WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Gaze Detection – A Better Way to Type

Mohammad Sufiyaan Khan[1], A. Akhil[2], R. Vaishnavi[3], Y. Srija[4], P. Bhavani[5], Kaparthi Sriram[6]

[1,2,3,4] B. Tech School of Engineering, Computer Science – AI&ML Malla Reddy University, India.
[3]Guide, Asst Professor School of Engineering, Computer Science – AI&ML Malla Reddy University, India.
DOI: https://doi.org/10.55248/gengpi.5.0524.1117

ABSTRACT:

The research provides a hands-free text input method using facial landmarks and blinking analysis. It uses OpenCV and Mediapipe to process webcam frames in real time and detect facial landmarks. By analyzing the blinking ratio, the system interprets blinks as character selections on a virtual keyboard. The text input is constantly updated on a text board, enabling for hands- free typing. Deep learning increases the system's stability and usability for consumers. Eventually, the code provides a simple yet efficient solution for hands-free text input by combining computer vision and deep learning approaches.

Keywords: *Gaze Detection, Eye Tracking, Virtual Keyboard, Blink Detection, Computer Vision, Deep Learning, OpenCV, Mediapipe, Convolutional Neural Networks (CNN), Machine Learning, Real-time Processing, Interactive Interfaces, Augmented Reality (AR), Virtual Reality (VR).*

1. INTRODUCTION

Human-computer interaction methods often fall short in meeting the needs of individuals with motor disabilities, posing substantial accessibility obstacles. Overcoming these challenges demands pioneering solutions that introduce intuitive interaction modalities. Beyond addressing accessibility challenges, the envisioned system holds potential for revolutionizing interaction paradigms across various domains, including virtual reality, gaming, market research, and user experience analytics. This project endeavors to create a real-time gaze detection system by harnessing MediaPipe and OpenCV and face landmarks detection models.

The envisioned project seeks to tackle these accessibility challenges head-on by developing a real- time gaze detection system. Leveraging cutting-edge technologies such as MediaPipe and advanced face

landmarks detection models, this system aims to track users' eye movements accurately and in real time. By analyzing the direction of a user's gaze, the system can interpret their intentions and translate them into meaningful interactions with digital interfaces. This approach offers a promising avenue for individuals with motor disabilities to navigate and interact with

technology more effectively, opening up new opportunities for inclusion and participation in various aspects of daily life.

Beyond its immediate impact on accessibility, the proposed gaze detection system holds vast potential for revolutionizing interaction paradigms across multiple domains. In virtual reality (VR) applications, for instance, the ability to control and manipulate virtual environments through gaze alone could enhance immersion and user engagement. Similarly, in gaming, integrating gaze-based input could introduce novel gameplay mechanics and accessibility features, making gaming experiences more inclusive and enjoyable for all players.

Furthermore, the suggested approach has potential applications in market research and user experience analytics. Researchers and analysts can acquire a better understanding of user behavior, preferences, and engagement levels by analyzing their gaze patterns while they interact with digital interfaces. This data-driven approach to studying human-computer interaction can help design and optimize user interfaces, resulting in more user-friendly and intuitive products and services.

2. LITERATURE REVIEW

Eye tracking systems are useful in a variety of applications, including assistive technology and online examination monitoring. Su Yeong Gwon et al. [2021] proposed a fuzzy-based approach for assessing gaze tracking accuracy. This system employed a variety of validation markers and display techniques, as well as the Min rule and Centre of Gravity defuzzification. The fuzzy technique showed a good association with actual gaze monitoring error, implying that it may increase system reliability in real-world applications. Future research could focus on determining its generalizability and broader application.

In their study "Automatic Gaze Analysis: A Survey of Deep Learning-based Approaches," Shreya Ghosh et al. examine gaze estimation and segmentation methods, focusing on unsupervised and weakly supervised approaches. They explore several techniques, such as CNN-based models, GazeNet, and Spatial Weight CNN, emphasising their benefits and stated evaluation measures. The authors argue that comprehensive gaze analysis must continue to address real-world issues such as uncontrolled settings and minimal supervision.

Another study, undertaken by Debajit Datta et al. [2022], examines the difficulty of eye gaze detection in online tests, a topic that emerged to significance during the 2020 pandemic. This study presented a system that combines face landmarks, computer vision, and Convolutional Neural Networks (CNNs), specifically using the AlexNet and VGG16 models.

Pier Luigi Mazzeo et al. investigated several CNN architectures for eye gaze estimation and prediction. They investigated gaze estimation and prediction utilising Long Short Term Memory (LSTM), Transformers with Self-Attention, and positional encoding. The designs were trained on a redesigned OPENEDS2020 dataset, which included a new structure called ResNext50 and two fully connected layers. Their proposed models have a smaller angular error than the state of the art.

Andronicus Akinyelu et al. offer a calibration-free Convolutional Neural Network (CNN) method for estimating gaze on mobile devices. To estimate gaze in unconstrained situations, the system uses full-face photos and a 39-point facial landmark component. This strategy seeks to improve the user experience by reducing the requirement for explicit calibration.

Sameer Rafee et al. created a low-cost system for eye movement analysis that employs Convolutional Neural Network (CNN) algorithms with the Kalman filter. This technique is intended for real-time applications and seeks to estimate and analyse eye position. The study demonstrates that their system accurately classifies and predicts eye movements while detecting pupil location, even in the presence of face tracking problems. The results show that this system is more efficient and effective than Recurrent Neural Network (RNN)-based techniques.

Anuradha Kar et al. examine advances in eye gaze estimating techniques and applications, focusing on typical use cases such as desktop, TV, head-mounted, automobile, and handheld devices. They identify platform-specific elements that influence gaze tracking accuracy and emphasise the importance of standardised methods for evaluating gaze tracking systems. To address this need, they present a methodological framework for consistent performance evaluation and comparison across several systems.

Mohamed Nasor et al. present a technique for controlling a computer screen cursor through eye movements, specifically iris monitoring. The device correlates the iris position to a matching place on the computer screen, allowing physically impaired people to move the cursor in any direction (left, right, up, down). The algorithm also contains a clicking function, which allows users to open and close folders, files, or apps, making computer interaction more accessible.

Garcia et al. created a real-time eye-gaze tracking system for applications such as driving that achieves high accuracy through precise pupil centre estimate. The system recovers from tracking failures by using cameras with narrow field of view (NFV) and wide field of view (WFV). Four synchronised infrared lamps provide corneal glints to help establish gaze direction. Calibration, hardware setup, and face detection algorithms allow for real-time processing and ±45º gaze tracking. The system's accuracy and robustness make it ideal for a variety of applications, including driving scenarios.

Fayeem Azeez et al. examine the many uses of Eye Gaze Tracking (EGT) systems, emphasising its importance as a bio-signal medium in Human-Machine Interface (HMI) development. The study discusses applications in diagnostics, authentication, therapy for autistic children, cursor control, eye typing, robotics, interactive software, driving assistance, cell phone operation, and hybrid interfaces. The authors emphasise EGT's adaptability in supporting persons with impairments and improving user-friendly interactions in a variety of settings, including healthcare and gaming. EGT is portrayed as a precise, cost-effective, and efficient HMI system that helps to advances in rehabilitation engineering.

Ayesha Shaikh et al. examine virtual keyboard technology, concentrating on human-computer interaction using image processing. The survey investigates numerous methods for touch detection and fingertip recognition, such as contour-based techniques and shadow analysis. The authors examine virtual keyboards used with webcams to provide low-cost, adaptable interfaces, emphasising the technology's potential to improve user experience across devices such as smartphones, tablets, and personal PCs.

Shaun P. Vecera et al. investigated the development of gaze detection in newborns and adults, with an emphasis on cortical processing in face recognition. The study included two trials, which revealed that newborns as young as four months can distinguish between direct and averted gaze, demonstrating that this capacity is not solely due to low-level visual processing. The study also found that individuals are more sensitive to gaze discrimination when their eyes are in an upright face context, as opposed to an inverted or scrambled face. These findings indicate that gaze detection may be mediated by cortical circuits involved in face recognition.

Kang Ryoung Park et al. study gaze detection with an infrared (IR) LED-based camera and Support Vector Machines (SVM). Their method detects where a user is looking on a monitor by computing 3D face features and measuring the normal vector of the plane defined by these characteristics. The study indicates that the approach has an RMS error of roughly 4.8 cm in accuracy between the computed and actual gaze positions. They compare their technique to previous gaze detection methods, highlighting disadvantages such as the requirement for manual depth measuring and high computation times due to sophisticated algorithms.

By addressing these concerns, the authors hope to create a more dependable and efficient method for facial and ocular gaze recognition

2.1 Existing System:

Existing systems for people with disabilities include a variety of technologies aimed at providing alternate input ways and increasing accessibility. These include eye- tracking systems such as the Tobii Eye Tracker, which allows users to control computer interfaces using their eyes, as well as assistive communication devices. Assistive communication technologies, such as the Dyna Vox Eye Max, can be pricey and may not fulfil all users' needs. Brain-computer interfaces (BCIs) may have limited compatibility and require substantial training to operate well. Existing solutions may not fully meet the different needs and preferences of people with disabilities, emphasizing the continual need for innovation and development in assistive technology.

2.2 Limitations:

Despite their ability to provide accessible connection with digital devices and facilitate productive typing sessions, these systems have significant limitations, including:

- In order to activate the system and run the application, the disabled user must be accompanied by a companion.
- A time-consuming setup procedure.
- Compatibility concerns with various devices or systems.
- Long-term use may cause eye fatigue and discomfort.

2.3 Proposed System:

The proposed system is a gaze-controlled keyboard integrated with blink detection, designed to offer an accessible and hands-free text input method for individuals with physical impairments or limitations. Using a webcam, the system tracks the user's facial landmarks, particularly focusing on eye movements to determine the direction of gaze.

Concurrently, a blink detection module monitors the user's eye regions for blink events, which serve as triggers for specific actions such as selecting keyboard keys or submitting typed text. The virtual keyboard interface, displayed on-screen, allows users to navigate through rows and columns using gaze control, with selected keys highlighted visually for feedback. Typed characters are instantly displayed in a separate area of the interface, providing real-time feedback to the user.

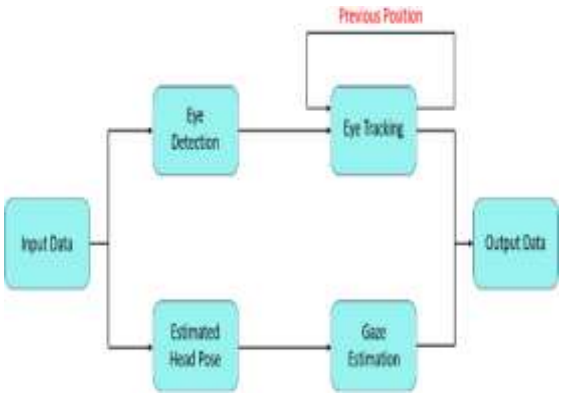

Figure 1: Working of Gaze Detection Model

3. PROBLEM STATEMENT

Current human-computer interaction methods lack intuitive interfaces for users with motor disabilities, necessitating innovative solutions.

The abstracted project aims to address this gap by developing a real-time gaze detection system using TensorFlow.js and face landmarks detection models. The system aims to predict gaze direction accurately, facilitating seamless interaction with virtual keyboards and addressing accessibility challenges in user interfaces.

Traditional human-computer interaction methods often fail to meet the needs of individuals with motor disabilities. To bridge this gap, a project is underway to create a real-time gaze detection system. Leveraging Deep Learning techniques and OpenCV and advanced face landmarks detection models, the system accurately predicts users' gaze directions, enabling seamless interaction with virtual keyboards and tackling accessibility barriers in user interfaces. This innovation promises to enhance inclusivity and revolutionize how people with motor disabilities engage with technology.

3.1 Description of Data:

The 68 facial landmark coordinates are a collection of particular places on the human face used to identify and define major facial traits. These points are widely utilized in computer vision, facial recognition, and analysis applications such as face alignment, emotion detection, facial expression recognition, and eye gaze tracking.

Figure 2: Face shape predictor with 68 points face landmarks

The 68 facial landmark coordinates are a set of essential sites used to map human facial features. The first 17 landmarks follow the jawline from ear to ear. The brows have five points individually, which define their contour and arch. The nose's bridge, nostrils, and tip are all outlined with nine points. Each eye contains six points that outline the top and lower eyelids. The mouth is represented by 20 landmarks that define the outer and interior form of the lips. The remainder of the landmarks highlight various facial features, such as the sides of the nose and the philtrum. These 68 coordinates form a comprehensive map of facial structure that can be used in a variety of applications, including facial recognition, eye gaze tracking, and emotion detection.

4. METHODOLOGY

The methods used for accomplishment of eye controlled virtual keyboard is shown in Fig. 1.

A. Video Capture and Grayscale Conversion

To start the eye-controlled virtual keyboard, open a connection to the webcam using cv2.VideoCapture(0). The program records video frames from the camera in a continuous loop. To improve speed, the collected frames are scaled to half of their original dimensions with *cv2.resize(frame, None, fx=0.5, fy=0.5).* After capturing the frame, it is converted to grayscale with *cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY).* This conversion makes subsequent processing easier, such as face detection, by lowering the computing load associated with color data.

B. Face detection and landmark extraction

The next stage is to detect faces in the collected grayscale frames. This is accomplished using Dlib's face detector, which is initialized using *dlib.get_ frontal_face_detector().*

After detecting faces, the program utilizes *dlib.shape_predictor()* to extract 68 facial landmarks for each one. These landmarks reflect important facial characteristics including the eyes, nose, mouth, and jawline. These face landmarks are required for subsequent processing, including eye cropping and blink detection.

Figure 3: Face detection and landmark extraction

C. Virtual Keyboard Initialization

The virtual keyboard is generated with a np.zeros array that is large enough to support the keyboard layout. The keyboard has dimensions of 600x1200 pixels and employs predetermined keys, each with a fixed width and height. Along with the virtual keyboard, a text board is initialized to show the text generated by the virtual keyboard input.

D. Virtual Keyboard Rendering

The program uses a key-switching mechanism to choose keys from the virtual keyboard at regular intervals. It indicates which key is being changed to help the user navigate visually. The letter function is used to draw keys on the virtual keyboard at precise positions. The function also has a highlighting mechanism that distinguishes the current key from the others. This switching process simulates the virtual keyboard's key rotation behaviour, allowing users to select keys using eye blinks.

E. Eye Blink Detection

To detect eye blinks, the program computes the eye blinking ratio, which is the ratio of the horizontal and vertical distances of specified facial landmarks that represent the eyes. The function *get_blinking_ratio* calculates this ratio to determine whether a blink occurred. A valid blink occurs when the blinking ratio surpasses a predetermined threshold for a specific amount of frames. When a valid blink is detected, the highlighted key is added to the text string to simulate keyboard input. The blinking frame count is reset after inserting the key to guarantee that the operation can be repeated.

F. Display and Exit

The program displays the webcam frame, virtual keyboard, and text board in separate OpenCV windows, providing a visual interface for the user. The loop continues until the user presses the ESC key, detected with *cv2.waitKey(10).* Upon exiting, the program releases the webcam with *cap.release()* and closes all OpenCV windows using *cv2.destroyAllWindows().* This clean-up ensures that resources are properly released, preventing any memory leaks or hanging processes.

5. DESIGN

The proposed structure consists of numerous interconnected modules, each of which performs a distinct function in the text input process. Initially, the system uses a powerful facial detection algorithm to find and track the user's face within the camera frame. Next, face landmark localization techniques are used to pinpoint significant characteristics, allowing for precise eye tracking and analysis.

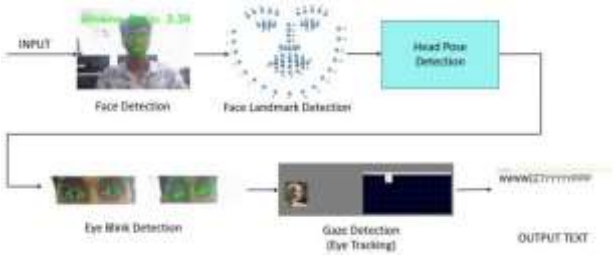

Figure 4: Architecture

The system's main capability is blink detection, which monitors the user's eye movements to infer blinking patterns. By analyzing the ratio of key facial landmarks connected with the eyes, the system can accurately detect blinks and distinguish them from other face motions.

The virtual keyboard interface is a critical component of the architecture, providing users with a selection of letters and symbols for text input. Using the detected eye movements, users may navigate and interact with the virtual keyboard, picking characters using natural gaze- based interactions.

A dynamic text board is built into the architecture to give users real-time feedback and make text authoring easier. This text board shows the accumulated text input, which updates in real time as users pick characters from the virtual keyboard. The text board acts as a visual help, increasing user interest and delivering useful feedback during the text entry process.

6. EXPERIMENTAL RESULTS

The system's main capability is blink detection, which monitors the user's eye movements to infer blinking patterns. By analyzing the ratio of key facial landmarks connected with the eyes, the system can accurately detect blinks and distinguish them from other face motions.

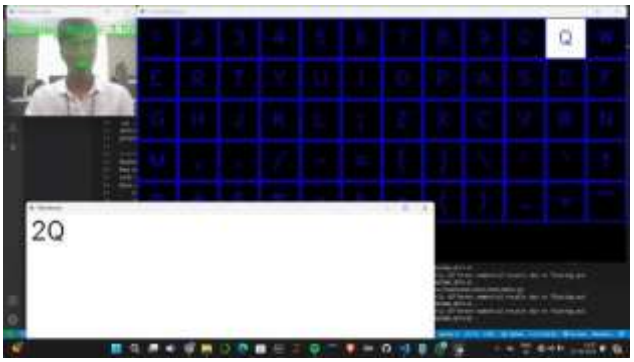

Figure 5: Output

The virtual keyboard interface is a critical component of the architecture, providing users with a selection of letters and symbols for text input. Using the detected eye movements, users may navigate and interact with the virtual keyboard, picking characters using natural gaze- based interactions.

A dynamic text board is built into the architecture to give users real-time feedback and make text authoring easier. This text board shows the accumulated text input, which updates in real time as users pick characters from the virtual keyboard. The text board acts as a visual help, increasing user interest and delivering useful feedback during the text entry process.

6.1 EVALUATION METRICS:

A. Confusion matrix:

The confusion matrix provides a thorough evaluation of a blink detection system's performance, emphasizing its ability to discern between blinks and non-blinks. However, it outperforms blinks in terms of correctly identifying non-blinks, which could be attributed to dataset imbalances. Furthermore, the system produces more false positives than false negatives, indicating that it detects blinks incorrectly. This insight is critical for improving the system's accuracy and dependability, and hence increasing its usefulness in real-world circumstances.

These findings highlight the significance of improving the system's algorithms and parameters in order to achieve more balanced and precise performance.

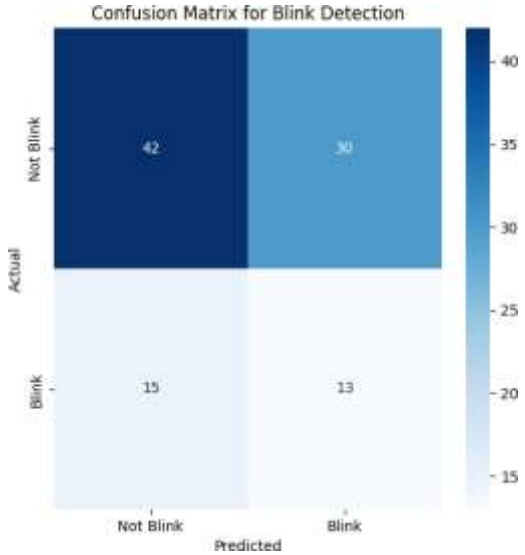

Figure 6: Confusion Matrix

B. Blinking Ratio Distribution

The graph shows the blinking ratio distribution, highlighting outliers and the range of ratios. It aids in determining a suitable blink detection threshold by revealing common blinking patterns and identifying outliers.

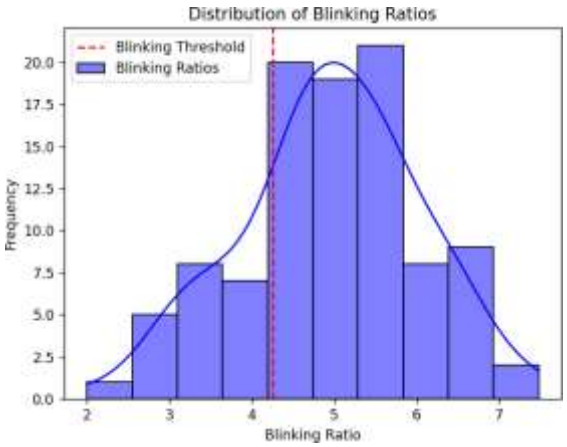

Figure 7: Blinking ratios distribution

The histogram displays the distribution of blinking ratios, with the ratios on the x-axis and frequency on the y-axis. The majority of ratios fall between 2 and 5, with only a few exceeding this range. A blinking threshold of 4 identifies ratios above it as blinks. The appropriateness of this threshold is determined by the application's goal: for accurate blink tracking, a lower threshold that captures all blinks may be desirable, whereas a higher threshold may be used to remove phenomena such as eye twitches.

7. CONCLUSION

The study on gaze detection for hands-free text input provides a unique approach for typing without physical input devices by detecting facial landmarks and analyzing blinking patterns. The system uses OpenCV and MediaPipe to correctly track users' eye movements in real time, interpreting blinks as inputs on a virtual keyboard. This concept improves accessibility for people with motor limitations by providing real-time feedback via a virtual keyboard interface and a text board. The system's feasibility and effectiveness are proved through experimental results, which highlight prospective applications in assistive technology, virtual reality, gaming, and user experience analytics.

In conclusion, the proposed gaze detection system provides a simple yet effective solution for hands-free text input, addressing accessibility issues while also opening up possibilities for future study and improvements.

8. FUTURE ENHANCEMENTS

The study on gaze detection for hands-free text input provides a unique approach for typing without physical input devices by detecting facial landmarks and analyzing blinking patterns. The system uses OpenCV and MediaPipe to correctly track users' eye movements in real time, interpreting blinks as selections on a virtual keyboard. This concept improves accessibility for people with motor limitations by providing real-time feedback via a virtual keyboard interface and a text board.

Future developments could concentrate on improving the accuracy and reliability of the gaze detection system. This could include optimizing algorithms for facial landmark detection and blink analysis, as well as adding machine learning approaches to adapt to specific user behaviors and environmental situations. In addition, the virtual keyboard interface can be optimized to shorten the time it takes to write text. Predictive text input, auto-completion suggestions, and clever word prediction algorithms could all help speed up typing.

Automatic text saving functionality might also be incorporated to ensure that entered text is constantly saved and stored, lowering the chance of data loss due to unexpected interruptions or system crashes. By incorporating these future upgrades, the gaze detection system will be more effective, efficient, and versatile, catering to a broader range of users and applications across multiple domains.

To make the system useful for AR/VR games, improvements could include integrating the gaze detection system into present game engines and platforms. This would enable users to interact with virtual worlds and control game operations using their gaze, resulting in a more immersive and accessible game playing experience. Furthermore, subsequent versions of the system may offer advanced functionality such as launching and closing programs simply via gaze input, removing the need for help from another person.

9. REFERENCES

[1] Basnetrikesh. (n.d.-b). GitHub - basnetrikesh/eye_ controlled_keyboard . GitHub. https://github .com/basnetrikesh/eye_controlled_keyboard

[2] Swethakoyyyana. (n.d.). GitHub – swethakoyyyana /eye-blink-controlled-keyboard: Eye blink controlled keyboard. GitHub. https://github.com/swethakoyyy ana/ eye-blink-controlled-keyboard

[3] Serengil, S. (2022, January 20). Facial Landmarks for Face Recognition with Dlib - Sefik Ilkin Serengil. Sefik Ilkin Serengil. https://sefiks. com/2020/11/20/f acial -landmarks-for-face-recognition-with-dlib/

[4] Gwon, S.Y.,Cho,C.W.,Lee,H.C.,Lee,W.O., & Park, K. R. (2013). Robust eye andpupil detection method for gaze tracking. International Journal of Advanced Robotic Systems, 10(2), 98. https://doi.org/10.5772/55520

[5] Lee, T., Kim, S., Kim, T., Kim, J., & Lee, H. (2022). Virtual keyboards with Real-Time and robust Deep -Based gesture recognition. IEEE Transactions on Human-machine Systems, 52(4), 725–735. https:// doi.org/10.1109/thms.2022.3165165.

[6] Nasor, M.,Rahman, K.K.M.,Zubair, M. M.,Ansari, H.,&Mohamed,F.(n.d.).Eye-controlled mouse cursor for physically disabled individual. 2018 Advances in in Science and Engineering Technology International Conferences (ASET). https://doi.org/10.1109/icaset.2 018.8376907

[7] Akinyelu, A. A.,&Blignaut,P.(2022).Convolutional neural Network-Based technique for gaze estimation on mobile devices. Frontiers in Artificial Intelligence, 4. https://doi.org/10.3389/frai.2021.796825

[8] Automatic Gaze Analysis: A Survey of Deep Learning based Approaches. (n.d.). In arXiv:2108 . 05479v3 [cs.CV] 21 Jul 2022 [Journal-article].

[9] Shaikh, A., Kanade, A., Fernandes, M., Chikhalthane, S., & Computer Department, Modern Education Society's College of Engineering, and Pune. (2015). Review of virtual keyboard [Research Article]. International Journal of Engineering and Techniques, 1(5), 75. http://www.ijetjournal.org

[10] Vecera, S. P., & Johnson, M. H. (1995). Gaze detection and the cortical processing of faces: Evidence from infants and adults. Visual Cognition, 2(1), 59–87. https://doi.org/10.1080/1350628950840 1722

[11] Park, K. R., Lee, S. H., & Kim, J. (2002). Facial and eye gaze detection. In Lecture notes in computer science (pp. 368–376). https://doi.org/10.1007/3-540-36181-2_37

[12] Deep Learning based Eye gaze estimation and prediction. (2021, September 8). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.iee e.org/document/9566413

[13] Datta, D., Maurya, P. K., Srinivasan, K., Chang, C., Agarwal, R., Tuteja, I., & Vedula, V. (2021). Eye gaze detection based on computational visual perception and facial landmarks. Computers, Materials & Continua/Computers, Materials & Continua (Print), 68(2), 2545–2561. https://doi.org/10.32604/cmc.2021.015478

[14] Rafee, S., Xu, Y., Zhang, X., & Yemeni, Z. (2022). Eye-movement Analysis and Prediction using Deep Learning Techniques and Kalman Filter. International Journal of Advanced Computer Science and Applications/International Journal of Advanced Computer Science & Applications, 13(4). https://doi.org/10.14569/ijacsa.2022.01304107

[15] A review and analysis of Eye-Gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. (2017). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8003267

[16] Aziz, F., Mokhtar, N., Arof, H., Mubin, M., University of Malaya, & University Malaysia Pahang. (2014). Review of eye gaze tracking Application. In Conference Paper.

[17] Face detection guide. (n.d.). Google for Developers. https://developers.google.com/mediapipe/solutions/vision/face_detector

[18] OpenCV:OpenCV-Python Tutorials. (n.d.). https:/ /docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

[19] dlib. (2024, April 3). PyPI. https://pypi.org/projec t/dlib/

[20] Italojs.(n.d.).facial-landmarks-recognition/shape_predictor_68_face_landmarks.dat at master · italojs/facial-landmarks-recognition. GitHub. https://github.com/italojs/facial-landmarks-recognition/blob/master/shape_predictor_68_face_landmarks.dat