movements of eyes that take place when visual attention transfers between two fixated areas, with the aim of an bringing area of interest within the narrow visual field.

This paper presents a survey of different Iris tracking techniques reported in the literature along with examples of various applications employing these technologies. The rest of the paper is outlined as follows. Different methods of Iris tracking and Gaze Tracking Techniques are investigated in Section 2. My Approach and Method for Iris Movement and Gaze Tracking of basic eye movement types is described in Section 3. Fields of applications for eye tracking are described briefly in Section 4. An Experimental result of existing Iris tracking Techniques and My approach to the Iris and Gaze Tracking is also described in Section 5. Finally, Section 6 draws the conclusions.

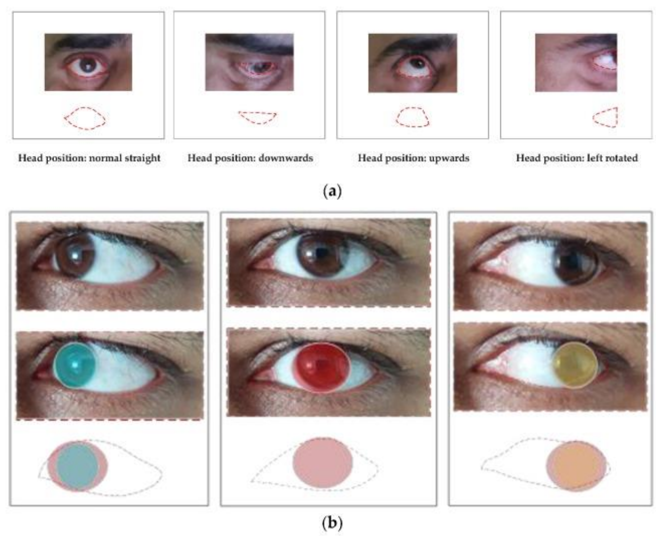
II. EYE TRACKING

The geometric and motion characteristics of the eyes are unique which makes gaze estimation and tracking important for many applications such as human attention analysis, human emotional state analysis, interactive user interfaces and human factors. There are many different approaches for implementing Iris Movement and Gaze Tracking systems. In General most of these approaches execute a recursive process similar to the flowchart shown in Figure 3.[34]



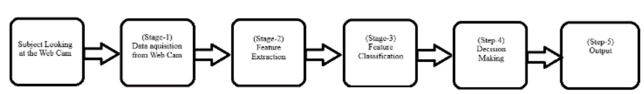
Many eye tracking methods were presented in the literature. However, the research is still on-going to find robust Iris Movement and Gaze Tracking methods to be used in a wide range of applications.

The first step in eye tracking is to detect the eyes. The Tracking of eyes in image or video data is based on eye models. An exemplar eye model should be sufficiently meaningful to accommodate the variability in eyes' dynamics and appearance while adequately constrained to be computationally efficient. Eye Tracking and tracking is an arduous job due to exceptional issues, such as degrees of eye openness; variability in size, head pose, and reflectivity; and occlusion of the eye by eyelids [3,25,26]. For instance, a small variation in viewing angle or head position causes significant changes in the eye appearance or gaze direction, as shown in Figure 2. The eye's appearance is also influenced by ethnicity of the subject, light conditions, texture, iris position within eye socket, and the eye status (open or closed). Eye detection methods are broadly categorized based on eyes' shape, features, and appearance, as explained below.



The appearances of eyes and eye parts change with head and eye movements. (**a**) Variability in eye appearance when eye position is fixed but head position varies. (**b**) Variability in gaze direction when head position is fixed but eyeball rotates.

The Stages of Visual data for most of the face detection algorithm's is shown below.



2.1 Sensor-based eye tracking (EOG)

Some eye tracking systems detect and analyse eye movements based on electric potentials measured with electrodes placed in the region around the eyes. This electric signal detected using two pairs of electrodes placed around one eye is known as electrooculogram (EOG). When the eyes are in their origin state, the electrodes measure a steady electric potential field. If the eyes move towards the periphery, the retina approaches one electrode and the cornea approaches the other. This changes the orientation of the dipole and results in a change in the measured EOG signal. Eye movement can be tracked by analysing the changes in the EOG signal.

2.2 Computer-vision-based eye tracking

Most eye tracking methods presented in the literature use Computer Vision based techniques. In these methods, a camera is set to focus on one or both eyes and record the Iris Movement. The main focus of this paper is on Computer Vision based Iris Movement and Gaze Tracking. There are two main areas investigated in the field of Computer Vision based eye tracking. The first area considered is Iris detection in the image, also known as eye localization. The second area is eye tracking, which is the process of eye gaze direction estimation. Based on the data obtained from processing and analysing the detected eye region, the direction of eye gaze can be estimated then it is either used directly in the application or tracked over subsequent video frames in the case of real-time eye tracking systems. Iris Movement and Gaze Tracking is still a challenging task, as there are many issues associated with such systems. These issues include degree of eye openness, variability in eye size, head pose, etc. Different applications that use eye tracking are affected by these issues at different levels. Several Computer-Vision based eye tracking approaches have been introduced.

2.2.1 Pattern recognition for eye tracking

Different pattern recognition techniques, such as template matching and classification, have proved effective in the field of eye tracking. Raudonis et al[4] used principal component analysis (PCA) to find the first six principal components of the eye image to reduce dimensionality problems, which arise when using all image pixels to compare images. Then, Artificial Neural Network (ANN) is used to classify the pupil position. The training data for ANN is gathered during calibration where the user is required to observe five points indicating five different pupil positions. The use of classification slows the system and hence it requires some enhancements to be applicable. In addition, the system is not considered a real time eye tracking system. The proposed algorithm was not tested on a known database which means the quality of the system might be affected by changes in lighting conditions, shadows, distance of the camera, the exact position in which the camera is mounted, etc. The algorithm requires processing which cannot be performed by low computational hardware such as a microcontroller.

2.2.2 Eye tracking based on corneal reflection points

Many Computer Vision based eye trackers use light reflection points on the cornea to estimate the Gaze direction. Figure 1 shows the corneal reflection points in an eye image [12]. Another name for eye images containing corneal reflection points is Purkinje Image. When using this approach, the vector between the centre of the pupil and the corneal reflections is used to compute the gaze direction. A simple calibration procedure of the individual is usually needed before using the eye tracker [2_12].

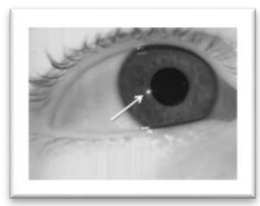


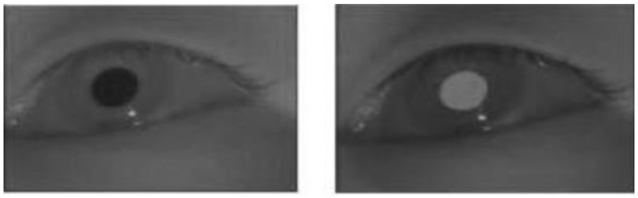
Figure1. Corneal reflection points

2.2.3 Eye tracking based on shape

Another approach for Iris Movement and Gaze Tracking is to find the location of the iris or the pupil based on their circular shape or using edge detection. Chen and Kubo [15] proposed a technique where a sequence of face detection and Gabor filters is used. The potential face regions in the image are detected based on skin colour. Then, the eye candidate region is determined automatically using the geometric structure of the face. Four Gabor filters with different directions ($0, \pi/4, \pi/2, 3\pi/4$) are applied to the eye candidate region. The pupil of the eye does not have directions and thus, it can be easily detected by combining the four responses of the four Gabor filters with a logical product. The system uses a camera which is not head mounted. The accuracy of the algorithm is not investigated and the required CPU time is not mentioned which does not make the algorithm preferable for real world applications compared to other algorithms.

2.2.4 Eye tracking using dark and bright pupil effect

There are two illumination methods used in the literature for pupil detection-the dark pupil and the bright pupil method. In the dark pupil method, the location of a black pupil is determined in the eye image captured by the camera. This causes some issues when the user has dark brown eyes because of the low contrast between the brown iris and the black pupil. The bright pupil method uses the reflection of infrared light from the retina which makes the pupil appear white in the eye image. Figure 4 shows the dark and bright pupil effect.



(a) (b)

Figure 4 (a) Dark and (b) Bright pupil effects.

2.2.5 Hybrid eye tracking techniques

A mix of different techniques can be used for eye tracking. Huang et al. [24] suggested an algorithm to detect eye pupil based on intensity, shape, and size. Special Infrared (IR) illumination is used and thus, eye pupils appear brighter than the rest of the face. The intensity of the eye pupil is used as the primary feature in pupil detection. However, some other bright objects might exist in the image. To separate the pupil from bright objects existing in the image, other pupil properties can be used, such as pupil size and shape. Support Vector Machine is used to locate the eye location from the detected candidates. The used hardware, including the IR LEDs and the IR camera, is not expensive. The algorithm has been used in a driver fatigue detection application. The algorithm can be considered a new beginning for real-time eye tracking systems if it is tested further with different test subjects and different classification functions in order to reach the most optimized eye algorithm. The required CPU time was not mentioned although it is important in driver fatigue detection applications as they are real-time applications. Using a corneal reflection and energy controlled iterative curve fitting method for efficient pupil detection was proposed by Li and Wee [12]. Ellipse fitting is needed to acquire the boundary of the pupil based on a learning algorithm developed to perform iterative ellipse fitting controlled by a gradient energy function. This method uses special hardware which has been implemented specifically for this algorithm. It has been used in a Field-of-View estimation application and can be used in other applications. Coetzer and Hancke [25] proposed a system for eye tracking which uses an IR camera and IR LEDs. It captures the bright and dark pupil images subsequently such that they are effectively the same image but each has been taken in different illumination conditions. Two groups of infrared LEDs are synchronized with the IR camera. The first is placed close to the camera's lens to obtain the bright pupil effect and the second about 19.5 cm away from the lens, to produce the dark pupil effect. The images are then subtracted from each other and a binary image is produced by thresholding the difference. This technique has been presented by Hutchinson [26]. The binary image contains white blobs that are mapped to the original dark pupil image. The sub-images that result from the mapping are potential eye candidates. The possible eye candidate sub-images are classified into either eyes or no eyes. Artificial neural networks (ANN), support vector machines (SVM) and adaptive boosting (AdaBoost) have been considered as classification techniques in this work. The system is ready for further improvements and enhancements. It was utilized in a driver fatigue monitoring system. It does not require calibration for each user because it uses a dataset for training and feature extraction. The best features to be used and flexible hardware implementation can be investigated further in order to make the algorithm a part of a bigger eye tracking system or eye location classification system. The images used in experiments were eye images after the background and noise were eliminated. This reduces the expectations of this algorithm performance in real applications.

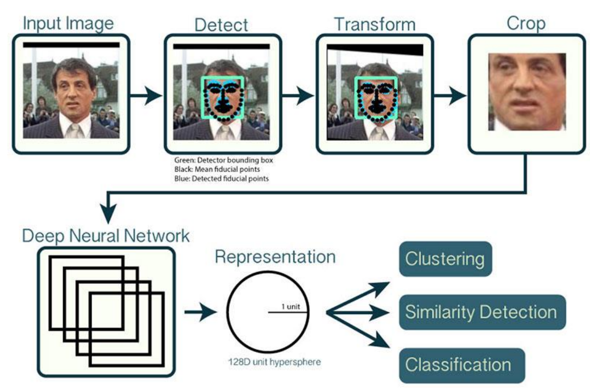
III. APPROACH AND METHOD

Subject's face containing eyes and portion of surrounding regions is captured with a high-resolution, high-frame-rate camera. Subject's face containing eyes and portion of surrounding regions is captured with a high-resolution, high-frame-rate camera or WebCam.
From the Captured Face, I used OpenCV and DLib for the Face Recognition.

OpenCV:

OpenCV is a library of programming functions mainly aimed at real-time computer vision.

Working:



DLib:

This is an advanced machine learning library that was created to solve complex real-world problems. This library has been created using the C++ programming language and it works with C/C++, Python, and Java.

Working:

Our face has several features that can be identified, like our eyes, mouth, nose, etc. When we use DLib algorithms to

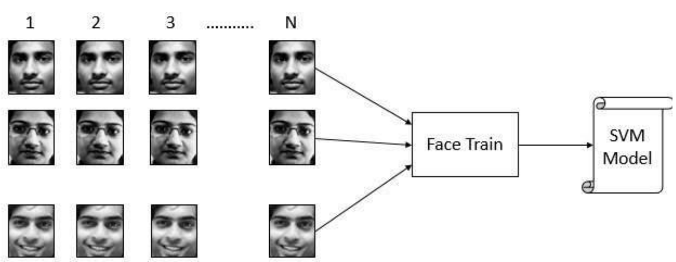
detect these features we actually get a map of points that surround each feature.
This map is composed of 67 points (called landmark points) and can identify the following features:
Jaw Points = 0–16
Right Brow Points = 17–21

- Left Brow Points = 22–26

Nose Points = 27–35
Right Eye Points = 36–41
Left Eye Points = 42–47
Mouth Points = 48–60
Lips Points = 61–67
Now, For extracting the eye region I used the points [36-41] (For Right Eye), [42-47] (For Left Eye). I used the DLib function get_frontal_face_detector() to retrieve the face information. This detector is based on Histogram of Oriented Gradients (HOG) and Linear Support Vector Machine (SVM).[35][36]

What are Support Vector Machines (SVMs)?

Support vector machines (SVMs) are supervised machine learning models that divide and classify data.

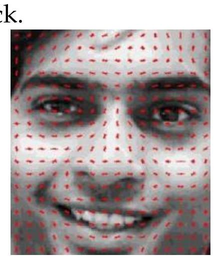


Block diagram of face training

SVMs are widely used for applications such as face detection, classification of images, handwriting recognition, etc. An SVM model can be considered as a point space wherein multiple classes are isolated using hyperplanes.

What is a Histogram of Oriented Gradients (HOG)?

A HOG is a feature descriptor generally used for object detection. HOGs are widely known for their use in pedestrian detection. A HOG relies on the property of objects within an image to possess the distribution of intensity gradients or edge directions. Gradients are calculated within an image per block. A block is considered as a pixel grid in which gradients are constituted from the magnitude and direction of change in the intensities of the pixel within the block.



HOG features sample face

In the current example, all the face sample images of a person are fed to the feature descriptor extraction algorithm; i.e., a HOG. The descriptors are gradient vectors generated per pixel of the image. The gradient for each pixel consists of magnitude and direction, calculated using the following formulae:

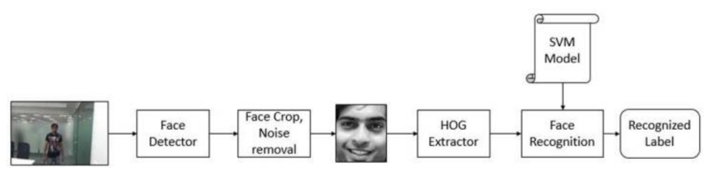
$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

In the current example, Gx and Gy are respectively the horizontal and vertical components of the change in the pixel intensity. A window size of 128 x 144 is used for face images since it matches the general aspect ratio of human faces. The descriptors are calculated over blocks of pixels with 8 x 8 dimensions. These descriptor values for each pixel over 8 x 8 block are quantized into 9 bins, where each bin represents a directional angle of gradient and value in that bin, which is the summation of the magnitudes of all pixels with the same angle. Further, the histogram is then normalized over a 16 x 16 block size, which means four blocks of 8 x 8 are normalized together to minimize light conditions. This mechanism mitigates the accuracy drop due to a change in light. The SVM model is trained using a number of HOG vectors for multiple faces.

Face Recognition

The recognition of a face in a video sequence is split into three primary tasks: Face Detection, Face Prediction, and Face Tracking. The tasks performed in the Face Capture program are performed during face recognition as well. To recognize the face obtained, a vector of HOG features of the face is extracted. This vector is then used in the SVM model to determine a matching score for the input vector with each of the labels. The SVM returns the label with the maximum score, which represents the confidence to the closest match within the trained face data.



Block diagram of the face recognition process

The task of calculating matching scores is exceptionally heavy to compute. Hence, once detected and identified, the labeled face in an image needs to be tracked to reduce the computation in future frames until the face eventually disappears from the video. Of all the available trackers, the Camshift tracking algorithm is used since it produces the best results with faces.

Since the above function get_frontal_face_detector() works only with grayscale images I have used OpenCV to convert it to a grayscale image. I then calibrated the pupil detection algorithm by finding the best binarization threshold value for the person and the webcam. Then I got the Left and

Right pupil coordinates which I used for calculating the Horizontal and Vertical Ratio by a simple formula.

-Horizontal ratio indicates the Horizontal direction of the Gaze

pupil_left = eye_left.pupil.x / (eye_left.center[0] * 2 - 10)

pupil_right = eye_right.pupil.x / (eye_right.center[0] * 2 - 10)

Horizontal_ratio = (pupil_left + pupil_right) / 2

-Vertical ratio indicates the Vertical direction of the Gaze

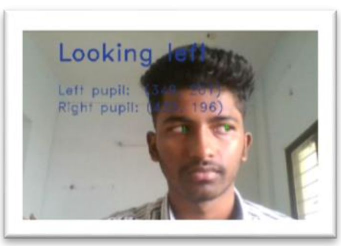
pupil_left = eye_left.pupil.y / (eye_left.center[1] * 2 - 10)

pupil_right = eye_right.pupil.y / (eye_right.center[1] * 2 - 10)

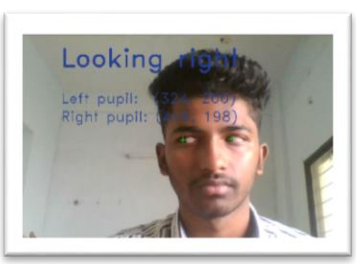
Vertical_ratio = (pupil_left + pupil_right) / 2

By comparing the Horizontal ratio we can tell whether the person is looking Left, Right or Centre.

Looking Left:
The function gaze.is_left() returns True if the subject is looking left (horizontal_ratio() >= 0.65).



Looking Right:
The function gaze.is_right() returns True if the subject is looking right (horizontal_ratio() <= 0.40).



LookingCentre:
When both of the above function returns false then the function gaze.is_centre() returns True.

IV. EYE TRACKING APPLICATIONS

The field of research in eye tracking has been very active due to the significant number of applications that can benefit from robust eye tracking methods. Many application areas employ eye tracking techniques. In the following, some of these areas are described.

A. Eye Control for Accessibility and Assistive Technology

People who have lost the control over all their muscles and are no longer able to perform voluntary movements as a result of diseases or accidents can benefit widely from eye tracking systems to interact and communicate with the world in daily life. Eye tracker systems provide many options for these individuals such as an eye-typing interface that could have text-to-speech output. They also enable eye-control, including directing electric wheelchairs or switching on the TV or other devices.Human computer interaction with graphical user interface actions or events may be classified into two main categories [27]:
Pointing: moving the pointer over an object on the screen such as text or icon on the screen.
Selection: action interpreted as a selection of the object pointed to.Kocejko et al. [16] introduced the “Eye Mouse” which people with severe disabilities can use. The mouse cursor position is determined based on the acquired information about the eye position towards the screen which provides the ability to operate a personal computer by people with severe disabilities.Raudonis et al. [4] dedicated their eye tracking system to the assistance of people with disabilities. The system was used with three on-line applications. The first controls a mobile robot in a maze. The second application was “Eye Writer” which is a text-writing program. A computer game was the third application.Fu and Yang [10] suggested employing information obtained from tracking the eye gaze to control a display based on video. Eye gaze is estimated and the display is controlled accordingly.Lupu et al. [27] proposed “Asistsys” which is a communication system for patients suffering from neuro-locomotor disabilities. This system assists patients in expressing their needs or desires.

B. E-Learning

E-learning systems are computer-based teaching systems and are now very common. However, despite the fact that users are usually accustomed to machine interactions, the learning experience can be quite different. In particular, the "emotional" part is significant in the interaction between teacher and learner and it is missing in Computer based learning processes.
Porta et al. [29] sought to build an e-learning platform which determines whether a student is having difficulty understanding some content or is tired or stressed based on the interpreted eye behaviour.

C. Car Assistant Systems

Research is done on applying eye tracking methods in the vehicle industry with the aim of developing monitoring and assisting systems used in cars. For example, an eye tracker could be used in cars to warn drivers when they start getting tired or fall asleep while driving.
Driver fatigue can be detected by analysing blink threshold, eye state (open/closed) and for how long the driver's gaze stays in the same direction. Many eye tracking methods were used in this area of application [24], [25], [30].

D. Iris Recognition

Iris recognition is being widely used for biometric authentication. Iris localization is an important and critical step upon which the performance of an iris recognition system depends [18],[19].

E. Field of View Estimation

Another interesting application of eye tracking system's is that these systems can serve as an effective tool in optometry to assist in identifying the visual field of any individual, especially identifying blind spots of vision.
Li and Wee [12] used eye tracking to estimate the field of view to be used for augmented video/image/graphics display.

V. Experimental result of existing Iris tracking Techniques & My approach to the Iris and Gaze Tracking

The performance of eye tracking and head movement detection systems is evaluated in terms of accuracy and required CPU processing time.

Method	Detection Accuracy (%)	Angle Accuracy (degree)	CPU time (ms)
Eye tracking using Pattern Recognition			
Raudonis *et al.* [4]	100%	N/A	N/A
Kuo *et al.* [6]	90%	N/A	N/A
Yuan and Kebin [9]	N/A	1	N/A
Lui and Lui [7]	94.1%	N/A	N/A
Khairosfaizal and Nor'aini [17]	86%	N/A	N/A
Hotrakool *et al.* [8]	100%	N/A	12.92
Shape-based eye tracking			
Yang *et al.* [13]	N/A	0.5	N/A
Yang *et al.* [14]	N/A	Horizontal: 0.327 Vertical: 0.3	N/A
Mehrubeoglu *et al.* [11]	90%	N/A	49.7
Eye tracking using hybrid techniques			
Li and Wee [12]	N/A	0.5	N/A
Huang *et al.* [24]	95.63%	N/A	N/A
Coetzer and Hancke [25]	98.1%	N/A	N/A

My Approach Results:

Method	Detection Accuracy (%)	Angle Accuracy (degree)	CPU time (ms)
OpenCV and DLib	92%	N/A	28

VI.CONCLUSIONS

Iris Movement and Gaze tracking is considered effective and reliable human-computer interaction and communication alternative methods. Hence, they have been the subject of many Research works. Many approaches for implementing these technologies have been reported in the literature. This paper investigated existing methods of eye tracking and an Efficient method of using OpenCV and DLib which had resulted in better performance and accuracy than the previous methods. Many applications can benefit from utilizing effective eye tracking methods. However, the research is still facing challenges in presenting robust methods which can be used in applications to detect and track eye accurately. Iris Movement and Gaze tracking methods rarely investigate the required CPU time. However, real-time application requires investigating and optimizing the performance requirements. In addition, most studies do not test eye tracking using a known image database that contains variant images of different subjects in different conditions such as lighting conditions, noise, distances, etc. This makes the reported accuracy of a method less reliable because it may be affected by different test conditions.

REFERENCES

[1] T. Ohno, ``One-point calibration gaze tracking method,'' in Proc. Symp. Eye Track. Res. Appl., 2006, p. 34.

[2] D. W. Hansen and Q. Ji, ``In the eye of the beholder: A survey of models for eyes and gaze,'' IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 3, pp. 478_500, Mar. 2010.

[3] J. Gips, P. DiMattia, F. X. Curran, and P. Olivieri, ``Using EagleEyes_An Electrodes based device for controlling the computer with your eyes to help people with special needs,'' in Proc. Interdisciplinary Aspects Comput. Help. People Special Needs, 1996, pp. 77_84.

[4] V. Raudonis, R. Simutis, and G. Narvydas, ``Discrete eye tracking for medical applications,'' in Proc. 2nd ISABEL, 2009, pp. 1_6.

[5] J. Tang and J. Zhang, ``Eye tracking based on grey prediction,'' in Proc. 1st Int. Workshop Educ. Technol. Comput. Sci., 2009, pp. 861_864.

[6] Y. Kuo, J. Lee, and S. Kao, ``Eye tracking in visible environment,'' in Proc. 5th Int. Conf. IIH-MSP, 2009, pp. 114_117.

[7] H. Liu and Q. Liu, ``Robust real-time eye detection and tracking for rotated facial images under complex conditions,'' in Proc. 6th ICNC, vol. 4. 2010, vpp. 2028_2034.

[8] W. Hotrakool, P. Siritanawan, and T. Kondo, ``A real-time eye-tracking method using time-varying gradient orientation patterns,'' in Proc. Int. Conf. Electr. Eng., Electron. Comput. Telecommun. Inf. Technol., 2010, pp. 492_496.

[9] Z. Yuan and J. Kebin, ``A local and scale integrated feature descriptor in eye-gaze tracking,'' in Proc. 4th Int. CISP, vol. 1. 2011, pp. 465_468.

[10] B. Fu and R. Yang, ``Display control based on eye gaze estimation,'' in Proc. 4th Int. CISP, vol. 1. 2011, pp. 399_403.

[11] M. Mehrubeoglu, L. M. Pham, H. T. Le, R. Muddu, and D. Ryu, ``Realtime eye tracking using a smart camera,'' in Proc. AIPR Workshop, 2011, pp. 1_7.

[12] X. Li andW. G.Wee, ``An ef_cient method for eye tracking and eye-gazed FOV estimation,'' in Proc. 16th IEEE Int. Conf. Image Process., Nov. 2009, pp. 2597_2600.

[13] C. Yang, J. Sun, J. Liu, X. Yang, D. Wang, and W. Liu, ``A gray difference-based pre-processing for gaze tracking,'' in Proc. IEEE 10th ICSP, Oct. 2010, pp. 1293_1296.

[14] X.Yang, J. Sun, J. Liu, J. Chu,W. Liu, andY. Gao, ``Agaze tracking scheme for eye-based intelligent control,'' in Proc. 8th WCICA, 2010, pp. 50_55.

[15] Y. W. Chen and K. Kubo, ``A robust eye detection and tracking technique using gabor _lters,'' in Proc. 3rd Int. Conf. IEEE Intell. Inf. Hiding Multi-Media Signal Process., vol. 1. Nov. 2007, pp. 109_112.

[16] T. Kocejko, A. Bujnowski, and J. Wtorek, ``Eye mouse for disabled,'' in Proc. IEEE Conf. Human Syst. Interact., May 2008, pp. 199_202.

[17] W. Khairosfaizal and A. Nor'aini, ``Eye detection in facial images using circular Hough transform,'' in Proc. 5th CSPA, Mar. 2009, pp. 238_242.

[18] A. Pranith and C. R. Srikanth, ``Iris recognition using corner detection,'' in Proc. 2nd ICISE, 2010, pp. 2151_2154.

[19] R. M. Sundaram, B. C. Dhara, and B. Chanda, ``A fast method for iris Localization,'' in Proc. 2nd Int. Conf. EAIT, 2011, pp. 89_92.

[20] N. Alioua, A. Amine, M. Rziza, and D. Aboutajdine, ``Eye state analysis using iris detection based on circular hough transform,'' in Proc. ICMCS, Apr. 2011, pp. 1_5.

[21] D. H. Yoo, J. H. Kim, B. R. Lee, and M. J. Chung, ``Non-contact eye gaze tracking system by mapping of corneal re_ections,'' in Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit., May 2002, pp. 94_99.

[22] D. H. Yoo and M. J. Chung, ``A novel nonintrusive eye gaze estimation using cross-ratio under large head motion,'' Special Issue on Eye Detection and Tracking, Computer Vision and Image Understanding, vol. 98, no. 1, pp. 25_51, 2005.

[23] Z. Zhu and Q. Ji, ``Novel eye gaze tracking techniques under natural head movement,'' IEEE Trans. Biomed. Eng., vol. 54, no. 12, pp. 2246_2260, Dec. 2007.

[24] H. Huang, Y. S. Zhou, F. Zhang, and F. C. Liu, ``An optimized eye locating and tracking system for driver fatigue monitoring,'' in Proc. ICWAPR, vol. 3. 2007, pp. 1144_1149.

[25] R. C. Coetzer and G. P. Hancke, ``Eye detection for a real-time vehicle driver fatigue monitoring system,'' in Proc. IEEE Intell. Veh. Symp., Jun. 2011, pp. 66_71.

[26] R. Lupu, R. Bozomitu, F. Ungureanu, and V. Cehan, ``Eye tracking based communication system for patient with major neuro-locomotor disabilities,'' in Proc. IEEE 15th ICSTCC, Oct. 2011, pp. 1_5.

[27] C. Calvi, M. Porta, and D. Sacchi, ``e5Learning, an e-learning environment based on eye tracking,'' in Proc. 8th IEEE ICALT, Jul. 2008, pp. 376_380.

[28] M. Porta, S. Ricotti, and C. J. Pere, ``Emotional e-learning through eye tracking,'' in Proc. IEEE Global Eng. Educ. Conf., Apr. 2012, pp. 1_6.

[29] M. S. Devi and P. R. Bajaj, ``Driver fatigue detection based on eye tracking,'' in Proc. 1st Int. Conf. Emerg. Trends Eng. Technol., 2008, pp. 649_652.

[30] A.T. Duchowski, Eye Tracking Methodology: Theory and Practice. Springer-Verlag New York, Inc., 2007.

[31] D.L. Donoho, ―De-Noising by Soft-Thresholding,‖ IEEE Trans. Information Theory, vol. 41, no. 3, pp. 613-627, May 1995.

[32] A Comprehensive Guide on How to detect faces with python by LIVE CODE STREAM (ONLINE).

[33] Antoine Lamé, Gaze Tracking.

[34] Muhammad Qasim Khan and Sukhan Lee, Gaze and Eye Tracking Techniques and applications in ADAS (ONLINE)

[35] Rajeev Thaware, Real-Time Face Detection and Recognition with SVM and HOG Features

[36] Arun ponnuswamy, CNN based face detector from dlib