Received: 27 April 2025| Revised: 25 June 2025| Accepted: 11 July 2025| Published online: 27 August 2025

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2025, Vol. 00(00) 1–14
DOI: 10.47852/bonviewJCCE52026027

BON VIEW PUBLISHING

Multitask Virtual Keyboard Controlled by an Eye-Gaze-Based Intelligent Entry System

Amal Hameed Khaleel[1,*], Thekra Abbas[2], and Abdul-Wahab Sami Ibrahim[2]

[1] *Department of Computer Science, Basrah University, Iraq*

[2] *Department of Computer Science, Mustansiriyah University, Iraq*

Abstract: In modern society, computers and cell phones play an integral role in individuals' everyday lives. Individuals utilize messaging platforms such as WhatsApp, WeChat, and Facebook daily to communicate with others. Virtual keyboards are a significant tool in assistive technology because they aid individuals with significant motor disabilities in making effective contact with computers. This study proposes an eye-gaze-controlled virtual keyboard using a new method called Distance Eyelid-Iris-MediaPipe, which enables individuals with disabilities to write effectively. The suggested virtual keyboard is an affordable assistive device because it only necessitates a webcam for operation. The proposed virtual keyboard consists of six supplementary menus: two menus for English letters, one for numerical digits and mathematical operations, one for important text operations, one for Latin symbols, and an emoji menu. The suggested keyboard has special characteristics such as auditory feedback and visual highlighting of pressed keys. The testing of the proposed system received positive feedback from several users, and the empirical results of the proposed system were better than those of the previous models. It achieved this with an average typing rate of 18 characters per minute and 4 words per minute, and it had a NASA-TLX score of 10% and a system usability scale score of 93.

Keywords: eye gaze, eye-typing, human–computer interaction, computer vision, virtual keyboard

1. Introduction

Currently, worldwide, individuals communicate using portable electronic devices such as mobile phones and personal digital assistants, as well as nonportable desktop computers such as PCs and laptops [1].

According to Cook and Polgar, assistive technology (AT) is an all-encompassing tool set, service, method, and strategy collection that has been developed and is employed widely for helping individuals reduce the functional limitations caused by handicaps. One particular subset in the field is augmentative and alternative communication, which includes tools, programs, or machines that facilitate communication between individuals with a disability who have speech impairments [2].

Computer vision, as a discipline, focuses mainly on human–computer interaction (HCI) through visual perception. HCI is an emergent field. It encompasses various software programmers and strategies aimed at improving human–machine interfacing [3, 4].

The eye is an important part of the face. It communicates clear and valuable details regarding individuals. In addition to being a means of expression, the human eye intuitively interprets the communication and interaction of people, from which we can obtain information on the surrounding environment and respond appropriately [5].

Severe disabilities such as spinal cord injuries do not affect eye movement. Impairments therefore have a minimal effect on gaze control. This is why eye tracking represents an excellent opportunity for AT and enables innovative solutions in HCI [6].

Disabilities upset people. Hence, several researchers tried to develop eye-operated systems. In cases of profound disabilities that prevent conventional mouse and keyboard use, knowing the user's gaze direction can greatly help [2].

Eye-gaze technology is an important area of research today that has many earmarks. One of the most promising earmarks is eye-gaze-tracking technology, an auxiliary feature for inputting text on digital devices for those with impairments. This system employs input methods that are greatly similar to conventional text input technology, with the only distinction being that the human organ that directs is the eye and not the hand [7].

For people with severe speech and motor impairments, virtual keyboards or screens are often used as a means of augmentative communication. The layout and entry method are two important characteristics of any on-screen keyboard [8, 9].

Text entry by gaze is handy for individuals with severe disabilities, for whom eye movements may be the only means of communicating. Because the system is mainly intended for people with disabilities, all possible forms of disabilities have been taken into consideration. Therefore, it chose the eyes because of the following:

1) The eyes have a direct connection to the brain rather than the spinal cord, which is responsible for the majority of the motor impairments. Therefore, no portion of the disabled body was employed (i.e., head, hand, and leg motions) because only the eye can move.
2) Brain signals were not used because they needed to be linked to the user's body, which could lead to a psychological and physical burden and could cause discomfort. Therefore, it chose the eyes because it is an appearance-based method and does not require physical interaction with a device.

***Corresponding author:** Amal Hameed Khaleel, Department of Computer Science, Basrah University, Iraq. Email: amal.khaleel@uobasrah.edu.iq

3) Sound was not utilized because some people may suffer from dysarthria. Unlike eyesight, speech is sensitive to noise. In addition, word pronunciation may violate the user's confidentiality and privacy.

The proposed paper reports on a virtual keyboard that uses eye gaze for efficient text entry. The approach involves utilizing a webcam to track the user's gaze location to enable a nontouchable input method with fewer devices and lower prices. The main objective of this study is to reduce the amount of effort necessary for typing and to make it easier for those who are physically incapacitated and do not even have any control over their fingers but can communicate using their eyes.

The mainstream eye-typing system employs a camera to follow eye movements to enter information by directing their gaze toward the keyboard appearing on the screen. This process can be divided into two parts: an eye-tracking system as the initial part and a text input system as the second part. The purpose of the eye-tracking system is to locate eyes in images and then determine where the gaze lies simultaneously.

This study has three broad objectives in the design of a gaze-based text-entering interface:

1) Establishing a new method to reduce the amount of effort necessary for typing and to make it easier for those who are physically incapacitated. It is achieved by shortening the eye-typing duration and reducing the search time to increase the maximum typing speed.
2) This study was intended to find out how virtual keyboard design and feedback influenced the efficiency of text entry.
3) Predicting individual letters and entire words would improve responses and reduce errors.

The remainder of this paper is organized as follows. Section 2 begins with a concise review of relevant literature on virtual keyboards. Section 3 contains the methodology for the structure of the virtual keyboard. Section 4 evaluates the performance of the suggested algorithms and analyzes the findings using real data examples. Finally, Section 5 provides the conclusion and outlines the future study.

2. Literature Review

Individuals with profound impairments frequently have significant challenges and may be unable to input text using conventional keyboards. The recent advancements in gaze-controlled apps have enabled the use of an alternative text input system, which is now a thriving field of study.

Eye-typing has a significant and extensive history in the field of HCI research. It serves as a primary means of interaction for those who can only move their eyes [10].

The majority of the studies rely on dwell time, with shorter durations being more prone to false-positive selections, and longer thresholds result in increased waiting time for the user. Multiple research projects were undertaken to optimize the interface design of virtual keyboards to improve users' comfort and typing speed [11].

In prior studies, researchers often developed systems by utilizing pre-existing eye tracker devices, focusing solely on the design of text input systems. De Rosa et al. [12] proposed T18, a soft keyboard for smartwatches with a QWERTY layout and 18 buttons that can accommodate multiple characters. It employed three rows of six keys for each of the alphabet's 26 letters, with a fourth row for control characters (space and backspace). In addition, Islam et al. [13] proposed an eye-gaze-controlled virtual keyboard. The suggested keyboard has 40 keys, including the delete key, English letters, number digits, and Latin symbols. Each key lights up successively in the forward or backward direction. Eye gazing activates the key, and blinking is utilized to type it.

Jeevithashree et al. [14] suggested an eye-gaze-directed layout of the two-level English virtual keyboard optimized using a genetic algorithm as a static adaptation procedure. A dynamic adaptation procedure using a Markov-model-based technique that tracks user interactions and reduces dwell time came next.

Benabid Najjar et al. [15] proposed an optimization system for the arrangement of keys on an Arabic keyboard for applications that employ a single pointer input device. It used three methods for layout: common (QWERTY), genetic algorithm (GA), and simulated annealing (SA). Eye tracking was used to evaluate the usability of the optimized layouts.

Bharath et al. [16] suggested a system that gives an alternative option for those suffering from paralysis and physical infirmities by employing their facial expressions via a web camera as the fundamental input mechanism instead of a physically handled virtual keyboard and mouse. The proposed device operates by detecting facial expressions such as eyeball and lip movements and uses a Haar classifier to identify the area of the face, eyes, and mouth. Abhaya et al. [17] suggested a method involving a user-friendly keyboard interface for text that was developed to track the eyes' movement through OpenCV and the Dlib package. The system relies on deep-learning-based eye-tracking techniques that imply no need for calibration for the user. The system's gaze estimation models utilize convolutional neural networks to increase typing speed through a word prediction engine.

Cui et al. [18] described a technique called GlanceWriter, which offers a way to enter a sentence by looking at the letters of the input one by one without any overlapping, return, or stopping of any key. GlanceWriter is a dwell-free input technique that relies on a reverse crossing gesture to detect the beginning and end keys. The GlanceWriter system allows entering texts without pauses and overlaps using probabilistic analysis of eye movement trajectories.

Recently, Mifsud et al. [19] suggested a unique approach for modeling the user's dwell-free eye movement to create a robust dwell-free typing algorithm capable of determining the user's typing content. This study introduces three innovative HMM-based frameworks that are designed, evaluated, and tested for the real-time swipe typing application. In addition, Mallik et al. [20] offered a virtual keyboard that enables users to input characters using specific hand movements. This model utilizes machine learning technology, specifically convolutional neural networks with fully connected layers and an LSTM for improved hand motion recognition and sequential data processing. Furthermore, it identifies the precise coordinates of significant locations obtained using the MediaPipe hand detection method to detect tapping. Therefore, the goal of the proposed virtual keyboard system is to develop an eye-tracking-based text entry technique with condensed menus on a multimodal virtual keyboard. Therefore, the goal of the proposed virtual keyboard system is to develop an eye-tracking-based text entry technique with condensed menus on a multimodal virtual keyboard. This proposed virtual keyboard utilizes a webcam for input and serves as a cost-effective and user-friendly replacement alternative to traditional hardware keyboards.

The summary of the literature review for the past few years is shown in Table 1.

3. Research Methodology

The proposed keyboard system is split into two phases: the first phase is to control the keyboard, and the second phase is to type in the keyboard. In the first phase, determine the letter by determining the direction of the eye gaze that is detected using a proposed

Table 1
Summary of the literature survey

Ref.	Layout	Type input	Device type	Methods	Input key	WPM	CPM	Disadvantages
[12]	QWERTY	Single finger	No	---	English letters	15.7	---	-Cannot be used for gaze-controlled interfaces
[13]	Alphabetic	Eye gaze	Webcam	Dlib	English letters, numerical digits, and symbols	2.35	9.50	-All letters and numbers are in a single layout -Poor layout design -Can only use direction of gaze (left, right, and center)
[14]	2 different layouts	Eye gaze	Tobii SDK	Genetic algorithm	English letters	2.70	14.88	-It requires long-time user training
[15]	3 different layouts	Eye gaze	Tobii X120	Simulated annealing algorithm	Arabic letters	3–5	---	-Complex to use
[16]	QWERTY	Eye ball and mouth	Webcam	Haar classifier	English letters, numerical digits, and symbols	---	---	-Complex to use -All letters and numbers are in a single layout -Poor layout design
[17]	QWERTY	Eye gaze	Webcam	CNN algorithms and Dlib	English letters, numerical digits, and symbols	---	---	-All letters and numbers are in a single layout -Poor layout design
[18]	QWERTY	Eye gaze	Tobii Dynavox	Reverse crossing	English letters	10.89	---	-All letters are in a single layout -Poor layout design
[19]	QWERTY	Eye gaze	EOG	HMM-based frameworks	English letters	2.14–12.85	---	-Complex to use -Poor layout design
[20]	QWERTY	Hand gestures	Webcam	LSTM and MediaPipe	English letters	6.9	---	-Cannot be used for gaze-controlled interfaces -Poor layout design

engineering algorithm called Distance Eyelid-Iris-MediaPipe. The eye-closing gesture was used to fixate on the desired letter key, and the four eye movements (left/right/up/down) were used to move between keyboard keys. As for the second phase, a new keyboard design with an alphabetical layout helps in controlling it with the eye.

3.1. Virtual keyboard structure

The primary goal of creating a virtual keyboard is to provide a means for rapid text typing while maintaining a comfortable user experience [21]. The proposed study presents an eye-gaze virtual keyboard that may be utilized for text entry by blinking the eyes. It is controlled by eye gaze in four directions. This method is handy for impaired people and in circumstances when vocal and tactile input is not possible (e.g., in loud environments or when the hands are busy).

A multimodal virtual keyboard has been suggested, as shown in Figure 1. When the eye-keyboard was created, the focus was to employ the largest size of the window to be visible to the user, with an emphasis on ease of movement in all four directions. As a result, the keys are positioned as follows: three at the left, three at the right, two at the top, and two at the bottom. Experiments have shown that this design is best for moving the cursor easily on the keyboard by allowing the user to move their sight in four directions easily.

The keyboard is split into six major menus, each of which consists of two parts. The initial component (eye-keyboard) displays a command, with a total of 24 keys arranged in four lines and six columns to work equitably. The second component (eye-board) is an input text screen that allows the user to view the output text in real time. The size of the first component (eye-keyboard) is 1200 × 150, and the size of the second component (the eye-board) is 200 × 800. The size of each key is 200 × 200 in the shape of a rectangle because the keyboard's rectangular form is better for dealing with the four directions than the circular one.

It is crucial to immediately offer the user effective feedback on their command choices to prevent them from diverting their attention to the typing board (eye-board) to check its content. The user receives audio feedback as an acoustic beep after successfully executing a command to go from one menu to another. Furthermore, when selecting a letter, the corresponding key is visually emphasized, and the letter is audibly uttered. This auditory stimulus prompts individuals to be proactive, enabling them to anticipate the subsequent character.

The virtual keyboard operates on the idea of sequential illumination of keys based on gaze direction, where each key lights up individually. To achieve this, we changed the letter's background to a highlighted color while keeping the letter itself black. For example, if the letter "A" is illuminated at a certain moment and the eye blinks at that time, the letter "A" will be inputted and a sound will be emitted to indicate that a letter has been entered.

This paper proposes the use of change dwell time as the time required to transition from one key to another (1 s) to expedite the printing process, and the time needed to choose a key is 2 s to prevent the issue of Midas touch.

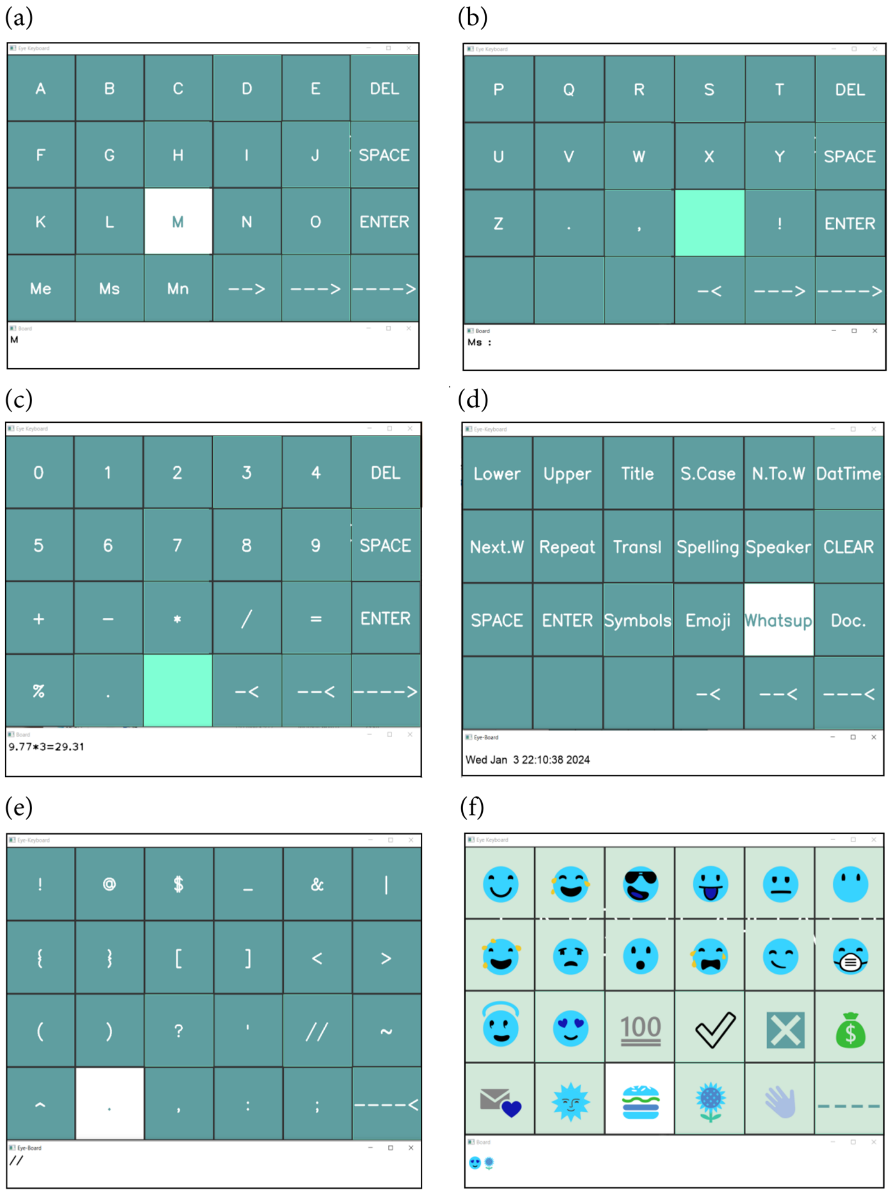
Figure 1 illustrates the layout of a virtual keyboard divided into six distinct sections:

(a) English alphabet A–O
(b) English alphabet P–Z
(c) Numeric keys 0–9
(d) Enhanced tools (e.g., copy, paste, and undo)
(e) Symbols (e.g., punctuation and special characters)
(f) Emojis

Each section visually groups related characters or functions, helping users quickly navigate between alphabets, numbers, tools, and expressive symbols:

a) The first menu has 15 keys representing the English alphabet (A–O) and three fundamental keys (space, enter, and delete), in addition to three buttons for anticipated words and three keys for navigating to the following menus.
b) The second menu has 15 keys representing the English alphabet (P–Z) and three fundamental buttons (space, enter, and delete). In addition, it includes three keys for anticipated words and three keys for navigating to the previous and subsequent menus.
c) The third menu comprises 10 numeric keys (0–9), 3 fundamental keys (space, enter, and delete), 5 keys for the four basic arithmetic operations (+, −, /, and *) the remainder of the division, 1 key for a comma for decimal numbers, 1 key for converting the number to binary, and 3 keys for navigating to the previous and subsequent menus.
d) The fourth menu consists of the essential tools used to enhance typing speed on virtual keyboards. These tools include the following: converting the letter case from lowercase to uppercase and vice versa, spelling, speaker, translation, and word prediction. In addition, print the current time, date, and day and store the text printed in a Word document or transmitted text using WhatsApp.

Figure 1
Visualization of the virtual keyboard



The aid and enhancement tools include the following:

- Lower: Make all printed text lowercase using (str. lower()).
- Upper: Make all printed text uppercase using (str. upper()).
- Title: Make the first letter of each word in the printed text uppercase using (str. title()).
- Swap Case (S. Case): Make the letters of the printed text change their state from large to small and vice versa using (str. swapcase()).
- Number to Word (N. To W.): Convert numbers (Digits) in text into words using the (num2words) library.
- Repeat: Repeat the text twice using (str*2).
- Speaker: Reading the text (converting words into sound) using (win32com.client) library.
- Spelling: Modifying the spelling of text words using the (spellchecker) library.
- DateTime (DatTime): Add time, date, and present-day using the (DateTime) library.
- Weather: Knowing the weather for a specific city by knowing the temperature and sky conditions. It can be run using the (pywhatkit) library. This is where the program is linked to the web via the Python library (requests) and specifies the browser link. After that, the city name is entered through the program to be linked to the browser to retrieve the required data, as shown in the following:
  url=”https://www.google.com/search?q=”+”weather”+city
  HTML = requests.get(url).content.
- WEB: Search for a word on the web using the (selenim. webdriver) library. It specifies the default browser (Chrome) in which the search is performed by instructions:

driver = webdriver.Chrome()
driver.get(“https://www.google.ps”).

- CLEAR: Clear the entire text.
- Next Word (Next.W): Predict the complementary word to the sentence using the (Gpt2) library and put the complementary words in three keys of the fourth row of the keyboard.
- Translation (Trans): Translating the text from English to Arabic using (googletrans) library.
- WhatsApp: Sending printed text via WhatsApp using the (pywhatkit) library, where the recipient’s number is typed and then the sent message is typed in a new line. As for the sending time, it can be specified in advance in the program or entered upon execution using the following Python instruction: pywhatkit. sendwhatmsg_instantly (num, text, time).
- Document (Doc.): Save text to be printed in a Word file using the (docx) library, where the name and path of the Word file are predetermined.

Figure 2 shows how the text would be saved in a Word document. The typed texts are added to it using a Python instruction (document. add_paragraph), and the text typed is saved in a Word document using (document. save).

e) The fifth menu consists of all 23 special symbols that are used on a standard keyboard.
f) The sixth option is dedicated to emojis that express the user’s emotional states, such as joy, sorrow, and others.

The layout optimization process for a virtual keyboard involves arranging the keys and decreasing the typing time to allow the user to type with maximum efficiency. The alphabetical layout was used to organize the keys on the keyboard because most people can know the location of the letters by memorizing the alphabet’s letter sequence.

3.2. Virtual keyboard architecture

Currently, several obstacles in the discipline of eye tracking must be overcome. The properties of the eye tracker limit the development of the text input system. To achieve precise eye tracking, it is necessary to have human-eye images with high resolution. This involves the use of a high-quality camera and the provision of optimum lighting conditions. Thus, calibrating an eye tracker requires a significant amount of time before its use. Throughout the calibration procedure, individuals are strictly prohibited from making any bodily movements because even little shifts might lead to inaccuracies in the final estimate. These requirements contribute to higher costs and restrict the range of suitable settings.

In addition, some people might show physiological signs of fatigue, such as drooping eyelids or a natural squint, which leads to inaccurate results [22].

To address the aforementioned issues and cater to the requirements of individuals with disabilities in the implementation of these systems, we propose a solution that combines the resolution of the eye-tracking problem with the challenges of the input methods. In this paper, it introduces a new approach that eliminates the need for an eye-tracking device and only uses the webcam. A person’s needs lie in designing a keyboard that can be typed with a person’s eyes only using an inexpensive device and does not affect the health of the eyes. Therefore, a virtual keyboard that operates with eye movements and uses only the webcam to capture eye movements was designed. This reduces the cost of expensive eye-tracking devices, protects the individual from the concentration of infrared rays produced by these devices, and provides comfort and safety because no device will be mounted directly on the body.

The proposed virtual keyboard employs a series of consecutive procedures. The methodology’s architectural structure is shown in Figure 3.

Figure 2
Text saved in a Word document

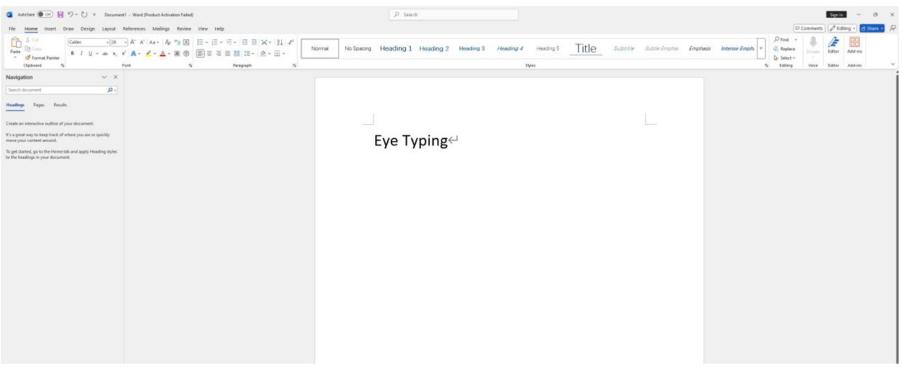
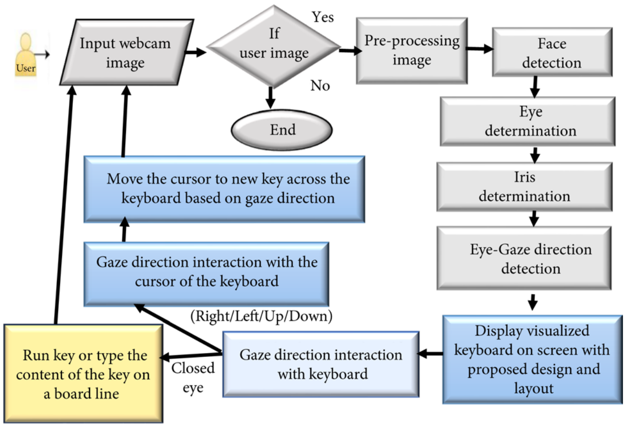


Figure 3
The architecture of the virtual keyboard



3.2.1. Preprocessing webcam images

The assumption is that a person is seated directly facing the screen, and the embedded camera is at the highest point of the horizontal axis of their gaze, with excellent lighting. This assumption allows for image capture in the real world rather than in traditional laboratory settings. The proposed system can correctly predict the sitting of a person using the algorithm of calibration seating in Khaleel et al. [23]. To improve the illumination of the input images and to recover the colors buried in darkness to obtain high-accuracy prediction, it used the method (Split-HSV) reported by Khaleel et al. [24].

3.2.2. Facial detection

The first goal in constructing the suggested model is to perform facial identification. Face detection is the identification of the location of human faces inside digital pictures. Face detection is performed using a combination of classifiers and feature-based methods [25]. In this study to detect facial features, the MediaPipe FaceMesh landmark detector library in Python was used. It used 36 landmarks to identify the face borders [23]. The facial landmarks used for face border detection are shown in Figure 4 [23] and listed in Table 2.

3.2.3. Eye detection

Face landmark points have been selected to just localize the eye points. There are 32 points of the eye in the face, where each eye has 16 points, as shown in Figure 5 and Table 3. These 32 points on the face have been localized using a MediaPipe FaceMesh shape predictor.

3.2.4. Eye gaze detection

The process of determining the gaze direction of both eyes using a suggested algorithm called Distance Eyelid-Iris-MediaPipe involves two phases:

The first phase is to determine the orientations (up, down, and closed) by measuring the distance between eyelids. In the second phase, the two orientations (left and right) are determined by identifying the position of the iris. The iris, which is the colored portion of the eye, is responsible for controlling the size of the pupil, as shown in Figure 6.

Utilize the MediaPipe FaceMesh landmark detector to ascertain the direction. Figure 7 identifies the distinctive landmarks of the iris in the left and right eyes. Table 4 shows the landmark points of the iris as follows: The distance between the upper and lower eyelids is the range of gaze direction ratios that have been calculated in a specific direction, as shown in Figure 8 and Table 5.

When the top and lower eyelashes connect, the eyelid is closed, rendering the eyeball hidden from view. Due to the relatively low

Figure 4
Mesh solution map of MediaPipe face

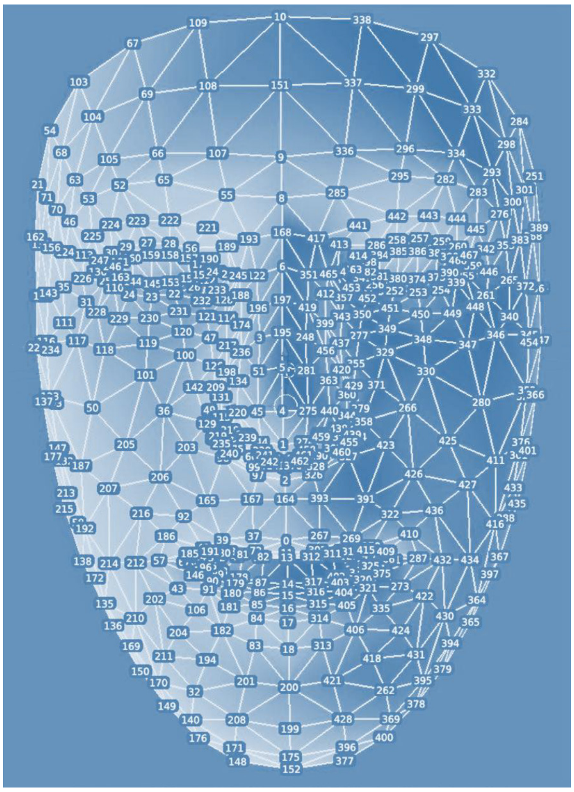


Table 2
Landmark indices of the face border

Parts of the face	Indices of landmarks
Face border	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

Figure 5
Mesh solution map of MediaPipe eyes

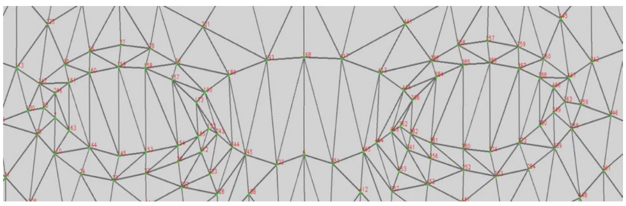


Table 3
Indices of the landmarks of eyes

Parts of Eyes	Indices of landmarks
Left eye	362, 382, 381, 380, 374, 373, 390, 249, 263, 466, 388, 387, 386, 385,384, 398
Right eye	33, 7, 163, 144, 145, 153, 154, 155, 133, 173, 157, 158, 159, 160, 161, 246

Figure 6
Iris of the eye

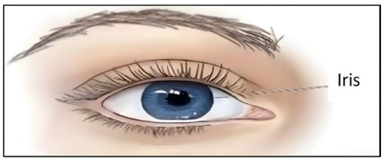


Figure 7
Landmarks of the iris

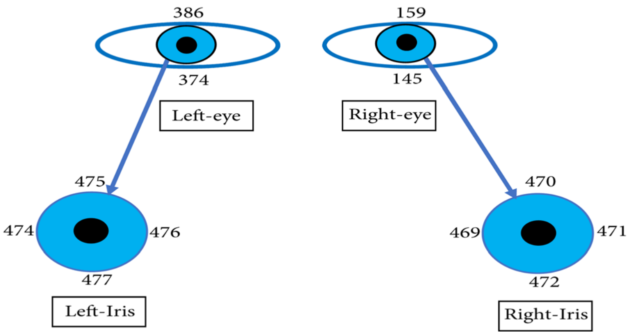


Table 4
Indices of the landmarks of the iris

Parts of the Iris	Indices of landmarks
Left iris	474, 475, 476, 477
Right iris	469, 470, 471, 472

Figure 8
Distance between eyelids

Table 5
Direction of gaze

Gaze ratio	Direction of gaze
Greater than 0.035	Up
Less than or equal to 0.035 and greater than 0.02	Center
Less than or equal to 0.02 and greater than 0.01	Down
Less than 0.01	Closed (Blink)

elevation angle of the eye in the proposed system, the value was adjusted from 0.04 to 0.035.

The distance between eyelids was measured separately for each eye, and the result of any eye to which the proposed distance law applies is taken because, if only one eye is used in prediction, it is possible that this eye may be damaged or the lighting may not reach it properly, thus negatively affecting the result.

The proposed system assumed n = 2, where n is a prime number. Prime numbers are used scientifically as building units, which are used

Distance Eyelid-Iris-MediaPipe Algorithm

Input: Eye image
Output: Direction of eye gaze
Begin
Step 1: Determine Vertical Direction (Up, Down, or Closed)
// Measure the distance between the eyelids to determine the vertical gaze direction
IF (gaze ratio of left eye < 0.01) OR (gaze ratio of right eye < 0.01):
Direction = Closed
ELSE IF (gaze ratio of left eye ≤ 0.02) OR (gaze ratio of right eye ≤ 0.02):
Direction = Down
ELSE IF (gaze ratio of left eye> 0.035) OR (gaze ratio of right eye > 0.035):
Direction = Up
Step 2: Determine Horizontal Direction (Left, Right, or Center)
// Identify the position of the iris within the eye to determine left or right gaze (for both eyes)
Step 2.1: Extract Iris Landmark Coordinates

- Determine the x, y coordinates of the upper and lower iris landmarks based on the camera frame size:
x=landmark.x × frame_width (1)
y=landmark.y × frame_height (2)

Step 2.2: Calculate the Center Point of the Iris

- Compute the axial center of the iris using the top and bottom iris landmark points:
Center.x= topIris.x + n (3)
Center.y=int((abs(buttomIris.y-topIris.y)/2)+ topIris.y + (2n-1) (4)

Step 3: Determine the Horizontal Gaze Direction
//Compare the distances between the center of the iris and the top/bottom iris points:
If abs(center.x- topIris.x) > abs(center.x- buttomIris.x) and
abs(center.y- topIris.y) < abs(center.y- buttomIris.y) then
Direction =left
Else Direction =right.
Step 4: Return the detected eye gaze direction.
End.

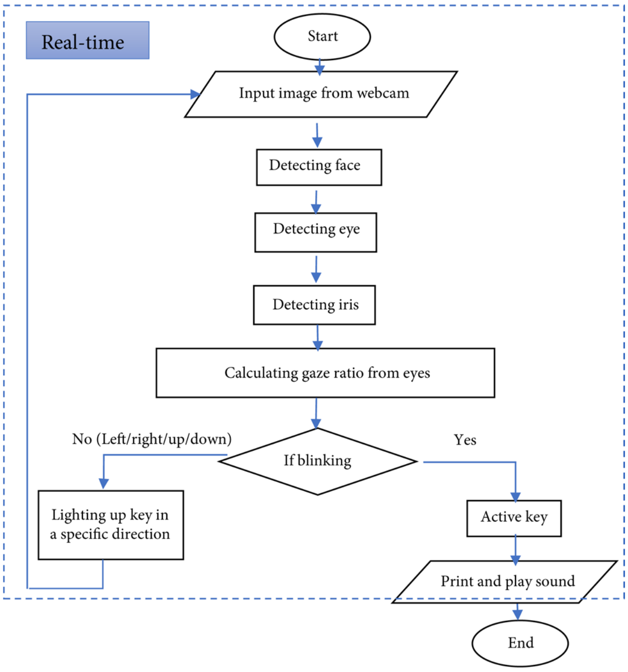
as the building units for detecting the eye. The smallest prime number (2) was chosen because eyes are naturally small and accept relatively small magnification.

3.3. Virtual keyboard working

The flowchart of the virtual keyboard's stages is illustrated in Figure 9.

The mechanism for moving and pressing a key is based on the calculation of the gaze ratio and the detection of eye blinking using a camera. The user's eye gaze is measured to determine the direction of key activation, guaranteeing the shortest horizontal and vertical routes to the requested key. This eye blink detection triggers the selection of the desired key and typing the letter. If the requirement for detecting eye blinks aligns with the specified activation time for a certain key, that key will be pushed, and a sound will be played while the corresponding letter is shown on the board. The cursor on the keyboard may be moved in four directions: left, right, up, and down, as shown in the Gaze

Figure 9
Virtual keyboard working



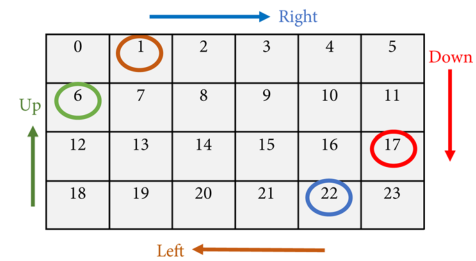
Direction Interaction with the Cursor (I) Algorithm, and the action and track of the gaze are detailed in Table 6.

Because the keyboard consists of six columns and four rows, the keys are numbered as shown in Figure 10, with values of i = (0–23).

Gaze direction interaction with the cursor (I) Algorithm

Input: Gaze direction, cursor number
Output: New cursor number
Begin
  Step 1: If Gaze Direction is Right: Check if I <= 22
    Yes → I = I + 1
    No → I = 23
  Step 2: If Gaze Direction is Down: Check if I <= 17
    Yes → I = I + 6
    No → I = I
  Step 3: If Gaze Direction is Up: Check if I >= 6
    Yes → I = I - 6
    No → I = I
  Step 4: If Gaze Direction is Left: Check if I >= 1
    Yes → I = I - 1
    No → I = 0
  Step 5: Return the cursor number (I)
End.

Figure 10
Layout of the keyboard



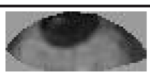
4. Results and Discussion

In this section, the results are presented and evaluated using various measures.

4.1. Apparatus and materials

The system was built using an MSI laptop equipped with an 11th Gen Intel(R) Core (TM) i7-11800H CPU running at 2.30 GHz. The laptop was also equipped with 16 GB of RAM and had a screen resolution of 1920 × 1080 pixels, with a size of “15 in.” The system is programmed in Python with the Anaconda environment on Windows 10.

Table 6
Gaze track and action Gaze track and action

Gaze Track		Gaze action
Up		Move to up
Down		Move to down
Left		Move to left
Right		Move to right
Blink		Select
Center		No action

4.2. Participants

The participants maintained a stable seated position, with their eyes approximately 35–50 cm away from the laptop screen. The proposed research included a wider range of ages, including both youthful, middle-aged, and elderly participants. We sent invitations to 20 individuals, evenly split between 10 males and 10 females, to participate in the experiment. Participants’ age ranged from 15 to 45 years. Just three individuals have prior familiarity with gaze interaction. Participants were recruited via informal communication and had diverse educational backgrounds, including middle school, high school, literature, engineering, and computer science. All participants were healthy, only one had disabilities, and five had corrected their eyesight using eyeglasses and contact lenses.

4.3. Performance evaluation

The evaluation and discussion of the implications of the proposed virtual keyboard’s design were conducted using objective and subjective measurements. Task performance indicators such as task execution time, error rate, and user satisfaction, representing the main influence of users’ behavior on their interactions with new apps, were evaluated.

The proposed system poses no danger to users because it only utilizes a webcam. One of the most important reasons for its creation was to protect the user’s eyes from the infrared rays that were emitted by eye-tracking devices. Therefore, it upholds the Institutional Review Board (IRB), and samples have passed the “IRB” rules (the individuals who agreed to participate in research activities are not placed at undue risk, and they can give their consent freely and without fear of coercion).

Many experiments were conducted to train the user under standard conditions of good lighting and a suitable sitting position. To measure the accuracy of eye directions, five experiments were conducted over separate periods, and the following was concluded, as shown in Table 7. The accuracy of the gaze direction upward was increasing, as was the accuracy of the gaze direction downward. The accuracy of the gaze direction to the left increased, and the accuracy of the gaze direction to the right decreased or vice versa. Finally, in the case of closing the eyes, the accuracy from the first test was very high.

In the user training phase, six sentences were used to train the user on how to use the virtual keyboard application and know what its interfaces contain. The experiment consists of a single session. The participants started the experiment with the single-word training "Hello." Then, they typed five offered phrases, four of which were from the suggestions of Albadawi et al. and MacKenzie and Soukoreff [26, 27]. In the test phase, 10 different sentences from MacKenzie and Soukoreff were entered in each trial. The output of the proposed method for text input was compared with the method (CNN of mesh) proposed by Khaleel et al. [23], where the same sentences were used in the same computer environment. The results showed that the new proposed method had higher accuracy and shorter load time, as shown in Table 8.

4.3.1. Objective evaluation measures

To evaluate the performance of the suggested virtual keyboard, many objective metrics were chosen, based on which the accuracy of the keyboard was assessed as follows:

1) Typing Speed

a. Characters per Minute (CPM)

Equation (5) was used to calculate the number of characters typed per minute for different individuals across diverse topics and trials [28, 29].

$$\mathrm{CPM} = \frac{\textit{No. of character}}{\textit{Time taken}} \times 60 \tag{5}$$

b. Word per Minute (WPM)

The number of words typed per minute (WPM) is measured using Equation (6) as an indicator of text entry rate:

$$\mathrm{WPM} = \frac{\textit{No. of word}}{\textit{Time taken}} \times 60 \tag{6}$$

Table 7
Accuracy of the gaze direction

Test	Up%	Down%	Left%	Right%	Close%
1	80	90	98	88	99
2	87	94	89	93	100
3	96	97	94	95	100
4	98	99	99	99	100
5	98	99	99	99	100

Table 8
Comparison of the proposed work with the previous study

Methods	Accuracy (%)	Mean error	Execution time	Load time
CNN of mesh [23]	98.774	0.0183	1 s	12 s
Proposed method	99	0.001	1 s	3 s

2) Keystrokes per Character (KSPC)

The term "KSPC" refers to a calculation that determines the average number of keystrokes required to input each character of a given text [30, 31], as expressed in Equation (7):

$$\mathrm{KSPC} = \frac{\textit{No. of keystrokes}}{\textit{No. of Character}} \tag{7}$$

KSPC is a metric that quantifies the level of precision by accounting for the additional effort required to rectify errors. Optimally, the keyboard stroke per character (KSPC) value should be 1.00, signifying that each keystroke results in the production of a single character. When participants rectify errors when entering data, the error rate becomes 0%. Nevertheless, the KSPC value exceeds 1.

3) Error rate (ER)

When typing the words, errors such as miswords, adding extra characters or words, and missing words can occur. As a result, it can compare the transcribed text (user-typed text) with the supplied text using the metrics:

a. Character Error Rate (CER): It is a measure of how well an automated speech recognition (ASR) system performs [30]. This study employs this measure to quantify the disparity between the original and typed characters on the eye-board. It quantifies the number of changes, such as replacements, deletions, and insertions, that may be made while inputting text. The CER value represents the proportion of characters that were inaccurately anticipated. A lower number indicates superior system performance, with a CER of 0 representing a perfect score. The CER may be calculated using Equation (8):

$$CER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C} \tag{8}$$

In this case, the following variables are used: S represents the substitution count, D represents the deletion count, I represent the insertion count, C represents the correct character count, and N represents the total number of characters in the reference, which is calculated as the sum of S, D, and C.

b. Word Error Rate (WER): It is a widely used measure to evaluate the accuracy of automated speech recognition systems. The proposed research employs this measure to quantify the disparity between the original words and those typed on the eye-board. It quantifies the number of changes, such as replacements, deletions, and insertions, that may be made while inputting text. This metric represents the proportion of words that were inaccurately forecasted. A lower number indicates superior system performance, and a word error rate of 0 represents a perfect score. Equation (9) calculates the word mistake rate as follows:

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C} \tag{9}$$

The variables S, D, I, C, and N represent the number of substitutions, deletions, insertions, correct words, and total words in the reference, respectively (N = S + D + C).

c. Total Error rate (TER): The total error rate is calculated using the average between the character error rate and word error rate, as shown in Equation (10):

$$TER = \frac{(CER+WER)}{2} \tag{10}$$

4.3.2. Subjective evaluation measures

1) NASA Task Load Index

The NASA Task Load Index (NASA-TLX) is a widely used, objective, multidimensional instrument to gauge the subjective level of effort needed to judge the effectiveness and/or other features of system operation. NASA-TLX breaks down the workload into six dimensions to produce a measure of the overall workload. Cognitive load, physical exertion, time constraints, task performance, level of exertion, and feelings of frustration are the six factors. The NASA-TLX final scores range from 0 to 100. Lower numbers reflect better performance. This indicator was utilized in the virtual keyboard application to measure how much load users were under [32, 33].

2) System Usability Scale

The system usability scale (SUS) is a 10-item Likert scale that provides a comprehensive assessment of subjective usability and learnability as well as satisfaction with the system. This measurement is used to help us evaluate a system through three primary dimensions: effectiveness, efficiency, and satisfaction [32]. The final scores on the SUS range from 0, representing low usability, to 100, which indicates good usability [32].

4.4. Comparison of the results and discussion

The results of the experiments using these measures are presented here. The experiments were conducted for eight consecutive weeks (eight trials: one trial per week), which are shown below.

Table 9
Comparison of the typing experiments

Trials	WPM	CPM	TER
1	1.23	6.17	90
2	1.39	6.96	83
3	1.82	9.09	75
4	1.96	9.81	60
5	2.24	11.18	58
6	2.26	11.3	40
7	3.01	15.06	10
8	3.24	16.21	0

4.4.1. Comparison of the typing experiments

The word “Hello” was typed several times, as shown in Table 9. It has been shown that the results of the measures improve with each new experience. This is due to the user memorizing the letter’s location and knowing how to use eye movements better to reach the letter to be typed.

Table 9 demonstrates a positive correlation between the number of trials and the rising value in WPM and CPM. Due to the individual’s lack of familiarity with the visual keyboard, the early attempts resulted in low WPM and CPM rates but a large TER score. Nonetheless, as the individuals become used to the keyboard, their WPM and CPM rates increased with a progressive decrease in TER.

4.4.2. Comparison of the error typing experiments

When typing, various errors occur, such as deleting, adding, or replacing a letter or word, as shown in Table 10.

Table 10 shows the spelling errors that occurred in the word “HELLO” and the sentence “HI THERE.” It is noted that the TER scale showed the exact difference in all cases of typing errors, although the accuracy scale value was equal in some cases.

4.4.3. Comparison between hand-typing and eye-typing

To compare the speed of the user typing English letters using his hand (handwriting) and using his eyes (eye-typing), a well-known English pangram sentence comprising all 26 letters of the English language was used. The sentence was “The quick brown fox jumps over a lazy dog,” and six volunteers participated in this experiment.

Naturally, typing using our hands is faster than typing using our eyes because typing using our hands uses 10 fingers, which are typing in parallel. Table 11 shows that the first participant was proficient in hand-typing. Thus, there was a large difference in typing time. Hence, the participants were arranged from the least skilled (those who use one finger to type and do not know the locations of the letters on the keyboard) to the most skilled (those who use 10 fingers to type).

In summary, fast typing using our eyes is approximately equal to medium-speed typing using our hands, as shown in Table 11.

4.4.4. Comparison of eye-typing results by lengths of the word

Different word lengths were used to determine the effect of increasing the number of letters in a word on both measures (WPM and CPM), as shown in Table 12 and Figure 11.

It is noted that time is directly affected by the length of the word because it increases with an increase in the number of letters in the word

Table 10
TER comparison of the proposed system

Error typing	Word typing	WER	CER	TER%	Old ACC (%)	New ACC (%)
Deletion word	HI	1	0.6	80	25	20
Deletion word	THERE	1	0.6	80	0	20
Missing characters	H THERE	0.5	0.14	32	12.5	68
Missing characters	HI THER	0.5	0.14	32	87.5	68
Wrong characters	HE THERE	0.5	0.12	31	87.5	69
Wrong characters	HI THEER	0.5	0.25	37.5	75	62.5
Wrong characters	HE THEER	1	0.37	68.5	62.5	31.5
Extra characters	HE THEREE	0.5	0.11	30.5	88.8	69.5
Replacement word	THERE HI	1	0.75	87.5	0	12.5
Missing characters	HELL	1	0.25	62.5	80	37.5
Wrong characters	HILLO	1	0.20	60	80	40
Extra characters	HELLOO	1	0.16	58	83.3	42

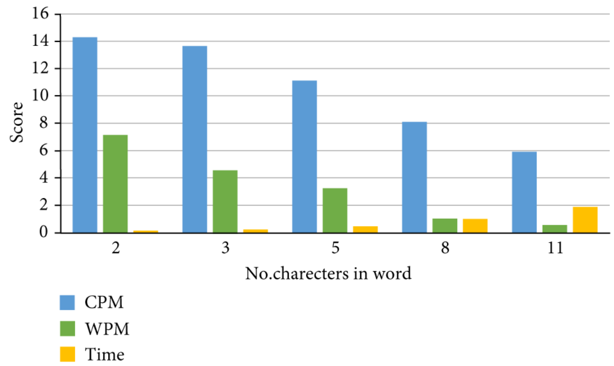
Table 11
Comparison between hand-typing and eye-typing

	Hand-typing			Eye-typing		
Users	**WPM**	**CPM**	**Duration (second)**	**WPM**	**CPM**	**Duration (second)**
1	1.42	6.5	6.30	0.66	3.00	13.64
2	1.47	6.72	6.1	0.8	3.64	11.27
3	2.04	9.29	4.41	0.99	4.53	9.05
4	3.53	16.07	2.55	1.09	4.95	8.28
5	6	27.33	1.5	1.42	6.50	6.3
6	8.82	40.19	1.02	2.07	9.46	4.33

Table 12
Comparison of eye-typing results by lengths of the word

Words	Length of word	WPM	CPM	Time
Hi	2	7.14	14.28	0.14
Eye	3	4.55	13.64	0.22
Hello	5	2.22	11.12	0.45
Internet	8	1.01	8.1	0.99
Information	11	0.54	5.9	1.87

Figure 11
Comparison of eye-typing results by lengths of the word



to be typed. In contrast, the values of the WPM and CPM metrics are inversely proportional to the length of the word.

4.4.5. Comparison of eye-typing results by lengths of the English sentences

To know the effect of increasing the number of words in sentences on the two measures (WPM and CPM), five sentences with different numbers of words were used, as shown in Table 13.

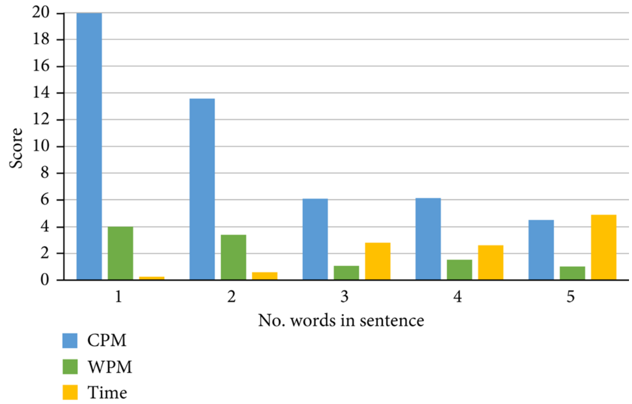
It is clear from Table 13 and Figure 12 that the values of the two scales (WPM and CPM) decrease as time increases. The decrease is likely because the user's ability to type decreases over time due to the stress of typing many characters and words. It can also be noted that if the number of letters in the sentences is equal but the number of words is unequal, the two measures (WPM and CPM) increase, as in the case of sentences 3 and 4.

Table 13
Comparison of eye-typing results by lengths of the sentences

No.	Sentences	No. of words	No. of char.*	WPM	CPM	Time
1	Hello	1	5	4	20	0.25
2	Hi there	2	8	3.39	13.5	0.59
3	Fish are jumping	3	16	1.07	5.71	2.8
4	I agree with you	4	16	1.53	6.13	2.61
5	This is very good idea	5	22	1.02	4.5	4.89

Note: *No. of Char. = No. of Letters + No. of Spaces

Figure 12
Comparison of eye-typing results by lengths of the sentences



4.4.6. Comparison of prediction of English words

The feature of word prediction (PW) by eye-typing is used in the proposed system. Therefore, it was observed that the proposed system's typing speed increased when comparing the previous results shown in Table 13 with those shown in Table 14.

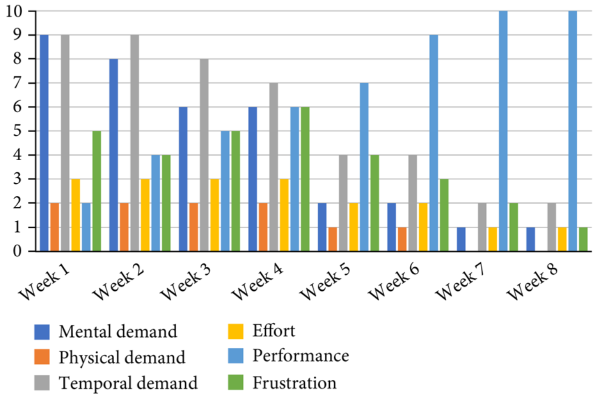
4.4.7. Comparison of the NASA Task Load Index

The proposed system achieved a NASA-TLX score of 10%, which means that it only puts a small load on the user. Through Figure 13, we notice the changes in the NASA-TLX scale and conclude the following: Mental demand decreases over time, and physical demand is small from the beginning and decreases more over time. In addition, the effort is relatively low from the beginning and decreases with time, and the time demand decreases significantly over time and is inversely proportional to performance, which increases significantly over time. As for frustration, its value ranges from up to down, but in the end, it decreases significantly.

Table 14
Comparison of prediction of English words

Sentences	No. of words	WPM+PW	CPM+PW	Time
Hello	1	13.78	68.88	0.07
Hi there	2	11.44	45.74	0.17
Fish are jumping	3	1.56	8.8	1.93
I agree with you	4	1.85	7.39	2.17
This is very good idea	5	1.43	6.30	3.49

Figure 13
Average NASA-TLX scores



4.4.8. Comparison of the SUS

The proposed system showed a remarkable result. It achieved 93 (Figure 14), surpassing the results of previous systems.

Such an achievement can be attributed to using alphabetical order in the keypad design, which makes learning easier. In addition, to improve efficiency, various auxiliary submenus have been included. For example, there is a calculator menu dealing with both integer and fractional numbers. Moreover, the diverse tool menu not only gives variation but also is used for structuring and modification of texts to enhance system efficiency. Finally, a menu of emojis is a novel feature that can convey the user's emotions even without speaking or typing, resulting in a significant increase in user contentment.

4.5. Comparison of the proposed visual keyboard system and earlier systems

Much of the research deals with the challenges faced in designing a virtual keyboard and the techniques used to control it. The following illustrates the comparison of the proposed visual keyboard system and the earlier systems.

4.5.1. Comparison using objective evaluation measures

The CPM and WPM of the proposed system were compared with those of the previous systems. The suggested system outperformed the earlier systems, as shown in Table 15.

Most modern research utilizes eye-tracking devices in eye-typing, but the proposed system employs only a webcam. Therefore, a small number of the papers have been compared with the proposed system. Table 15 shows that across all tested words, the proposed system significantly outperformed reference systems in terms of both speed (WPM and CPM) and efficiency (lower time per input). The improvement is especially dramatic in the most recent benchmark [21], highlighting the effectiveness of the new method in high-performance scenarios.

Figure 14
Average SUS scores

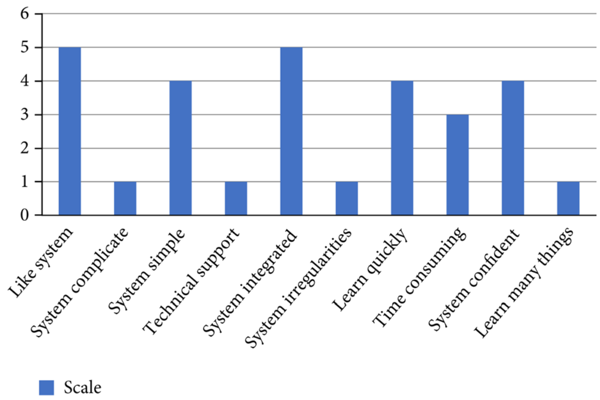


Table 15
Comparison of the proposed visual keyboard system and earlier systems

Word	REF.	WPM of REF.	CPM of REF.	WPM of Prop.	CPM of Prop.	TIME
HI THERE	[12]	2.35	9.50	4.68	18.60	0.43
HELLO	[14]	2.70	---	3.7	18.51	0.27
HELLO	[20]	6.9*	---	13.78**	68.88	0.07

Note: * by hand, **CPM+PW

Table 16
Comparison of the proposed system and earlier systems

REF.	SUS of REF.	NASA-TLX of REF.
[34]	87	17
[12]	75.4	---
Proposed	93	10

4.5.2. Comparison using subjective evaluation measures

The comfort of the users in terms of flexibility and ease of use is the new influential factor in measuring the importance of the proposed system. The characteristics of the virtual keyboard, including its shape, color, size, and number of keys, and even an alteration in the arrangement of keys can substantially boost typing efficiency. Therefore, the proposed keyboard system is better than the previous ones, as shown in Table 16.

5. Conclusion

Recently, millions of people have suffered from severe physical handicaps that they cannot meet their basic needs. A person's needs are focused on communicating with others in society. Communication means interacting with the world through gestures, which is one of the most essential human needs. The language of communication is important for a person to express what is inside him and communicate what he wants to others. Communication is conducted through three methods: speech, writing, and signaling by moving the hand or head. In the case of disabled people who cannot move or speak, communication is considered difficult and takes only one method, which is through writing. In most severe degrees of disability, only the eyes can move, and an electronic interface that could write using only eye gaze could improve the quality and productivity of those who are disabled, making them more independent.

The progress of computer vision has resulted in several technologies that aid the disabled in interacting with computers or digital devices to perform various tasks. One of the most crucial technologies for inputting information into computers is the virtual keyboard. This paper introduces a new model for a virtual keyboard. It uses gesture control technology, allowing a disabled person to control the keyboard by moving their eyes. It does not use expensive equipment to track a person's gaze but a webcam to reduce costs. The proposed virtual keyboard consists of six secondary menus containing English letters, in addition to assistant tools that help in formatting the text, checking its spelling, and writing it quickly through prediction.

Another feature of the proposed new keyboard is the use of emojis to express what the person wants without typing. The keyboard was also linked to WhatsApp to enable the user to communicate with others, in addition to enabling the user to store what was written in a Word file. Due to the speed and accuracy of the proposed keyboard, as well as its ability to provide the user with comfort and satisfaction, these new features significantly improved the evaluation results of the participants. The proposed system achieved a NASA-TLX score of 10% and an SUS score of 93. In the future, system assessment will include more complicated phrases and more participation from users with disabilities.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data are available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Amal Hameed Khaleel: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Thekra Abbas:** Methodology, Validation, Resources, Writing – original draft, Writing – review & editing, Supervision. **Abdul-Wahab Sami Ibrahim:** Conceptualization, Formal analysis, Investigation, Resources, Writing – review & editing, Supervision.

References

[1] Samanta, D., Sarcar, S., & Ghosh, S. (2013). An approach to design virtual keyboards for text composition in Indian languages. *International Journal of Human-Computer Interaction*, *29*(8), 516–540. https://doi.org/10.1080/10447318.2012.728483

[2] de Sousa Gomide, R., Loja, L. F. B., Lemos, R. P., Flôres, E. L., Melo, F. R., & Teixeira, R. A. G. (2016). A new concept of assistive virtual keyboards based on a systematic review of text entry optimization techniques. *Research on Biomedical Engineering*, *32*(2), 176–198. https://doi.org/10.1590/2446-4740.01715

[3] Chakraborty, P., Roy, D., Rahman, Z., & Rahman, S. (2019). Eye gaze controlled virtual keyboard. *International Journal of Recent Technology and Engineering*, *8*(4), 3264–3269. https://doi.org/10.35940/ijrte.D8049.118419

[4] Shabaz, M., & Soni, M. (2024). Generative adversarial-based ubiquitous data integration model for human re-identification. *Journal of Computational and Cognitive Engineering*. Advance online publication. https://doi.org/10.47852/bonviewJCCE42022872

[5] Dahmani, M., Chowdhury, M. E., Khandakar, A., Rahman, T., Al-Jayyousi, K., Hefny, A., & Kiranyaz, S. (2020). An intelligent and low-cost eye-tracking system for motorized wheelchair control. *Sensors*, *20*(14), 3936. https://doi.org/10.3390/s20143936

[6] Cecotti, H., Meena, Y. K., Bhushan, B., Dutta, A., & Prasad, G. (2019). A multiscript gaze-based assistive virtual keyboard. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1306–1309. https://doi.org/10.1109/EMBC.2019.8856446

[7] Sarcar, S., Panwar, P., & Chakraborty, T. (2013). EyeK: An efficient dwell-free eye gaze-based text entry system. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, 215–220. https://doi.org/10.1145/2525194.2525288

[8] Prabhu, V., & Prasad, G. (2011). Designing a virtual keyboard with multi-modal access for people with disabilities. In *2011 World Congress on Information and Communication Technologies*, 1133–1138. https://doi.org/10.1109/WICT.2011.6141407

[9] Krishnasamy, N., Zade, N., Khambholia, D., Henry, R., & Gupte, A. (2025). Ensemble deep learning framework for hybrid facial datasets using landmark detection: State-of-the-art tools. *Journal of Computational and Cognitive Engineering*. Advance online publication. https://doi.org/10.47852/bonviewJCCE52024451

[10] Souto, D., Marsh, O., Hutchinson, C., Judge, S., & Paterson, K. B. (2021). Cognitive plasticity induced by gaze-control technology: Gaze-typing improves performance in the antisaccade task. *Computers in Human Behavior*, *122*, 106831. https://doi.org/10.1016/j.chb.2021.106831

[11] Velichkovsky, B. B., Rumyantsev, M. A., & Morozov, M. A. (2014). New solution to the Midas touch problem: Identification of visual commands via extraction of focal fixations. *Procedia Computer Science*, *39*, 75–82. https://doi.org/10.1016/j.procs.2014.11.012

[12] de Rosa, M., Fuccella, V., Costagliola, G., Adinolfi, G., Ciampi, G., Corsuto, A., & di Sapia, D. (2020). T18: An ambiguous keyboard layout for smartwatches. In *2020 IEEE International Conference on Human-Machine Systems*, 1–4. https://doi.org/10.1109/ICHMS49158.2020.9209483

[13] Islam, R., Rahman, S., & Sarkar, A. (2021). Computer vision based eye gaze controlled virtual keyboard for people with quadriplegia. In *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0*, 1–6. https://doi.org/10.1109/ACMI53878.2021.9528213

[14] Jeevithashree, D. V., Jain, P., Mukhopadhyay, A., Saluja, K. P. S., & Biswas, P. (2021). Eye gaze controlled adaptive virtual keyboard for users with SSMI. *Technology and Disability*, *33*(4), 319–338. https://doi.org/10.3233/TAD-200292

[15] Benabid Najjar, A., Al-Wabil, A., Hosny, M., AlRashed, W., & Alrubaian, A. (2021). Usability evaluation of optimized single-pointer Arabic keyboards using eye tracking. *Advances in Human-Computer Interaction*, *2021*(1), 6657155. https://doi.org/10.1155/2021/6657155

[16] Bharath, R. R., Dinesh Kumar, L., Jayasuriya, C., & MeenatchiSundaram, T. (2022). Controlling mouse and virtual keyboard using eye-tracking by computer vision. *Journal of Algebraic Statistics*, *13*(3), 3354–3368.

[17] Abhaya, V., Bharadwaj, A. S., Bagan, C. C., Dhanraj, K., & Shyamala, G. (2022). Eye-move: An eye gaze typing application with OpenCV and Dlib library. In *2022 International Conference on Automation, Computing and Renewable Systems*, 952–957. https://doi.org/10.1109/ICACRS55517.2022.10029276

[18] Cui, W., Liu, R., Li, Z., Wang, Y., Wang, A., Zhao, X., …, & Bi, X. (2023). GlanceWriter: Writing text by glancing over letters with gaze. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, *719*. https://doi.org/10.1145/3544548.3581269

[19] Mifsud, M., Camilleri, T. A., & Camilleri, K. P. (2023). HMM-based gesture recognition for eye-swipe typing. *Biomedical Signal Processing and Control*, *86*, 105161. https://doi.org/10.1016/j.bspc.2023.105161

[20] Mallik, B., Rahim, A., Miah, A. S. M., Yun, K. S., & Shin, J. (2024). Virtual keyboard: A real-time hand gesture recognition-based character input system using LSTM and MediaPipe holistic. *Computer Systems Science and Engineering*, *48*(2), 555–570. https://doi.org/10.32604/csse.2023.045981

[21] Benligiray, B., Topal, C., & Akinlar, C. (2019). SliceType: Fast gaze typing with a merging keyboard. *Journal on Multimodal User Interfaces*, *13*(4), 321–334. https://doi.org/10.1007/s12193-018-0285-z

[22] Zhang, C., Yao, R., & Cai, J. (2018). Efficient eye typing with 9-direction gaze estimation. *Multimedia Tools and Applications*, *77*(15), 19679–19696. https://doi.org/10.1007/s11042-017-5426-y

[23] Khaleel, A. H., Abbas, T. H., & Sami Ibrahim, A. W. (2024). Best low-cost methods for real-time detection of the eye and gaze tracking. i-com, *23*(1), 79–94. https://doi.org/10.1515/icom-2023-0026

[24] Khaleel, A. H., Abbas, T. H., & Ibrahim, A. W. S. (2024). A novel convolutional feature-based method for predicting limited mobility eye gaze direction. *International Journal of Advances in Intelligent Informatics*, *10*(2), 220–238. https://mail.ijain.org/index.php/IJAIN/article/view/1370

[25] Hu, X., Jeon, Y., & Gwak, J. (2023). Heterogeneous ensemble approaches for robust face mask detection in crowd scenes. *Journal of Computational and Cognitive Engineering*, *2*(4), 343–351. https://doi.org/10.47852/bonviewJCCE3202478

[26] Albadawi, Y., AlRedhaei, A., & Takruri, M. (2023). Real-time machine learning-based driver drowsiness detection using visual features. *Journal of Imaging*, *9*(5), 91. https://doi.org/10.3390/jimaging9050091

[27] MacKenzie, I. S., & Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, 754–755. https://doi.org/10.1145/765891.765971

[28] Khaleel, A. H., Abbas, T. H., & Ibrahim, A.-W. S. (2023). Enhancing human-computer interaction: A comprehensive analysis of assistive virtual keyboard technologies. *Ingenierie des Systemes d'Information*, *28*(6), 1709–1717. https://doi.org/10.18280/isi.280630

[29] Wan, T., Shi, R., Xu, W., Li, Y., Atkinson, K., Yu, L., & Liang, H.-N. (2024). Hands-free multi-type character text entry in virtual reality. *Virtual Reality*, *28*(1), 8. https://doi.org/10.1007/s10055-023-00902-z

[30] Roux, T. B., Rouvier, M., Wottawa, J., & Dufour, R. (2022). Qualitative evaluation of language model rescoring in automatic speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association*, 3968–3972. https://doi.org/10.21437/Interspeech.2022-10931

[31] Li, T., Yu, L., Zhou, L., & Wang, P. (2023). Using less keystrokes to achieve high top-1 accuracy in Chinese clinical text entry. *Digital Health*, *9*, 20552076231179027. https://doi.org/10.1177/20552076231179027

[32] Rajanna, V., & Hansen, J. P. (2018). Gaze typing in virtual reality: Impact of keyboard design, selection method, and motion. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, *15*. https://doi.org/10.1145/3204493.3204541

[33] Whittington, P., & Dogan, H. (2017). SmartPowerchair: Characterization and usability of a pervasive system of systems. *IEEE Transactions on Human-Machine Systems*, *47*(4), 500–510. https://doi.org/10.1109/THMS.2016.2616288

[34] Meena, Y. K., Cecotti, H., Wong-Lin, K., & Prasad, G. (2019). Design and evaluation of a time adaptive multimodal virtual keyboard. *Journal on Multimodal User Interfaces*, *13*(4), 343–361. https://doi.org/10.1007/s12193-019-00293-z

How to Cite: Khaleel, A. H., Abbas, T., & Ibrahim, A.-W. S. (2025). Multitask Virtual Keyboard Controlled by an Eye-Gaze-Based Intelligent Entry System. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonview-JCCE52026027