Policy Research Journal
ISSN (E): 3006-7030 ISSN (P) : 3006-7022 Volume 3, Issue 3, 2025

BEST LOW-COST METHOD FOR REAL TIME DETECTION OF THE EYE AND GAZE TRACKING

Sidra Gulzar[*1], Muhammad Nadeem Gul[2], Ayesha Mumtaz[3], Muhammad Arif[4]

[*1,2,3,4]*Department of Computer Science and IT, Superior University Lahore, 54000, Pakistan*

[*1]sidragulzar25@gmail.com, [2]prof.nadeemgul@gmail.com, [3]ayeshanouman031@gmail.com, [4]md.arif@superior.edu.pk

DOI: https://doi.org/10.5281/zenodo.15100015

Keywords
computer vision; human computer interaction; deep learning; eye gaze tracking; iris landmarks etc.

Article History
Received on 21 February 2025
Accepted on 21 March 2025
Published on 28 March 2025

Corresponding Author: *

Abstract
One important field of computer vision research is the study of gaze tracking. It emphasizes practical usage and the interaction between people and technology. The demand for inexpensive techniques has increased recently due to new eye-tracking applications. An essential component in tracking the gaze's direction is the ocular area. This research proposes a number of novel eye-tracking techniques that use techniques to identify the eye region and the direction of gaze. Eye-tracking may be done using unmodified cameras without the requirement for specific hardware or software. The eye area was identified using either the Haar cascade approach or facial landmarks. Furthermore, the orientation of the eye was ascertained using the engineering technique, which relies on distances identifying the iris area, and the direct method, which is based on the convolutional neural network model. Two engineering approaches are used in the paper: the iris region is divided into five parts, with the blackest region indicating the look direction, and the gaze direction junction point is identified by drawing perpendicular lines on the iris region. While engineering techniques increase their efficacy in broad mobility, the suggested network model has demonstrated efficacy in predicting the eye's gaze direction in restricted mobility.

INTRODUCTION

Human-computer interaction has increased recently, and eye tracking and detection is one of the most crucial aspects of this connection. Numerous applications, such as virtual reality, augmented reality, consumer behavior identification, computer control, identification, medical treatment systems, and security programs, have benefited from this research [1]. It is the eyes that are the most conspicuous and consistent of all the facial characteristics [2]. Eye recognition still faces a number of challenges, such as differences in eye appearance, occlusion, and background noise. Iris color, size, and shape are key aspects of eye appearance. The illumination of the camera causes occlusion, which is a white reflection in pictures. The look of the eyes can also be affected by outside variables, such as lighting and eyewear [3]. The eyes are one of the many nonverbal communication techniques that may be especially helpful. The position of a person's glance, the amount of time spent gazing, and other comparable parameters are only a few of the more specific information about user behaviors that eye gaze monitoring may be able to supply [4].

One method is called eye gaze tracking (EGT), which involves tracking eye activity. This tool is frequently used to assess an individual's attentional concentration. Additionally, eye gaze analysis may help us understand human behavior, attention, and a

number of other cognitive processes [5]. Additionally, EGT might be used as a user interface for people with impairments. People may be able to operate a computer gadget with just their eyes. The development of EGT technology as a replacement for several input devices, including the mouse and touch screen, has resulted in the replacement of these devices [6]. One of the biggest challenges for scholars interested in computer vision is eye tracking research. This is a result of the eye detection method's requirement for the expensive installation of a high-quality camera in specialist equipment. Because the pupil cannot always be easily located while using unmodified cameras, they are more difficult to operate than infrared cameras. Unadjusted webcams can only function in visible light, which is the first issue. The second issue is that their lenses frequently have a wide-angle focal length and little zoom power. Because of this, the quality of the eyes' picture is poor and much influenced by the lighting; hence, it may be challenging to gauge someone's gaze depending on their head posture and the lighting [7].

While remote tracking uses a computer monitor or screen to identify the gaze's position, eye gaze tracking uses cameras and infrared sensors to capture eye movements. Tobii EyeX is a commercially available EGT device, however it is fairly expensive and needs certain equipment to work well. Due of their potential high cost of up to $25,000, these gadgets are not affordable for most individuals [8]. There are several shortcomings of the existing gaze-tracking systems, such as a high degree of setup complexity, costly components, stringent configuration, and laborious data collection methods [9]. Because of this, current research has concentrated on creating techniques for measuring gaze based on appearance. Based on images captured by traditional cameras, these algorithms can predict eye gaze. This method is not only more accessible but also less costly for the purpose of recording eye movements [10]. Appearance-based eye tracking requires specific traits to be present for gaze estimation. These variables include the user's distance from the screen or camera, head position, and camera settings.

In contrast, much of the existing research ignores information about the distance between the user and the camera and gathers data in a controlled setting with a fixed head angle [11–12]. Because of this, the gaze estimate's conclusions are imprecise, which limits its use in applications requiring precise location [13–15]. Image/video oculography, electro-oculography, and scleral contact lenses are some of the techniques used in the eye detection procedure. The visible spectrum approach locates the iris's location efficiently and non-invasively by using infrared light. However, in situations when I have little control over the lighting, these methods are less accurate and need for extra equipment [16].

The aim of this research is to present an eye identification method that utilizes low-resolution facial images taken using a reasonably priced camera. The main contributions of this work are as follows: We generate a fresh dataset with images of the eyes. We provide two methods for identifying the eye region: the Haar cascade technique and facial landmarks. Both direct methods (convolution neural networks (CNN)) and distance-based engineering techniques may be used by cameras to identify the iris region of the eye in real time. Within a limited mobility range, the proposed deep learning-based method successfully detects the eye's look direction. However, engineering techniques improve its performance throughout a wide range of mobility.

Research Problem

Virtual reality, augmented reality, computer control, consumer behavior monitoring, medical treatment, and security systems are just a few of the many applications that depend on eye tracking and detection. However, a variety of problems, such as variations in eye appearance, occlusion, and ambient noise, make accurate eye tracking challenging to accomplish. Webcams that are not modified have trouble following their users' gaze due to the lighting conditions, wide-angle lenses, and poor image quality. Many consumers cannot afford the commercial eye-tracking systems that are already on the market, such Tobii EyeX, since they need specialized hardware and are prohibitively costly. Therefore, a system that can do real-time gaze tracking and eye identification while being inexpensive and able to work well with standard cameras is needed.

Aims of the Research

The aim of this project is to create an eye recognition and gaze tracking system that is both economical and

efficient using low-resolution facial pictures taken with a low-cost camera. The goal of this research is to determine whether applying deep learning-based processes in combination with conventional engineering methods may improve gaze estimation accuracy across a range of mobility ranges.

Research Objectives

- To develop a fresh eye image collection for study and development.
- To explore and deploy facial landmarks and Haar cascade eye region recognition algorithms.
- To test deep learning-based convolutional neural networks (CNN) for real-time eye tracking in a constrained mobility range.
- To evaluate engineering-based distance measuring approaches for gaze direction tracking throughout a greater mobility range.
- To create a low-cost eye tracking system utilizing webcams without infrared sensors or expensive gear.
- To evaluate the suggested methods under varying illumination and head motions.

2. LITERATURE REVIEW

In human-computer interaction (HCI), gaze tracking and eye recognition have grown in importance (17). Applications in a number of fields, including as virtual reality, augmented reality, healthcare, security, and consumer behavior research, have been made feasible by these elements. Inference of user intent, increased accessibility, and improved interaction efficiency are all made possible by system capabilities that can accurately detect and track eye movements. Due to their reliance on pricey infrared cameras, traditional gaze-tracking systems are only accessible to researchers and private clients. In response, recent advances in deep learning and artificial intelligence have enabled the low-cost implementation of gaze-tracking systems using standard webcams. As a result, the technology is now more affordable and widely available. In this study, the methods, issues, and uses of gaze tracking and eye detection are examined, with an emphasis on affordable solutions [18].

Initially, eye-tracking methods focused on hardware-intensive technologies like infrared-based tracking, video-oculography, and electro-oculography (EOG). EOG relies on the monitoring of electrical impulses generated by eye movements to give a high temporal resolution. Nevertheless, it is vulnerable to noise and signal drift [19–20]. In contrast, VOG uses high-speed cameras to track eye movements by observing corneal and pupil reflections. Although this approach requires the use of regulated lighting conditions, it provides higher spatial precision [21]. Near-infrared light is necessary for infrared-based eye tracking, which is frequently utilized in commercial systems like Tobii EyeX, to increase contrast and enhance detection accuracy. However, obtaining this technology is quite challenging due to its exorbitant cost. Even though there have been amazing advancements, eye recognition and gaze tracking remain challenging jobs because of several factors. Differences in eye appearance are linked to one of the biggest challenges. The detection accuracy is affected by differences in eye shape, pupil size, and iris color. Tracking is significantly hampered by occlusion, which can be caused by eyelashes, glasses, and reflections, particularly in environments that are realistic of the real world [22–23].

The existence of external noise, which can be brought on by changing lighting conditions, head movements, and camera quality, significantly reduces the accuracy of gaze estimation models. The necessity for controlled circumstances is sometimes a limitation of traditional gaze-tracking systems, making them less applicable to real-world scenarios. To overcome these problems, researchers have looked at a number of machine learning and deep learning-based methods for eye recognition [24]. Machine learning techniques like Haar cascades and Histogram of Oriented Gradients (HOG) in conjunction with Support Vector Machines (SVM) have demonstrated a high degree of accuracy in detecting eye regions [24–25]. Alternatively, these methods struggle with real-time performance and rely heavily on features that are developed. Eye recognition has been transformed by deep learning methods, specifically Convolutional Neural Networks (CNNs), which automatically extract important features from images [26]. As a result, the field has advanced significantly. CNN-based models, including EyeNet and GazeNet, have proven to be more accurate and resilient across a range of scenarios. Utilizing hybrid approaches, which combine traditional engineering methods with deep learning, can further increase gaze-tracking reliability [27].

Since appearance-based methods use photos of the eye region or the eye area itself to predict gaze direction, they are well suited for low-cost applications. CNN-based gaze estimation is frequently used after facial landmark identification to identify the eye region [28]. Face recognition techniques are another name for these methods. Haar cascade classifiers, a popular method for object identification, have also been adapted for eye detection, however their performance is limited under uncontrolled conditions [29]. This technology has been used in a variety of ways. Large datasets are used by deep learning algorithms such as i-Tracker and Gaze Capture to improve gaze estimate accuracy. Zhang et al. [30–34] claim that these models use multi-modal inputs, such as head position and eye appearance, to make them more resilient. Because they increase estimation accuracy in scenarios where there is a substantial concentration of motion, engineering-based techniques, including measuring the distance between ocular landmarks, are a helpful addition to deep learning methods.

Commercial eye-tracking devices, like Pupil Labs and Tobii EyeX, are prohibitively costly, usually running into the hundreds [31]. Despite their high precision, these technologies remain unaffordable. Such systems require specialized hardware, such as infrared cameras and sensors that are specifically made for the purpose, to precisely track eye movements. However, open-source and low-cost alternatives combine deep learning-based methods and simple cameras to achieve similar performance at a fraction of the cost [32-34]. By using lightweight models, systems like OpenFace and MediaPipe FaceMesh may provide real-time eye tracking. As a result, these systems may be used for both commercial and scientific purposes [35-36].

The creation of efficient and reasonably priced gaze tracking technologies enables the technology to be used more widely in fields including assistive technology, gaming, and education. The particular lighting conditions used have a significant influence on how well eye-tracking systems operate. Although they need specialized cameras, infrared-based solutions can lessen the effects of ambient lighting. Because shadows and reflections affect the subject's eye vision, appearance-based gaze assessment algorithms struggle under changing lighting situations [35]. To address this problem more effectively, researchers have looked at adaptive learning techniques [36–40]. To increase the resilience of these methods, models are trained under various lighting conditions. Histogram equalization and contrast correction are two picture enhancement techniques that may be used to increase the visibility of the eye region in low light. By capturing eye movements from many perspectives, multi-camera systems reduce the requirement for regulated lighting while also improving accuracy. The development of a low-cost, widely available gaze-tracking technology has significant implications for several disciplines [38]:

1. One example of assistive technology that increases accessibility is eye-tracking, which enables individuals with disabilities to use computers and communication devices.
2. Gaze-based interfaces may be used to improve the user experience in gaming, augmented reality, and virtual reality applications.
3. Businesses use gaze-tracking data to better understand client engagement and improve advertising campaigns while doing consumer behavior analysis.
4. Neurological disorders including Parkinson's disease and autism can be diagnosed with the use of eye movement analysis. This study was carried out in the fields of psychology and medicine.

Notwithstanding the advancements in low-cost gaze tracking, some research challenges remain. The present models' application in real-world circumstances is limited due to their inability to handle variations in head movement. To provide seamless interaction with consumer devices, real-time processing efficiency also has to be improved [34].

1. Improving model generalization across a range of head positions and lighting conditions.
2. Investigating hybrid approaches that combine conventional engineering expertise with deep learning.
3. Real-time processing might be enhanced by using edge computing and lightweight neural networks.
4. To address the ethical issues surrounding data collection, eye tracking solutions that safeguard users' privacy are being developed.

Research Gap

Even though eye tracking technology has advanced significantly, there are still a lot of issues and

limitations that need to be resolved and further research is required. The trade-off between accuracy and cost is one of the most significant issues presented by eye tracking systems. Despite the fact that costlier systems offer more precision, their unreasonably high cost and complex setup prevent many academics, developers, and businesses from investing in them. nonetheless, low-cost alternatives, including webcams or deep learning-powered systems, are more cost-effective; nonetheless, they lack infrared tracking, high-speed data processing, and stringent calibration procedures [41].

The need for innovative hybrid solutions that combine the precision of high-end devices with the affordability of low-cost systems without raising the overall complexity or cost of deployment is necessary. Another important research gap that has to be filled is the precision and dependability of gaze tracking in dynamic environments. The expensive systems are designed to be used in controlled laboratory settings with minimal background noise, head position, and lighting conditions. However, gaze tracking often faces challenges in real-world applications such virtual reality interactions, driver monitoring, and assistive technology for those with impairments [34]. These issues include head position variations, occlusions, and motion artifacts. Existing low-cost models struggle with real-time adaptability, especially for applications that run on mobile devices or outdoors. Future research should primarily focus on developing robust algorithms that maintain low latency while improving gaze prediction accuracy in various uncontrolled scenarios.

In fields like psychology, neurology, and marketing, where consistency and repeatability are crucial for confirming results, it is particularly problematic [42]. In some sectors, this discrepancy is particularly troublesome. The development of interoperability frameworks, standardized datasets, and universal benchmarking protocols will greatly increase the usability of both low-cost and high-cost systems across disciplines. Because there are so few open-source datasets for gaze tracking, researchers are constrained in their ability to effectively train and assess AI-based models. Even while gaze estimation in low-cost webcam-based tracking has improved using deep learning-based techniques, infrared-based high-speed tracking systems continue to outperform them in terms of accuracy [34]. This is due to the fact that deep learning algorithms rely on machine learning.

One of the biggest issues is the dependence on large datasets with a diverse variety of data that are used to train AI models, as most publicly accessible datasets are biased toward certain demographics and controlled conditions. Future research should primarily focus on developing more inclusive datasets, improving gaze estimation models driven by AI, and minimizing processing overhead for real-time applications. As the use of AI-based gaze monitoring becomes more commonplace, protecting users' privacy, ethically gathering data, and protecting their security will also be crucial challenges that need more investigation.

3. RESEARCH METHODOLOGY

The aim of this research is to comprehend Pakistani consumers' animosity towards online retailers and the consequent rise in the percentage of online sales.

Research Design

This study investigates low-cost real-time eye identification and gaze tracking methods utilizing quantitative experiments. The study collects, preprocesses, models, and evaluates several eye tracking methods for cost-effectiveness, accuracy, and utility. Comparative research compares appearance-based deep learning with engineering-based methods in particular situations.

Datasets

The approach provides a dataset of eye scans from diverse ethnic and geographical origins. The local eye dataset featured Asian eye photos, whereas the global eye dataset had European eye photos. This study employed four datasets: two for training models and two for testing results. Two datasets are used for training:

1. SBVPI (Sclera Blood Vessels, Periocular, and Iris) has 1800 images of 55 people. These photos are categorized by gender, eye class, and view/gaze-direction. The collection has 450 pictures per look direction with 1700–3000 pixels. Figure 1 shows how a Canon 60D DSLR was used to take RGB images in a single recording session [43-46].

Figure 1: Global Dataset

2. Only a camera, ideally incorporated into the laptop's upper screen, was needed to produce datasets using the suggested technique (Figure 2) [6].

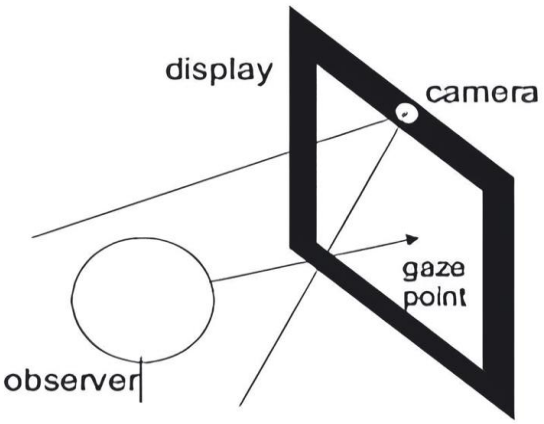


Figure 2: Experimental Setup

On the basis of these assumptions, a person is supposed to be sitting straight facing the screen with the embedded camera at the highest point of their horizontal gaze and fantastic illumination. This assumption allows real-world photography instead of lab photography. Following the algorithm's phases, the recommended system may reliably predict eye gaze:

1. Align the computer screen with the nostrils of the user.
2. Sit with your head straight and facing webcam, avoiding minimal head movement.
3. It is advised to set the computer within 35-50 cm of the individual. By mathematical equation, the optimal user-screen distance may be established [1].

$$D = \sqrt{[P_{Ux} - P_{Sx}]^2 + [P_{Uy} - P_{Sy}]^2} \text{..... [1]}$$

This equation uses D to indicate proxy distance, (P_{Ux}, P_{Uy}) to represent user location, and (P_{Sx}, P_{Sy}) to represent screen computer position. To create a local ocular dataset, two strategies are used:

1. The research used a local collection of human eye pictures to study eye movement in four directions. The images were obtained from fifteen persons with varying skin pigmentations. Haar cascade was used to identify the photographs' eye areas. The iris does not rise much, hence the technique usually focused on eye upward movement. The approach focuses on upward movement, whereas downward movement is similar to closing motion. For the investigation, 5,000 photos were obtained in different lighting (Figure 3).

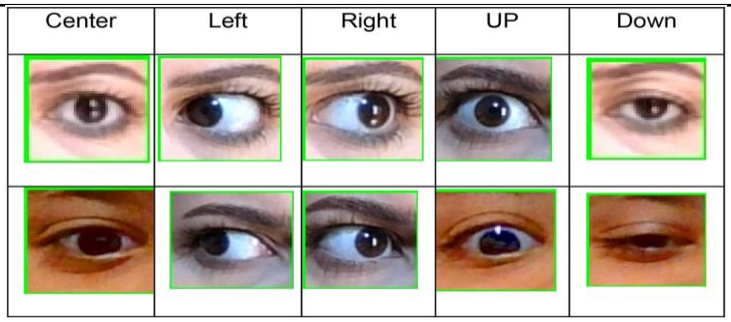


Figure 3: Haar – Local Dataset

2. The study used a local camera collection of human eye photographs to investigate eye movement in four directions while individuals were exposed to interior illumination and wearing different skin tones. The MediaPipe Face Mesh method identified the photographs' eye regions. The dataset included 15 contributors: One thousand photographs represent each of the five eye directions: left, right, center, top, and bottom (Figure 4).

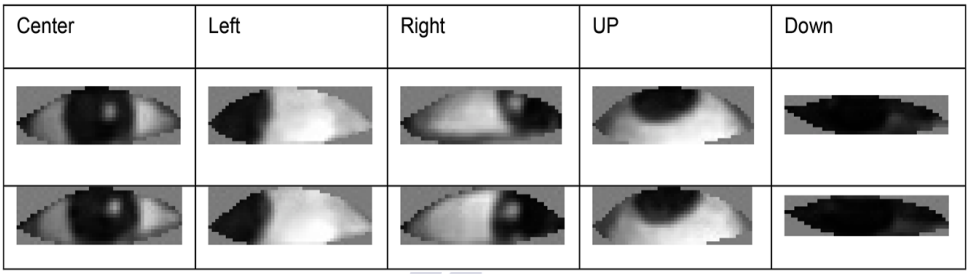


Figure 4: MediaPipe – Local Dataset

In the final analysis, local datasets were prioritized since they are more diverse and acceptable. In the real-time test, two approaches were used to gather data. The camera takes real-time photos in the first way. Downloading an eye movement video from YouTube is the second technique. The videos used were face-focused since the eye image must be extracted. The videos used are shown in Table 1.

Table 1: Video Dataset

Source	**YouTube**
Type	Mp4
Gender	Female
Size	8.87 MBs
Resolution	1920 × 1072
Frames per second	30
Duration [s]	20
Total images	500
Included images	451

Eye Detection Techniques

There are two distinct methods that are utilized, as will be discussed in the following:

1. Several research [47-49] suggest using the Haar cascade to recognize faces or feature coordinates. Viola and Jones [50] proposed the Haar cascade for visual object recognition. A trained Haar cascade uses a drawn rectangle to determine if a picture contains the required item. The Haar methodology is more efficient than previous methods because it uses high-speed computations based on the number of pixels inside the rectangle feature rather than the image's pixel values. Haar-like feature, integrated image, AdaBoost learning, and Cascade Classifier are used to detect the object [51-52]. Haar-like features are used to build the Haar Cascade Classifier for face detection.

Haar characteristics are used to identify if an image has a certain trait. Each feature generates one value by adding all pixels below the black rectangle. For fast face recognition, a picture is given a rectangular Haar-like component [53]. Figure 5 shows frequent Haar-like features [51]. Haar Cascade Classifier incorrectly identifies an additional image feature as the ocular characteristic [6]. The categorization process becomes less accurate.

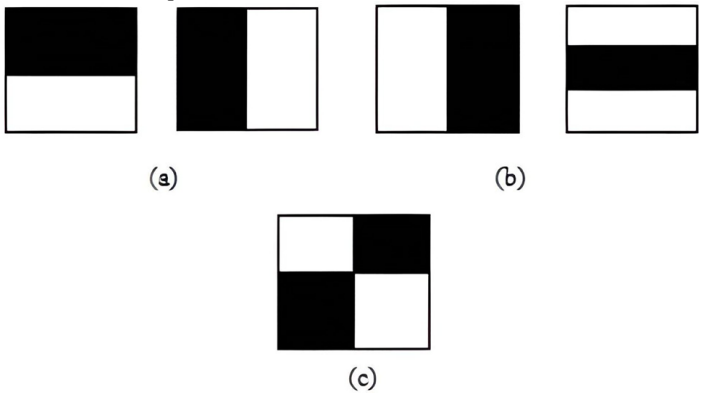


Figure 5: Type Of Feature (a) Edge, (b) Line, (c) Four-Triangle

2. For face landmark extraction, there are several methods, however these are significant for our study:

- Kazemi and Sullivan's Regression Tree Ensemble (ERT) [54] tells the Dlib package to employ landmark identification. According to MultiPIE [55], this approach can extract 68 face landmarks quickly and easily (Figure 6) [53].

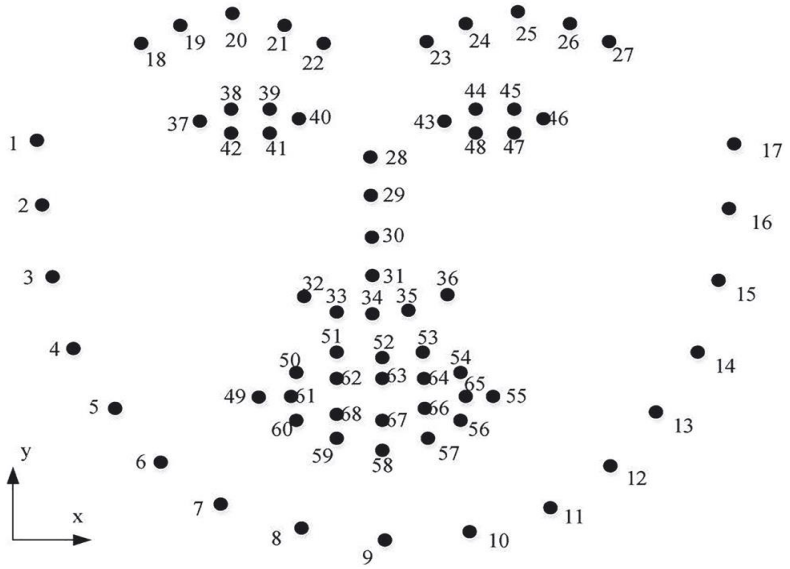


Figure 6: Map For Dlib Facial Landmarks

A continuous strategy using a cascade of regression coefficients changes these predicted locations. A new estimate is reliant on the previous one at each iteration for regressors. Misaligned projected points generate variance, which these estimates aim to eliminate [1]. In our investigation to determine eye shape, we used the dlib library. Starting with dlib.get frontal face detector, the facial shape is determined. We next entered facial data into dlib to estimate eye form (shape predictor 68 face landmarks.dat) [55].

- A powerful library, the media pipe [56-57], can detect faces and facial landmarks. Eye photos are available from the library. MediaPipe face Mesh extracts 478 facial landmarks using a residual neural network [58] (Figure 7).

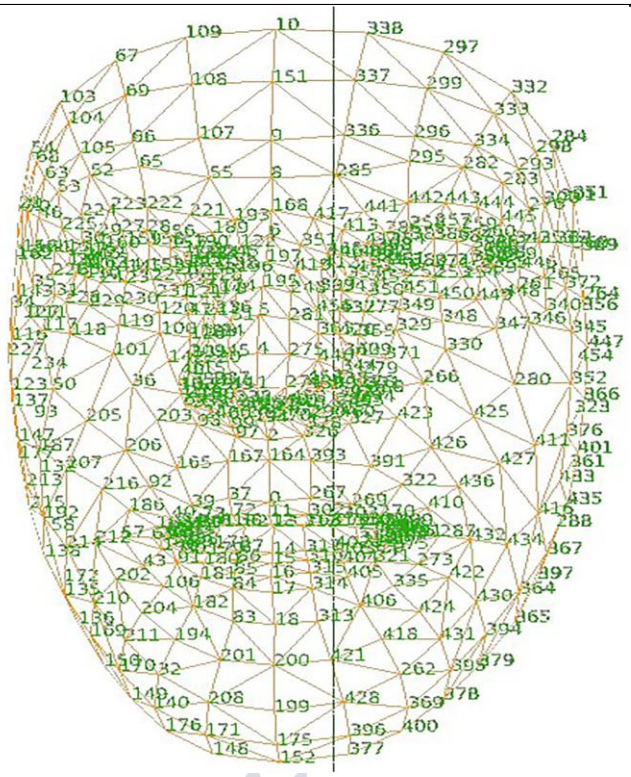


Figure 7: MediaPipe Face Mesh Solution Map

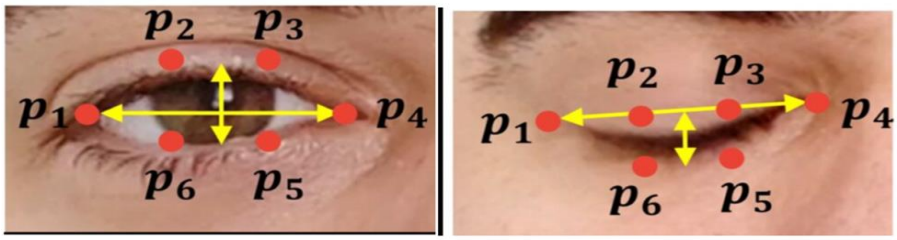
In order to identify the face and eyes, the following landmarks were utilized in this study (Table 2): Using MediaPipe face nets [59], one may construct a three-dimensional face representation and nose values for head position computation.

Table 2: Indices of the Landmark

Face Border	Left Eye	Right Eye
[10, 338, 297, 332, 284, 251, 389, 356, 454, 323, 361, 288, 397, 365, 379, 378, 400, 377, 152, 148, 176, 149, 150, 136, 172, 58, 132, 93, 234, 127, 162, 21, 54, 103,67, 109]	[362, 382, 381, 380, 374, 373, 390, 249, 263, 466, 388, 387, 386, 385,384, 398]	[33, 7, 163, 144, 145, 153, 154, 155, 133, 173, 157, 158, 159, 160, 161, 246]

The face mesh solution's X and Y output coordinates are normalized according to frame time. The z vector represents the face wire mesh depth, which represents the camera-head distance. The media pipe command map_face_mesh = mp. solutions specify facial characteristics for eye shape. FaceMesh [55]. The initial phase is reading each movie frame and submitting it to the library for facial landmark identification. Left and right eye pictures are preserved separately [1]. In terms of test reliability, MediaPipe measurements are better than dlib metrics, hence this article used them for eye detection.

- The EAR function monitors eye landmark distance to work effectively. Figure 8 [56] shows that the EAR measure calculates a ratio from six ocular landmark locations' vertical and horizontal distances.



a) open-eye b) close-eye

Figure 8: EAR Features

The EAR is calculated from this ratio. Starting with p_1 and ending with p_6, numerals indicate places clockwise from the left-eye corner. Rosebrock [57] states that all six coordinates in the range from p_1 to p_6 are two-dimensional. According to [58], opening the eyelids does not modify the EAR value considerably. However, the EAR ratio is lowered to zero [59], eliminating the gap between coordinates p_3 and p_5 and p_2 and p_6 while the eyes are closed. This study predicts eye closure using the EAR function. In Equation (2) [60], the numerator calculates the distance between horizontal landmarks, while the denominator calculates the distance between both vertical landmarks and multiplies it by two to equalize it. EAR is calculated by repeating this technique.

$$EAR = \frac{||p_2 - p_6| + |p_3 - p_5||}{2|p_1 - p_4|} \quad \text{..... [2]}$$

Proposed Method

Interactive technology makes eye recognition and gaze tracking more crucial than ever. A built system and a real-time run comprise the recommended system model. Figure 9 shows the preferred system technique.

Figure 9: Proposed System Algorithm

This model has two stages built as follows:

Pre-Processing:

1. A photograph is taken using a camera.
2. Improves image illumination using the HSV Color Space, which includes color [H], saturation [S], and value [V]. The first two describe hues, whereas the third and last one describes brightness [61].
3. HSV is preferred for geometric coordinate systems due to its natural appearance and better color hue modulation compared to HSL [62]. The camera of the recommended system takes RGB photos, but Python is used to convert these to BGR. The color formula is returned to [BGR] when the color format is converted to (HSV) to regulate saturation and color value degree. This improves input photo illumination for a precise prediction. We improved the lighting while keeping all colors to return the hues buried in the gloom and simplify the brightness model. The Split-HSV technique is recommended for illumination improvement:

- The color space should be switched from BGR to HSV.
- Isolate the saturation and value channels first. Next, increase values to achieve saturation of (1.5 × S) and value of (60 + V). Please note that none of these numbers should exceed 255.
- Re-merge channels.
- Return to BGR space.
- The third phase is image value normalization, or pixel value normalization. To normalize the operation, divide the data by 255. Each pixel in the picture data should have a value between 0 and 255.

4. Reduce image complexity and noise by converting to grayscale.
5. Use Haar cascade face bounding boxes or MediaPipe facial landmarks model to predict 478 facial signals and precisely chop off the eye region.

6. To aid models during training, pictures must be rescaled to a 64 × 64 input format. Image sizes should be equal.
7. Divide the dataset into training and testing divisions. We split the data into training and testing sets for this study.

8. **Processing:**
The most common way to establish eye direction is to track the iris. First, locate the eye's region to find the iris. This study utilized the MediaPipe library to better describe facial features. Landmarks from dlib or MediaPipe were used to do this. The Haar Cascade Classifier is used to determine the ocular area. Two approaches may be used to estimate eye look direction using simply the camera. Here are the methods:

1. **The Direct Method is the Convolution Neural Networks (CNN) Method:** Deep neural networks like CNN use grids for input analysis. Convolutional, pooling, and FC layers comprise this arrangement. The pooling layer samples feature maps while the convolutional layer filters data. The FC, the last layer, produces the final output using an activation function such sigmoid or SoftMax [63] (Figure 10, Table 3).

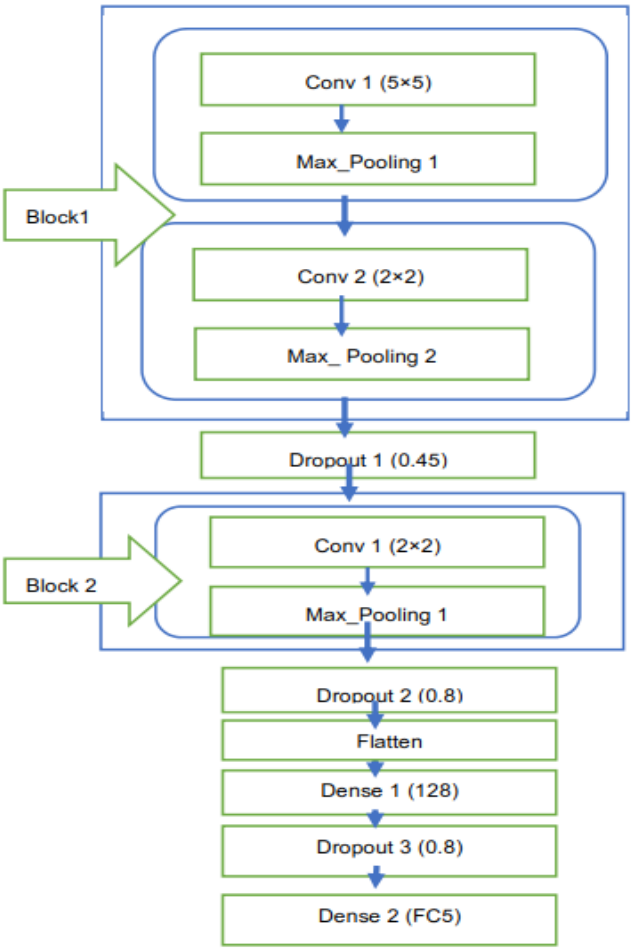


Figure 10: Architecture of Di-eyeNET

The Di-eyeNET model was trained on 5,000 eye images. The dataset consisted of 4,000 training photographs and 1,000 testing shots. This ensured fair assessment. The dataset was separated into five categories; hence the classification challenge was several classes. Since the model included 46,820 trainable parameters and no non-trainable parameters, all of them were actively tuned during training. Due to the 100-epoch training technique and 32-photo batch size, the model processed 32 photographs at a time before changing the weights. The categorical cross-entropy loss function, used for

multi-class classification, was chosen. This function measures the difference between the anticipated probability distribution and the actual class labels to assist optimization and decrease errors. A remarkable validation loss of 0.0183 was achieved by the model during training. A minimal variance exists between predicted and actual outcomes. The model's validation accuracy was 99.59%, indicating strong classification precision. Early stopping was adopted after 98 epochs since validation loss stopped improving. This prevented overfitting and reduced unnecessary calculations. These data suggest Di-eyeNET is very good at eye detection and gaze classification. The system's settings have also been tweaked for accuracy and reliability (Table 3).

Table 3: Summary of the Parameters for Training of Di-eyeNET

Parameters	Local
Total Images	5000
No. Images-Train	4000
No. Images-Test	1000
No. Class	5
Total Parameters	46820
Trainable Parameters	46820
Non-Trainable Parameters	0
No. Epoch	100
Batch-Size	32
Val-Loss	0.0183
Val-Accuracy	0.9959
Early Stopping	After 98 Epochs Val-Loss Not Improved
Loss Function	Categorical Cross-Entropy

The proposed model, Di-eyeNET, has two blocks, as illustrated in Figure 10 and Table 3:

1. The first block consists of two convolutional layers, each with 128 5x5 filters and a stride of
2. This collects enough geographic data while reducing output feature map dimension. ReLU activations in the max-pooling layer below it lowers the feature map by at least two dimensions.
3. Next, the second block is used, with a max-pooling layer with two filters and a convolutional layer. After block 1, feature map spatial dimensions are halved. After each max-pooling function, we gradually add filters to the convolutional layers.

The output of the final two blocks is routed to a filtered layer and then to a 128-D Fully Connected (FC) layer to ensure feature map representation capacity. The suggested design avoids several FC levels to reduce trainable parameters while retaining performance. In conclusion, the FC layer is coupled to a single SoftMax layer that identifies four eye directions. A two-step procedure was needed to extract only the necessary features and ignore the rest: Block 1 of the first stage has two convolutional layers. After the first stage, we removed half of the variables [characteristics], thus we applied dropout after block 2 to reduce the number of variables by around a fourth. The model was run for over 100 epochs using an Adam optimization approach, a learning rate of 0.001, a batch size of 16, and an MAE loss function. Additionally, the input shape was 64x64. Each model input picture has three channels. This study used Keras, an open-source Python neural network toolkit, to build CNN models.

We also used ResNet50 and VGG16, the most efficient pre-trained model architectures. The convolutional neural network VGG16 is developed for image recognition. Instead of several hyper-parameters, it uses sixteen layers and weights, making it unique [64-65]. We used pseudo code to import and load the VGG16 pre-trained model:

- Import VGG16 from Keras apps.
- Model: VGG16
- The picture size must be (224 × 224).

The Resnet50 classic neural network underpinned several computer processes related to the skip connection principle. We train CNN with this 150-layer model [64]. We used pseudo code to load the pre-trained ResNet50 model:

- Load map ['resnet_50.h5'] in keras. models.
- Image size must be [64 × 64].

The Di-eyeNET, ResNet50, and VGG16 training experiences included multiple trainings. The Di-eyeNET model outperforms other techniques in some parameters, according to results studies. Di-eyeNET, ResNet50, and VGG16 had accuracy of 0.9959, 0.7933, and 1.7965, respectively; MAEs of 0.0183, 0.5532, and 0.1416, respectively; loss values are almost identical during testing; VGG16 has greater loss values during training. Compared to other models, the suggested model is 75% smaller and has 80% less reaction time.

2. **Engineering Method:** The engineering approach employs Python modules to detect facial and eye characteristics after locating the eye region by calculating distances. Several approaches may be used to locate the eye's iris, which reflects the gaze:

Step 1: Draw two perpendicular lines on the iris region, the darkest part of the eye, using certain equations (horizontal and vertical). This area is determined via contour detection and Hough transform. After that, we find the place where the two straight lines connect, which represents the iris and the gaze.

Step 2: The Euclidean Distance function was used to find the intersection ratio between two perpendicular lines in Python.

Step 3: Divide the eye region into five unique regions: three horizontally (left, center, right) and two vertically (up, down).

Step 4: To assess closed-eye condition, the EAR feature calculated the proportion of distance between eye landmarks.

Experimental Setup and Evaluation

The proposed solution uses Python packages Num Py, Open CV, Tensor Flow, Keras, dlib, and MediaPipe to train models. We developed this system using a laptop with a GeForce GTX graphics card, 16GB RAM, and an Intel Core i7 processor. Anaconda was used for development. Training and testing were separate stages of system implementation. Experimental data are used during the key assessment phase to predict results and evaluate algorithms. Accuracy, loss [MAE], response time, and model size [Equation 3] are used to evaluate model performance. The symbols TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives.

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} \times 100 \text{..... [3]}$$

In statistics, the Mean Absolute Error [MAE] indicates the loss between the measured [y′] and the real [y]. This equation [4] shows the MAE formula.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} |Y_i - Y_i'| \text{..... [4]}$$

The total number of classes is n, the measured output is y′, and the actual output is y. Response time is crucial for real-time applications. In real time, size models must load swiftly and easily. Large models are hard to load quickly. Except for the provided model, all other models are too huge for real-time use.

Data Processing and Analysis

Data processing step affects eye identification and gaze tracking quality and efficiency. Noise, light variations, head movements, and occlusions affect low-resolution webcam image data, making preprocessing essential. Data processing begins with grayscale conversion to reduce computer complexity while preserving ocular features. Next, histogram equalization boosts contrast and feature detection in diverse lighting conditions. Gaussian and median filtering minimize noise and clarify eye contours. The ocular region is separated by features after preprocessing. Haar Cascade Classifier machine learning finds eye boundaries and pupil location using predefined patterns. A deep learning-based CNN is trained on the dataset to improve eye recognition in various lighting conditions and head postures. Deep visual signals retrieved by the CNN model allow accurate iris and gaze evaluation under challenging conditions. Facial landmark identification using Dlib-based mapping of eye-important locations is also incorporated in feature extraction for enhancement. The gaze tracking study uses engineering- and appearance-based methods after ocular area recognition. Appearance-based methods estimate gaze direction by measuring the relative placement of the iris inside the eye, whereas engineering-based methods use geometric distances

and vector calculations to improve gaze direction accuracy over a wide range of head movements. These methods are assessed by real-time processing speed, gaze estimation accuracy, and detection accuracy.

4. RESULTS AND DISCUSSION

This study offered four eye-look tracking methods: CNN of Haar, CNN of Mesh, Mesh of Inter. Point, and Mesh of Split. We outline both ways below. CNN of Haar and CNN of Mesh are first-direct approaches. Both approaches use convolutional neural network training. The Python module MediaPipe calculates facial landmark distances for Mesh of Inter. Point and Mesh of Split. The Mesh of Inter. Point technique uses functions to identify the eye's iris, draw the two perpendicular lines, and find the straight lines' intersection point. The Mesh of Split technique has two functions: to divide the eye into five portions and to determine the iris in any place after the division. The darker region represents the direction of look. The results in the following tables were obtained by applying the proposed lighting improvement strategy because the Haar cascade technique without an improvement strategy was rare and did not exceed 50% in a poorly lit examination room. Due to the high quality of the data, real-time research is difficult, but if standard criteria are not considered, the accuracy of these results may be reduced by 50%. CNN approaches were almost 99.0% accurate during training, but much lower during real-time evaluation due to illumination and subject seating position issues. This is because camera angles alter the quality of the image supplied to the network, which affects the intended output. Thus, when the following tables were created, these factors were included, allowing an objective comparison of the offered techniques.

Haar Cascade Classifier was used to create CNN using Haar training dataset images. This was done by sketching and truncating a rectangle around the eye. The cut-out eye region connected the brow and eye regions to the eye. Thus, the Di-eyeNET method's vast number of characteristics affected real-time test decision-making. It recognized three classes but not the top direction in real time. Despite 99% training accuracy for all classes, test accuracy was 93%. Table 4 shows accuracy findings from mobility-limited experiments. These trials tested many eye-tracking technologies. The table compares CNN of Haar, CNN of Mesh, Mesh of Interpolated Points, and Mesh of Split for recognizing right and center eye gaze positions in 15 people. Four methods are compared. CNN of Mesh was the most accurate approach. It has a remarkable 94.40% accuracy with 99.5% center gaze and 97.85% right gaze. This suggests CNN-based techniques, especially mesh detection ones, are effective in looking at someone. CNN of Haar had 94.55% accuracy and did well. It identified gaze direction with 98.32% accuracy for center gaze and 95.10% accuracy for right gaze. Mesh of Interpolated Points and Mesh of Split had lower statistical accuracy. Mesh of Interpolated Points finished with 90.90% accuracy due to lower right (93.80%) and center (95.95%) accuracies. Mesh of Split performed poorest, with 88.9% accuracy. Right and center gaze detection errors were higher (84.8% and 95.7%, respectively).

Table 4: Results of the Accuracy Experiments (Limited Mobility Range)

Proposed Methods	No. of Individual	Accuracy (%)					Total Accuracy %
		Right	Center	Right	Center	Right	
CNN of Haar	15	95.10	98.32	95.20	86.88	84.5	94.55
CNN of Mesh	15	97.85	99.5	99.5	95.50	95.8	94.40
Mesh of inter. P.	15	93.80	95.95	92.05	90.52	92.0	90.90
Mesh of split	15	84.8	95.7	85.0	87.9	78.56	88.9

Even after repeating the experiments with more photos, people, and training situations, no appropriate results were obtained. Due to comparable qualities that affected resolution, such as the brow, which makes up a large part of the image. Because of this, we used the mesh function to produce dataset pictures. These eye-only visuals were unique. This is because the mesh function constructs a frame around the eye and crops it solely. No more features are added because it solely focuses on iris properties throughout CNN training. This made the training and assessment outcomes 99.7% correct for every class. Table 5

compares several gaze-tracking systems' accuracy. We found that the Mesh of Interpolated Points method has the highest accuracy of 92.90% in all gaze directions. The center and left locations had 99.5% accuracy, followed by the right stare at 97.85%, the top gaze at 95.50%, and the down glance at 95.8%. This shows that it can handle high mobility with some inaccuracy.

CNN Haar also performed well, obtaining 93.05% accuracy and retaining accuracy throughout gaze directions. It achieved 95.95% center gaze accuracy, 93.80% right gaze accuracy, and above 90% gaze direction accuracy from other orientations, demonstrating robust tracking capabilities. Mesh of Split, with a total accuracy of 90.6%, performed somewhat worse but still competitively in the center (95.92%) and correct direction (93.77%). CNN of Mesh had the lowest accuracy (88.8%), with most problems in the left (85.0%) and down (78.56%) gaze directions. Even while it scored well in center identification (95.7%), its lower accuracy in other directions implies it has mobility management limitations. Mesh of Interpolated Points outperforms CNN-based eye gaze tracking techniques in wide mobility circumstances. CNN-based systems, like CNN of Mesh, lose accuracy with larger gaze shifts. Even though CNN of Haar is a strong alternative.

Table 5: Results of the Accuracy Experiments (Wide Mobility Range)

Proposed Methods	No. of Individual	Accuracy (%)					Total Accuracy %
		Right (0)	Center (1)	Left (2)	Top (3)	Down (4)	
CNN of Haar	15	93.80	95.95	92.05	90.52	92.0	93.05
CNN of Mesh	15	84.8	95.7	85.0	87.9	78.56	88.8
Mesh of inter. P.	15	97.85	99.5	99.5	95.50	95.8	92.90
Mesh of split	15	93.77	95.92	91.09	90.48	93.5	90.6

Accuracy, mean absolute error [MAE], model size, and reaction time all affect the offered approaches' results. The findings reveal in Table 4 that the CNN method outperforms the engineering method. Table 6 compares widely developed eye-tracking methods. Performance factors including accuracy, mean absolute error, response time, and load time are analyzed. Haar's CNN had the highest accuracy, 93.05% for broad mobility and 94.55% for confined mobility. Other approaches were less accurate. It also has a low MAE of 0.075 and 0.078, indicating few gaze detection errors. CNN of Mesh performed somewhat worse in the wide range (88.8% ACC) but considerably better in the restricted range (94.40% ACC) with a much lower mean absolute error (0.018 for wide and 0.031 for constrained mobility). The Mesh of Interpolated Points technique performed well, with 92.90% accuracy in broad mobility and 90.90% in confined mobility. However, this strategy had greater MAE values than CNN-based techniques (0.035 for wide mobility and 0.057 for limited mobility), suggesting gaze estimation mistakes. Like the preceding technique, the Mesh of Split method showed good accuracy (90.6% for broad mobility and 88.9% for limited mobility), but it had higher MAE values (0.045 and 0.197), indicating more prediction errors.

CNN of Mesh had the fastest real-time reaction time, one second, but the longest load time, twelve seconds. It was less efficient for fast-initialization applications. However, CNN of Haar balanced reaction time (1.5 seconds) and load time (4 seconds) for an efficient and stable approach. Mesh-based techniques yielded different results: Although the Mesh of Interpolated Points had a lengthy reaction time of three seconds, its load time was small at 5 S. Mesh of Split had a response time of 2 S but required 10 seconds to start, making it slower to load than similar applications. For real-time processing, CNN of Haar is the best option due to its high accuracy, low MAE, and good response/load times. CNN of Mesh has the lowest mean absolute error; however, its high load time may limit its use in dynamic situations. Mesh of Interpolated Points offers equivalent accuracy but slower reaction times, making it unsuitable for real-time processing. Mesh of Split has a higher mean absolute error than other alternatives, hence it may not be suited for precise applications.

Table 6: Measures of The Evaluation (The Proposed Methods)

Proposed Methods	Mobility Range	No. of Individual	ACC	MAE	Response Real-Time	Load Time
CNN of Haar	Wide	15	93.05	0.075	1.5 S	4 S
	Limit	15	94.55	0.078	1.5 S	4 S
CNN of Mesh	Wide	15	88.8	0.018	1 S	12 S
	Limit	15	94.40	0.031	1 S	12 S
Mesh of inter. P.	Wide	15	92.90	0.035	3 S	5 S
	Limit	15	90.90	0.057	3 S	5 S
Mesh of split	Wide	15	90.6	0.045	2 S	10 S
	Limit	15	88.9	0.197	2 S	10 S

CNN is the direct technique since it employs one function, while engineering requires numerous. Neural networks determine whether the human eye looks up, forward, or down, making the engineering technique more accurate than CNN. This is done by measuring the upper eyelid and iris movement. Additionally, CNN of Mesh is the fastest and most accurate method. The best approach (CNN of Mesh) takes longer to load than other methods, depending on software size. Its real-time reaction time is shortest. This is important because faster reaction times make software more user-friendly. Because of this, CNN-based methods perform better in the narrow mobility range and geometric methods in the vast mobility range. People must keep their heads still and stay 50 cm from the screen. This is because the camera cannot capture clearer photos from further away. We also advised improving camera lighting to improve real-time performance. The table shows that accuracy increased without affecting reaction time. When implemented in real time, response time is a key system performance indicator. Despite eye trackers, appearance-based gaze judgment is inaccurate. Due to this unpredictability, camera-based gaze detection algorithms struggle to attain high accuracy [69]. These include lighting, ocular image, and head position changes. All experimenters used intense continuous illumination and maintained a steady head position. In terms of adjustable aspects (equipment, accuracy, technique, dataset, and function), the focus on appearance-based approaches suggests a shift away from eye-tracking technology in research.

The first accuracy experiment assessed eye-tracking models to determine gaze direction with minimal head movement in a limited mobility range. CNN of Haar, CNN of Mesh, Mesh of inter-pupil (inter. P.), and Mesh of split were compared based on their performance across many gaze positions. CNN of Haar functioned best when the gaze was oriented toward the center (98.32%) and poorly when directed toward the right (95.10%) or left (86.88%). CNN of Mesh outperformed other methods with 94.40% accuracy and 99.5% accuracy at the distribution center. This suggests that mesh-based detection convolutional neural networks perform well for static head shape gaze estimation. The mesh of inter-pupil performed well in the middle but lowered rightward and leftward gaze detection accuracy. The mesh has 90.90% accuracy, somewhat lower than CNN-based techniques. The split mesh was the least accurate at 88.9%. It was likely challenging to correctly identify ocular features under different lighting conditions. These results are far better than previous webcam-based eye tracking research. Previous studies using infrared sensors [66] and wearable eye trackers [10] only achieved 80% accuracy, far lower than the least effective method (Mesh of split, 88.9%). This supports the use of deep learning methods like CNNs with Haar and Mesh detection in low-mobility situations.

Performance was assessed throughout a wide mobility range, with gaze positions including right, center, left, top, and down. Haar's CNN had the best overall accuracy (93.05%) and was consistent across all directions, proving its resilience in all head angles. The inter-pupil mesh detected center and leftward gaze with 99.5% accuracy. However, its rightward and downward look detection was far less impressive. This suggests that this strategy is accurate from a controlled frontal perspective but challenging from extreme angles. The CNN of Mesh performed worse throughout a wide range (88.8%) than in the narrow range (94.40%), perhaps because to head movement-

induced ocular distortions. Split-mesh performed worst (90.6%), supporting the trend that CNN-based models outperform split-mesh models. The CNN-based methods outperformed the webcam-based methods in previous research. In 2022, webcam-based research [67] had an accuracy of 84%, but our poorest technique had 88.8%. Only a camera-based CNN and ESR model proved competitive in prior study [15]. This model has 94.39% accuracy, matching CNN of Haar and CNN of Mesh. The new study offers several advantages, including real-time responsiveness and lower computer demands.

Real-time performance measurements reveal speed, accuracy, and computational burden trade-offs. CNN of Haar had a 1.5-second real-time reaction with a mean absolute error of 0.075 under limited and wide mobility settings. This precision was obtained with little error. Based on this, it looks to be suitable for fast, precise real-time applications. CNN of Mesh had the fastest reaction time, one second, but a twelve-second load time. CNN of Mesh looks to be incredibly efficient, but it demands a lot of processing power, making it suited for advanced hardware configurations. The mesh of inter-pupil required three seconds for real-time processing, making it less suitable for fast applications but still accurate. Split mesh had the slowest reaction time, two seconds, and the biggest mean absolute error, 0.197, indicating unreliability.

Research is focused on accuracy, while real-time processing efficiency is often disregarded. Early Tobii EyeX studies [23] used expensive eye-tracking equipment with reaction times of up to five seconds, making them unsuitable for real-world applications. Meanwhile, CNN-based models promise great accuracy and real-time application. They might replace expensive commercial eye-tracking technology because of this feature. Previous research has used webcams, infrared sensors, and Tobii eye trackers. A 2020 study using the Tobii eye tracker [10] and a 2021 study using infrared sensors and a wearable eye tracker [15] both reported 80% accuracy. However, webcam-based techniques have better accuracy, with 2021 [65] obtaining 94.39% accuracy and 2022 [66] achieving 84% accuracy. However, a CNN-based mesh approach improved accuracy to 98.77% in this study. This shows how advanced deep learning and dataset improvements enhance eye-tracking accuracy. Appearance-based models dominated previous research. These algorithms use eye visual properties to predict glance direction. Both the Tobii-based study [10] and the infrared sensor trial [15] used solely this technology, resulting in low accuracy. In webcam-based study [65] and [68], appearance-based methods were used, and the accuracy reached 94.39% in some cases.

The proposed research uses CNN of Mesh architecture, MediaPipe, and a locally curated dataset for model- and appearance-based techniques. This hybrid method enhances accuracy by using structural eye models, making predictions more resistant to light, head movement, and eye shape variation. This study uses a real-time tracking and deep learning-refined dataset. This enhancement allows for a more complete and generally applicable eye-tracking model. Each study used CNN-based methods, with some using ESR for feature extraction. Due to its efficiency, the CNN-based mesh approach utilized in the recommended research maps eye movements more accurately. This work optimizes a local dataset with MediaPipe for real-world and user context flexibility. This contrasts with previous research that used large datasets like 289,222 photographs [10] and 7094 eye scans [65].

5. CONCLUSION

This study produced affordable real-time gaze-tracking software for desktop and laptop computers. No further equipment is needed for this program. Using unmodified personal computer cameras, the cutting-edge CNN network Di-eyeNET was used with the MediaPipe library of Face Mesh. Unadjusted webcam photos are low-quality and light-sensitive. This made attaining great results in real time difficult. In contrast, our research shows that well-organized parameters and a CNN network may produce valuable real-world results. Our research used the proposed model (Di-eyeNET) to determine the direction of gaze on a computer screen using neural networks, and it was successful. However, engineering solutions work better when gaze direction is necessary within a large mobility range. Compared to previous methods, the proposed technique worked. It is accurate and user-friendly since it does not involve intrusions and is based on looks.

Successfully gathered findings show significant advancement in precision, real-time responsiveness, and computer efficiency. These improvements make CNN-based eye-tracking a viable alternative to conventional methods, which need expensive and specialized gear like infrared sensors and wearable eye trackers. One of the most notable discoveries of this research was that CNN of Haar and CNN of Mesh models consistently showed over 94% accuracy throughout mobility ranges. The precision of these models is a key discovery. CNN-based approaches appear to be able to detect eye movements robustly across a number of gaze positions, including center, left, right, top, and down, without losing accuracy. CNN models have proven dependable in assistive technology, human-computer interaction, and medical diagnostics, where precise gaze tracking is essential. CNN models are reliable due to their great accuracy.

Another finding of the study is the need for real-time processing in eye-tracking devices. The CNN of Haar and CNN of Mesh techniques have reaction times of one to one and a half seconds, making them ideal for dynamic applications that need fast input. Virtual reality, gaming, and disability aids benefit from this real-time capability. In many situations, even small reaction time delays might damage user experience and tool usability. The study shows that MediaPipe-based local datasets work. These datasets improve model flexibility and durability. Hybrid model-based and appearance-based monitoring offers improved versatility in many scenarios. It ensures perfect tracking independent of brightness, facial variances, or tiny head motions. CNN-based tracking is useful in uncontrolled situations because to its adaptability. Due to extrinsic variables, traditional eye-tracking devices often fail in these situations.

Another crucial lesson from this research is the accuracy-computing burden trade-off. CNN of Haar and CNN of Mesh both had high accuracy, but CNN of Mesh had a twelve-second load time. This signals stricter computing needs. This suggests that CNN-based models may be less efficient for real-time deployment on lower-end hardware, even if they perform better. Future advances in hardware acceleration and software optimization may help overcome this barrier, making these models more accessible. The study also shows that mesh-based methods like mesh of inter-pupil and mesh of split are effective but not as accurate or fast as CNN-based methods. This is the study result. These methods are less suitable for high-speed, precise applications due to their lower accuracy and longer processing times. They may be beneficial for examining eye movements offline for research, when processing efficiency is not a priority.

This work might build cost-effective and efficient eye-tracking technology, which is one of its biggest contributions. Business eye-tracking systems that employ infrared sensors or particular hardware are costly, limiting their price and accessibility. The researchers found that CNN models based on webcams may give equal accuracy and real-time monitoring without specific equipment. This discovery enables the widespread use of low-cost eye-tracking sensors in education, healthcare, and consumer technology. This research lays the framework for deep learning-based gaze estimation advancements. CNN-based models performed well in this study, thus future network topologies and dataset variety may improve accuracy and robustness. By using bigger and more diverse datasets, future research can ensure CNN models generalize across demographic groupings, eye shapes, and lighting conditions. Attention processing and reinforcement learning may also enhance gaze tracking, allowing for more precise glance direction prediction.

CNN-based models could be used in neurological and psychological research, where eye-tracking is crucial to understanding cognitive processes and diagnosing autism, ADHD, and neurodegenerative diseases. In such applications, the ability to precisely identify minute eye movement patterns may provide valuable insights on brain function and activity. Eye-tracking in accessibility solutions will benefit greatly from these discoveries. CNN-based models may improve communication and assistive equipment control for people with physical disabilities who use eye-tracking. These technology advances may increase tracking reliability, improving motor impaired people's quality of life and independence.

Driver monitoring systems are another promising automobile safety use. In view of the increased interest in driver distractions and exhaustion, accurate and real-time eye-tracking might avoid road accidents. CNN-based in-car safety systems might alert drivers to

fatigue and distraction, improving road safety. Since CNNs can accurately and quickly determine gaze direction, this is achievable. However, this study's amazing accomplishment is not without challenges. Maintaining consistent performance in very dynamic environments where excessive head motions, occlusions (such as glasses or reflections), and fast gaze shifts may affect accuracy is a major challenge. Increased variance resilience should be the focus of future research. Multi-frame analysis, enhanced preprocessing, and hybrid sensor integration may achieve this. CNN of Mesh has a longer load time since it depends on processing power. Despite modern GPUs being able to handle deep learning-based eye-tracking, optimizing models for low-power devices like mobile phones, tablets, and embedded systems remains tough. Future research should focus on lightweight model designs and hardware optimization to make CNN-based eye tracking more accessible across computers and platforms.

6. RECOMMENDATION

Future studies should follow these suggestions:

1. Optimizing CNN-based eye-tracking models for low-power devices. Such gadgets include smartphones, tablets, and embedded systems. Future study should focus on this. Model pruning, quantization, and hardware acceleration can reduce computational labor without compromising accuracy.
2. Eye-tracking systems require extensive improvement to adapt various lighting conditions, head movements, occlusions (glasses, reflections), and eye shapes. Multi-frame analysis, attention techniques, and adaptive preprocessing improve real-world tracking performance.
3. To train CNN models, expand the dataset to include diverse demographics, such as age groups, nationalities, and medical conditions. Thus, model generalization will increase and the system's performance will satisfy a wide variety of users.
4. Combining CNN-based eye tracking with other modalities like infrared sensors, accelerometers, or depth cameras may improve accuracy and resilience in challenging settings. Hybrid sensors can reduce occlusion and fast motion issues.
5. Refine and test CNN-based models to build assistive technology applications, especially for persons with mobility or communication challenges. Eye-tracking technologies in speech-generating devices, smart home controls, and accessibility aids can increase user flexibility.
6. Real-time deployment of eye-tracking models in driver monitoring systems for accident prevention, fatigue detection, and medical diagnostics is vital. Future research should focus on effective implementation. Continuous eye movement tracking may indicate cognitive deterioration, neurological illnesses, and mental fatigue. This might benefit healthcare and road safety.

REFERENCES

[1] Adnan M, Sardaraz M, Tahir M, Dar MN, Alduailij M, Alduailij M. A robust framework for real-time iris landmarks detection using deep learning. Appl Sci. 2022;12[11]:2022.
[2] Ahmed M, Laskar RH. Eye center localization using gradient and intensity information under uncontrolled environment. Multimed Tools Appl. 2022;81[5]:7145–7168.
[3] Ahmad N, Laskar RH, Hossain A, Ahmed M. Precise eye center localization in a practical environment. In: IEEE Region 10 Annual International Conference Proceedings/TENCON 2021. p. 533–538.
[4] Mou J, Shin D. Effects of social popularity and time scarcity on online consumer behaviour regarding smart healthcare products: an eye-tracking approach. Comput Hum Behav. 2018; 78:74–89.
[5] Kumar D, Sharma A. Electrooculogram-based virtual reality game control using blink detection and gaze calibration. In: 2016 International Conference on Advances in Computing, Communications and Informatics [ICACCI]. 2016. p. 2358–2362.
[6] Pastel S, Chen CH, Martin L, Naujoks M, Petri K, Witte K. Comparison of gaze accuracy and precision in real-world and virtual reality. Virtual Real. 2021;25:175–189.
[7] Ansari MF, Kasprowski P, Obetkal M. Gaze tracking using an unmodified web camera and convolutional neural network. Appl Sci. 2021;11[19]:2021.

[8] Farnsworth B. Eye tracker prices [Internet]. 2019 [cited 2025 Mar 13]. Available from: https://imotions.com/blog/eyetracker-prices/
[9] Yiu YH, Aboulatta M, Raiser T, Ophey L, Flanagin VL, Eulenburg P, et al. DeepVOG: open-source pupil segmentation and gaze estimation in neuroscience using deep learning. J Neurosci Methods. 2019;324:108307. https://doi.org/10.1016/j.jneumeth
[10] Meng C, Zhao X. Webcam-based eye movement analysis using CNN. IEEE Access. 2017;5:19581–19587.
[11] Sattar H, Fritz M, Bulling A. Deep gaze pooling: inferring and visually decoding search intents from human gaze fixations. Neurocomputing. 2020;387:369–382.
[12] Cheng Y, Zhang X, Lu F, Sato Y. Gaze estimation by exploring two-eye asymmetry. IEEE Trans Image Process. 2020;29:5259–5272.
[13] Ahmed M, Laskar RH. Evaluation of accurate iris center and eye corner localization method in a facial image for gaze estimation. Multimed Syst. 2021;27:1–20.
[14] Valtakari NV, Hooge ITC, Viktorsson C, Nyström P, Falck-Ytter T, Hessels RS. Eye tracking in human interaction: possibilities and limitations. Behav Res Methods. 2021;53:1–17.
[15] Zhuang Y, Zhang Y, Zhao H. Appearance-based gaze estimation using separable convolution neural networks. In: 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference [IAEAC]. 2021. p. 609–612.
[16] Ahmed M, Laskar RH. Evaluation of accurate iris center and eye corner localization method in a facial image for gaze estimation. Multimed Syst. 2021;27[3]:429–448.
[17] Cheng Y, Wang H, Bao Y, Lu F. Appearance-based gaze estimation with deep learning: a review and benchmark [Internet]. 2021 [cited 2025 Mar 13]. Available from: http://arxiv.org/abs/2104.12668
[18] Deng J, Guo J, Zhou Y, Yu J, Kotsia I, Zafeiriou S. RetinaFace: single-stage dense face localisation in the wild [Internet]. 2019 [cited 2025 Mar 13]. Available from: http://arxiv.org/abs/1905.00641
[19] Chen Y, Song L, Hu Y, He R. Adversarial occlusion-aware face detection. In: 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems [BTAS]. 2018.
[20] Dwisnanto Putro M, Nguyen DL, Jo KH. Fast eye detector using CPU-based lightweight convolutional neural network. In: International Conference on Control, Automation and Systems. 2020 Oct. p. 12–16.
[21] Ahmed NY. Real-time accurate eye center localization for low-resolution grayscale images. J Real-Time Image Process. 2021;18[1]:193–220.
[22] Leo M, Cazzato D, De Marco T, Distante C. Unsupervised approach for the accurate localization of the pupils in near-frontal facial images. J Electron Imaging. 2013;22[3]:033033.
[23] Wang N, Gao X, Tao D, Yang H, Li X. Facial feature point detection: a comprehensive survey. Neurocomputing. 2018;275:50–65.
[24] Ahmed M, Laskar RH. Eye center localization in a facial image based on geometric shapes of iris and eyelid under natural variability. Image Vis Comput. 2019;88:52–66.
[25] Ahmed M, Laskar RH. Eye detection and localization in a facial image based on partial geometric shape of iris and eyelid under practical scenarios. J Electron Imaging. 2019;28[3]:1.
[26] Xia Y, Lou J, Dong J, Qi L, Li G, Yu H. Hybrid regression and isophote curvature for accurate eye centre localization. Multimed Tools Appl. 2020;79[1]:805–824.
[27] Abbasi M, Khosravi MR. A robust and accurate particle filter-based pupil detection method for big datasets of eye video. J Grid Comput. 2020;18[2]:305–325.

[28] Choi JH, Lee KI, Song BC. Eye pupil localization algorithm using convolutional neural networks. Multimed Tools Appl. 2020;79[43–44]:32563–32574.
[29] Liu ZT, Jiang CS, Li SH, Wu M, Cao WH, Hao M. Eye state detection based on weight binarization convolution neural network and transfer learning. Appl Soft Comput. 2021;109:107565. https://doi.org/10.1016/j.asoc.2021.107565
[30] Ahmad N, Yadav KS, Ahmed M, Hussain Laskar R, Hossain A. An integrated approach for eye centre localization using deep networks and rectangular-intensity-gradient technique. J King Saud Univ Comput Inf Sci. 2022;34[9]:7153–7167.
[31] Sun Y, Wang X, Tang X. Deep convolutional network cascade for facial point detection. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2013;3476-83.
[32] Zhou E, Fan H, Cao Z, Jiang Y, Yin Q. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. Proc IEEE Int Conf Comput Vis. 2013;386-91.
[33] Chandran P, Bradley D, Gross M, Beeler T. Attention-driven cropping for very high-resolution facial landmark detection. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2020;5860-69.
[34] Zhang K, Zhang Z, Li Z, Qiao Y. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett. 2016;23[10]:1499-503.
[35] Feng ZH, Kittler J, Awais M, Huber P, Wu XJ. Wing loss for robust facial landmark localisation with convolutional neural networks. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2018;2235-45.
[36] Choi JH, Lee KI, Kim YC, Song BC. Accurate eye pupil localization using heterogeneous CNN models. 2019 IEEE Int Conf Image Process [ICIP]. 2019;2179-83.
[37] Lee KI, Jeon JH, Song BC. Deep learning-based pupil center detection for fast and accurate eye tracking system. Lect Notes Comput Sci [LNCS]. 2020;12364:36-52.
[38] Ablavatski A, Vakunov A, Grishchenko I, Raveendran K, Zhdanovich M. Real-time pupil tracking from monocular video for digital puppetry. 2020;4-7. Available from: http://arxiv.org/abs/2006.11341
[39] Ogino Y, Toizumi T, Tsukada M. Fast eye detector using Siamese network for NIR partial face images. arXiv:2202.10671v2 [Cs.CV]. 2023 Jan 4. Available from: http://arxiv.org/abs/2202.10671
[40] Bazarevsky V, Kartynnik Y, Vakunov A, Raveendran K, Grundmann M. BlazeFace: Sub-millisecond neural face detection on mobile GPUs. CVPR Workshop Comput Vis Augment Virtual Real. 2019;3-6.
[41] Abdullah RM, Alazawi SAH, Ehkan P. SAS-HRM: Secure Authentication System for Human Resource Management Reem. Al-Mustansiriyah J Sci. 2023;34[3]:64-71.
[42] Viola P, Jones M. Robust real-time face detection. Int J Comput Vis. 2004;57[2]:137-54.
[43] Vitek M, Rot P, Štruc V, Peer P. A comprehensive investigation into sclera biometrics: A novel dataset and performance study. Neural Comput Appl. 2020;32[24]:17941-55.
[44] Rot P, Vitek M, Grm K, Emeršič Ž, Peer P, Štruc V. Deep sclera segmentation and recognition. In: Advances in Computer Vision and Pattern Recognition. 2020.
[45] Rot P, Emersic Z, Struc V, Peer P. Deep multi-class eye segmentation for ocular biometrics. In: 2018 IEEE International Work Conference on Bioinspired Intelligence, IWOBI 2018 – Proceedings; 2018. p. 1-8.
[46] Ali Z, Park U, Nang J, Park JS, Hong T, Park S. Periocular recognition using uMLBP and attribute features. KSII Trans Internet Inf Syst. 2017;11[12]:6133-51.
[47] Zhu Y, Zabaras N. Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. J Comput Phys. 2018;366:415–47.
[48] Zhao Z, Zheng P, Xu S, Wu X. Object detection with deep learning: A review. IEEE Trans Neural Netw Learn Syst. 2019;PP:1–21.

[49] Al-Sabban WH. Real-time driver drowsiness detection system using Dlib based on driver eye/mouth monitoring technology. Commun Math Appl. 2022;13[2]:807–22.
[50] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2001. p. 1.
[51] Kamarudin N, Jumadi NA, Mun NL, Keat NC, Ching AHK, Mahmud WMHW, et al. Implementation of Haar cascade classifier and eye aspect ratio for driver drowsiness detection using Raspberry Pi. Univ J Electr Electron Eng. 2019;6[5]:67–75.
[52] Abdullah RM, Alazawi SAH, Ehkan P. SAS-HRM: Secure Authentication System for Human Resource Management Reem. Al-Mustansiriyah J Sci. 2023;34[3]:64-71.
[53] Viola P, Jones M. Robust real-time face detection. Int J Comput Vis. 2004;57[2]:137-54.
[54] Rakhmatulin I, Duchowski AT. Deep neural networks for low-cost eye tracking. Proc Comput Sci. 2020;176:685-94.
[55] Roesler O, Kothare H, Burke W, Neumann M, Liscombe J, Cornish A, et al. Exploring facial metric normalization for within- and between-subject comparisons in a multimodal health monitoring agent. In: ACM International Conference Proceeding Series. 2022. p. 160-5.
[56] Aman, Sangal AD. Drowsy alarm system based on face landmarks detection using MediaPipe FaceMesh. In: Proceedings of First International Conference on Computational Electronics for Wireless Communications; 2021 June 11-12; Haryana, India. Berlin/Heidelberg, Germany: Springer; 2022. p. 363-75.
[57] Albadawi Y, AlRedhaei A, Takruri M. Real-time machine learning-based driver drowsiness detection using visual features. J Imaging. 2023;9[5]:1-18.
[58] Tonsen M, Zhang X, Sugano Y, Bulling A. Labelled pupils in the wild: A dataset for studying pupil detection in unconstrained environments. In: Eye Tracking Research and Applications Symposium [ETRA]; 2016. Vol. 14. p. 139-42.
[59] Kartynnik Y, Ablavatski A, Grishchenko I, Grundmann M. Real-time facial surface geometry from monocular video on mobile GPUs. 2019 Jul 16 [cited 2022 May 20]. Available from: http://arxiv.org/abs/1907.06724
[60] Datahacker. How to detect eye blinking in videos using dlib and OpenCV in Python [Internet]. 2022 May 20 [cited 2022 May 20]. Available from: https://datahacker.rs/011-how-to-detect-eye-blinking-in-videos-using-dlib-and-opencv-in-python/
[61] Rosebrock A. Eye blink detection with OpenCV, Python, and dlib [Internet]. 2017 Apr 24 [cited 2022 May 7]. Available from: https://pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/
[62] Soukupova T, Cech J. Real-time eye blink detection using facial landmarks. Res Rep CMP, Czech Technical University, Prague. 2016;5:1-8.
[63] Jiang Z, Li H, Liu L, Men A, Wang H. A switched view of Retinex: Deep self-regularized low-light image enhancement. Neurocomputing. 2021;454:361-72.
[64] Khassaf NM, Shaker SH. Image retrieval based on convolutional neural network. Al-Mustansiriyah J Sci. 2020;31[4]:43-54.
[65] Al-Tai MH, Nema BM, Al-Sherbaz A. Deep learning for fake news detection: Literature review. Al-Mustansiriyah J Sci. 2023;34[2]:70-81.
[66] Kanade P, David F, Kanade S. Convolutional neural networks [CNN]-based eye-gaze tracking system using machine learning algorithm. Eur J Electr Eng Comput Sci. 2021;5[2]:36-40.
[67] Akinyelu AA, Blignaut P. Convolutional neural network-based technique for gaze estimation on mobile devices. Front Artif Intell. 2022;4:1-11.

[68] Ou WL, Kuo TL, Chang CC, Fan CP. Deep-learning-based pupil center detection and tracking technology for visible-light wearable gaze tracking devices. Appl Sci. 2021;11[2]:851.

Modi N, Singh J. Real-time camera-based eye gaze tracking using convolutional neural network: A case study on social media website. Virtual Real. 2022;26[4]:1489–506.