Eye Gaze Controlled Virtual Keyboard

Patchipulusu Sindhu - 191IT137
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: sindhu.191it137@nitk.edu.in

Alina Melony Pinto - 191IT206
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: alinapinto.191it206@nitk.edu.in

Sanjeevi Meghana - 191IT144
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
Email: sanjeevimeghana.191it144@nitk.edu.in

***Abstract*—Human computer interaction (HCI) is concerned with the interface and interaction between people and computers. Human-Computer Contact is a new field of study that encompasses a wide range of algorithms and strategies for improving human-computer interaction. HCI's major purpose is to create machinery that allows people to interact with computers in innovative ways. Disabilities caused by genetic or severe injury prevent people from expressing their basic needs or ideas, making it extremely difficult for them to communicate in order to express their ideas or information to others. A part of society needs to integrate into the community, communicate with others, and express themselves, yet they are unable to do so due to varying levels of impairment. Some people are unable to move or speak due to a variety of factors, including chronic disease, a congenital abnormality, or an accident in which their ability to speak or move has been lost. They are unable to communicate with the rest of the community as a result. Consequently, there is a need for an eye-computer interface system. In this paper, a virtual keyboard controlled by eye gaze is implemented, through which users can be able to type out characters with their eye gaze and eye blink. In addition, for a given text the application is analysed with characters per minute, error rate with respect to the different layouts of the keyboard.**

***Index Terms*—Face Detection, Gaze Detection and Blink Detection, Tkinker GUI**

I. INTRODUCTION

With the rapid advancement in new technical sectors such as AI, deep learning, and robotics, the communication between humans and computers is becoming easier. Human Computer Interaction (HCI) is an advanced technology field that focuses on improving and creating human-computer interaction systems. The quality of interaction has been the primary driver of progress in the HCI sector. HCI [1] is used in a variety of fields, including medical technology, robotics, gaming, virtual reality (VR), augmented reality (AR), health care and assistive technology and many more. Visual-based human-computer interaction is the most widely studied topic of HCI. Researchers are attempting to solve a number of open-ended challenges in light of the wide range of possible applications between humans and machines. The main purpose of HCI is to provide interaction, which is crucial in our life as it allows us to connect and engage with people around us. Everyone needs the ability to integrate into the community, communicate with others, and express themselves, but they are unable to do so due to various levels of impairment. Some people are unable to move or talk owing to a variety of reasons, including congenital anomaly, suffering from Quadriplegia or an accident that has taken away their capacity to move or speak. As a result, they're unable to communicate with the rest of the community.

Quadriplegia [2] is a type of paralysis that can also result in BRAIN DISEASES, SPINAL CORD DISEASES, PERIPHERAL and MUSCULAR DISEASES which can all cause severe or full loss of motor function in all four limbs. Quadriplegia combined with cranial muscular paralysis characterizes the locked-in syndrome. Consciousness is preserved, and the only voluntary motor activity that remains is maybe limited eye movements. This combined with cranial muscular paralysis characterizes the locked-in syndrome. As a result, an eye-computer interface system is a necessitate. Consciousness is preserved, and the only voluntary motor activity that remains is maybe limited eye movements. A lesion [3] in the upper BRAIN STEM usually causes this disorder, which damages the descending cortico-spinal and cortico-bulbar tracts. It can be infected for people of any age, irrespective of being child, adult, men or women, but the recovery of the injury will definitely depend on the age, caring etc. There are many problems for the people who are suffering from this disease, as they cannot speak, write and are not able to do their own tasks, it is very difficult to communicate with the outside world.

In order to resolve this problem of communication for people who are suffering from quadriplegia, with the rest of the outside world, 'Eye Gaze Controlled Virtual Keyboard' is implemented, through which these people can communicate with the help of the eye gaze. Here Tkinker GUI is used to represent the keyboard and with the help of the user's gaze and blink the letter will print on the screen, which in turn will be helpful for the people to communicate with the people. Three layouts are created based on qwerty, colemax and maltron layout of the keyboard inorder to check which keyboard is more convenient for the user. The application is checked and

TABLE I
SUMMARY OF LITERATURE SURVEY

Authors	Methodology	Merits	Limitations	Additional Details
R. Islam, S. Rahman and A. Sarkar	Highlighting of the letters is done with the help of eye gaze and text printing is done with the help of eye blink detection.	Performance of error rate is decreased compared with the existing models.	The features and number of keys are less in the keyboard.	Significant improvement over the existing model.
Afraa Z. Attiah & Enas F. Khairullah	Eye blink detection for a virtual keyboard is used.	Usability of the system is measured with the users.	Mobile application is not available.	Designed mainly for the people who has the disability.
Chakraborty,Dipa Roy,Zahidur&Saifur	Key board selection and eye blink detection are used to print the text.	Provides flexibility of selecting the keyboard based on character.	Interface is only limited to python.	Added selection of the key board part.
V.I. Saraswati, Riyanto& T. Harsono	Eye gaze is used to choose the letter the user wants.	Segmentation process of the iris has generated good result.	User must be stable.	Compared performance of writing and typing using system.

analyzed with the help of end users.

The remaining paper is outlined as follows: In section II contains an overview of the papers that have been covered to implement the project. Section III, comprises the problem statement and the objectives of the application. Section IV, describes the methodology of the paper. In Section V, results of the empirical and statistical analysis with respect to the users are presented and finally, the work is concluded along with future work.

II. LITERATURE SURVEY

Authors in [4] researched about the highlighting of the letters forward and backward using the eye gaze detection and based on the eye blink detection the characters are printed on the screen and the movement of forward and backward based on the gaze improved the performance with the existing model.

In [5], the authors created the virtual keyboard and detected the eye blink to print the respective letter in the screen and is manly designed for the people who has the specific disability of motoric function. The usability of the users on the application is considered and analysed with the help of bar graphs.

Authors in [6], divided the keyboard into two parts and selected the keyboard using the gaze detection and printed the text on the screen using blink detection, provides the flexibility to select the key board based on the character required to print on screen. The main limitation in this way of printing is that the user need to remember the letters where it was present that is either left or right.

Communication is difficult for the people who are facing disability like quadriplegia, so to make the communication with the outside world [7] eye gaze is used to choose the letter from the virtual keyboard.

Table 4. provides a brief overview of the literature survey.

III. PROBLEM STATEMENT

To build a virtual keyboard that is controlled by eye gaze and blink detection, which helps aid people with disabilities that cannot efficiently use their hands to type or write.

A. Objectives

The objective of the application is to detect the eye gaze and blink, and then gaze to move right or left and blinking in order to select the desired key. This application includes the following objectives:

1) An Imaging processing module that detects the face and the eye
2) Eye gaze and blink detection
3) Integrate with a virtual keyboard GUI

IV. METHODOLOGY

Due to varied levels of impairment, a segment of society needs to integrate into the community, speak with others, and express themselves, but they are unable to do so. Some people are unable to talk or move for a variety of reasons, including chronic disease, a congenital defect, or an accident in which they have lost their capacity to speak or move. As a result, they are unable to communicate with the rest of the community. Consequently, there is a need for an eye-computer interface system.

An eye-computer interface is a system that allows acting operation depending on the eye. In this paper, an eye gaze and blink controlled keyboard is implemented. In this system, no muscle movements are required. This application uses Eye gaze to traverse right or left columns of the keyboard and Eye blinking in order to select the desired key to type a letter without using hands. Essentially, methodology mainly comprises of the following steps:

- Face and eyes part detection
- Eye blink detection
- Eye gaze detection
- Integration with keyboard layout

Methodology starts with live video capture of the user, from the video capture object of the camera and the user's face and facial data points are detected. Based on the eye data points obtained, left and right eye regions are analyzed for eye blink and gaze detection [8] . Then this is integrated with keyboard layout design in the user interface. This keyboard is incorporated with keyboard part highlighting where based

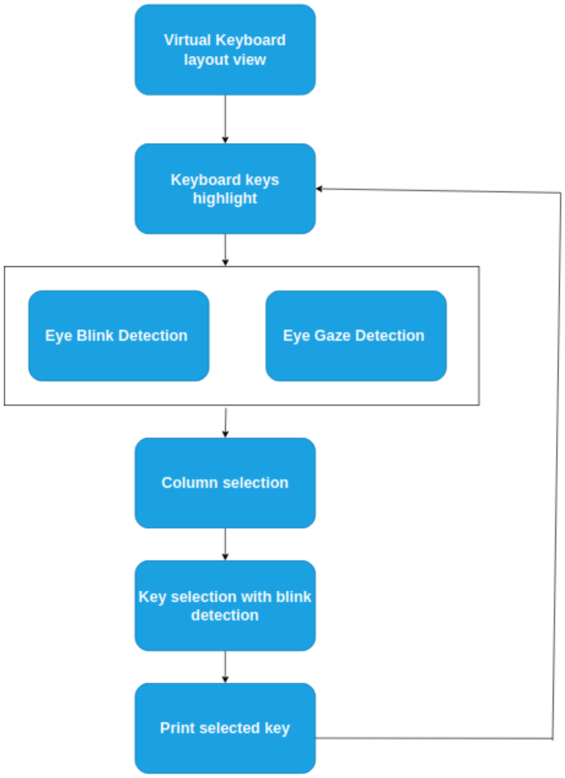

Fig. 1. Flowchart

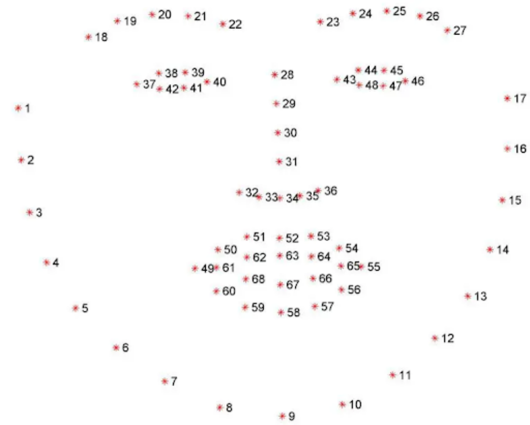

Fig. 2. The full set of facial landmarks that can be detected via dlib

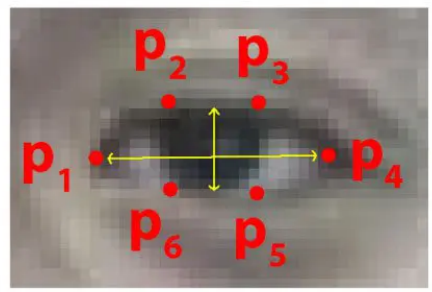

Fig. 3. The full set of facial landmarks that can be detected via dlib

on eye blink and eye gaze detected, parts of the keyboard are highlighted. As the user shifts their gaze, the application highlights the corresponding part of the keyboard. Blink is used to select the highlighted region.

A. Face Detection

For the face detection part, dlib's 68 face feature model is used, which detects 68 facial landmark points.This is better visualised in Fig 2. This model allows us to localize not just the face but the eye, nose, jaw, etc. For our application, eye landmarks points are used which correspond to 36 to 41 indexes for the left eye and 42 to 47 for the right eye.(from Fig 2)

B. Blink Detection

The detected eye region data points are then further analyzed to detect blink. For this, the Eye Aspect Ratio (EAR) of the left and right eye are calculated using formula shown in Fig.4. Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye and then working clockwise around the remainder of the region as shown in Fig. 3. Where p1, . . . , p6 are 2D facial landmark locations.

The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, weighting the denominator appropriately since there is only one set of horizontal points but two sets of vertical points. The eye aspect ratio(EAR given by equation 1) is approximately constant while the eye is open, but will rapidly fall to zero when a blink is taking place. So, based on EAR, eye blink can be detected i.e., If EAR <EYE-AR-THRESH , eye blink is detected Else No blink detected Where EYE-AR-THRESH corresponds to the threshold eye aspect ratio, EAR measure of 5 is considered.

C. Gaze Detection

Similarly for the gaze detection part, the eye gaze ratio for each eye is calculated. Here the eye region is taken into consideration rather than the distance measures. This region is processed and the iris is distinguished from the rest of the eye. Based on the position of iris being either in the left part or right part of the eye, gaze ratio is calculated.

A gaze ratio below 0.9 corresponds to left gaze, a gaze ratio above 1.7 denotes right gaze and the value between 0.9 and 1.7 denotes center gaze.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Fig. 4. Equation 1

D. Integration with keyboard layout design UI

Inorder to meet user expectations and support the effective functionality of our application, the above sections are integrated with a keyboard design layout. Here tkinter, a python binding GUI toolkit is used. For the keyboard layout, this paper includes three different keyboard designs(as shown in Fig 8,9,10). Based on Qwerty, Maltron and Colemak respectively. Qwerty provides user familiarity while Maltron and Colemak are based on frequency of use of characters.The three keyboard layouts are analyzed and compared.

- Qwerty Keyboard Layout
- Maltron Keyboard Layout
- Colemak Keyboard Layout

Out of the these, QWERTY keyboard layout(shown in Fig. 5) is by far the most popular, and it is the only one that is not limited to a single location. The Maltron (Fig. 6) letter arrangement is based on substantial frequency of use (FOU) analysis, and it places the most frequently used letters and functions where fingers and thumbs can reach them easily and comfortably. Another popular alternative to the standard QWERTY layout is the Colemak layout(shown in Fig. 7), which provides a more familiar transition for users who are used to the standard layout. Colemak is a modern alternative to the QWERTY It uses the QWERTY layout as a foundation, modifying the positions of 17 keys while keeping the rest positions of most non-alphabetic characters and many famous keyboard shortcuts, ostensibly making it easier to learn for those who already type in first layout. It shares numerous design aims, including decreasing finger path distance and utilizing the home row prominently. One-Way Anova test is done to compare the three keyboard layouts. Character per minute (cpm) and error rate (er) are taken as independent variables.

Furthermore to improve the ease of use, auto complete suggestions are added to the keyboard. This is done with fast autocomplete using a data set of frequently used words.

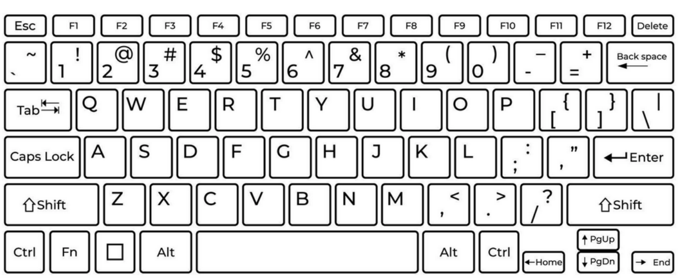

Fig. 5. Qwerty keyboard layout

V. Human Computer Interaction Principles

Our application follows some of Nielsen's ten heuristics principles and Shneiderman's eight golden rules. Guidelines followed are enlisted below as follows.

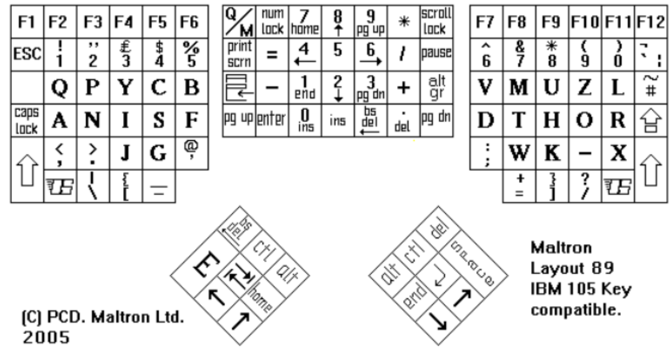

Fig. 6. Maltron keyboard layout

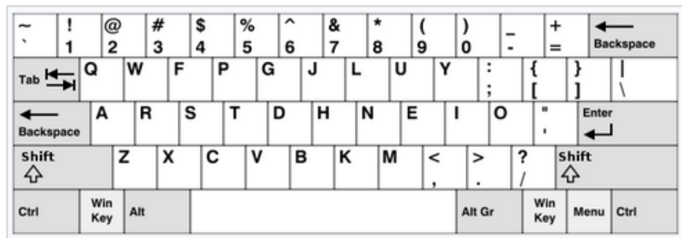

Fig. 7. Colemak keyboard layout

A. Nielsen's ten heuristics

Nielsen's heuristics are empirically based derivations. Widely used by the Usability professionals (including the Interface designers), and they quickly identify likely interface design problems in an application. Method suggested by Nielsen is popular because of its simplicity and low cost.

1) Match between System and the Real World: The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow the real-world conventions, making information appear in a natural and logical order. Here, in our application every stage of character selection is shown clearly using key highlights. And the selected character is displayed on screen once the user chooses it. Thereby, information updated in real world conventions.

2) Aesthetic and Minimalist Design: Simplicity is equal to the efficiency which is equal to the elegance which is equal to beauty that is the aesthetic algorithm in minimalism. Our application provides the most aesthetic and simple keyboard with minimal design with most used character keys. Therefore, efficient communication dialogue between the user and the interface is ensured.

3) Visibility of System Status: The system always keeps users informed about the system state and provides relevant feedback. The application on detection of left or right gaze makes it visible to the user that it is detected, providing conformation.

B. Shneiderman's eight golden rules

These rules are useful in heuristic evaluation to identify the GUIs that fall out of normal 'pattern'. The guidelines can be used to rate GUI's as good or bad.

1) Strive for Consistency: Users need to be able to do the same thing the same way that they have been doing every time. Interfaces need to exhibit 'consistent' quality across screens/applications both visually as well as behaviorally. Here, in our application the process followed for selecting a character in the keyboard layout is fixed. Therefore, the user follows the same process that they have been doing while selecting character keys on the keyboard.

2) Support Internal Locus of Control: This guideline allow users to always feel 'in control' of the system and of the situation. User should believe that they are controlling the system and not the other way around. This is achieved by more opportunities for 'interactions'. Our application is completely based on user interaction i.e, every step from selecting a part of the keyboard to selecting the aspired character is completely under user eye interaction with the application. Therefore, this guideline is ensured.

VI. Results and Analysis

The python, Opencv, fast-autocomplete, dlib libraries along with tkinker gui is used to detect the face, eye, eye blink and eye gaze to print the characters on the screen along with the word suggestions for the current word in the text box using the virtual keyboard of different layouts like qwerty, maltron and colemak. A box on the top right of the interface shows the current gaze direction of the user, whether it is right, left or center. In the following three images, Fig 8, 9, 10 the respective qwerty with right gaze, maltron with left gaze and the colemak with center gaze are shown.

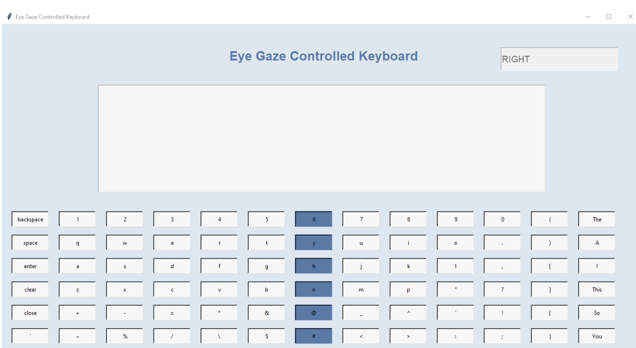

Fig. 8. GUI for Qwerty keyboard layout

As the working of the application, already seen in the methodology, Fig. 12 shows the layout when a text is printed on the screen with the help of the gaze, Word suggestions is provided on the rightmost column of all the layouts, as shown in Fig 11. As an example, qwerty layout is shown in Fig 12. The process of printing the character on the text box of the layout is the same for all the keyboard layouts.

Empirical Analysis

Empirical analysis is a method of studying and interpreting data that is based on evidence. Rather than ideas and con-

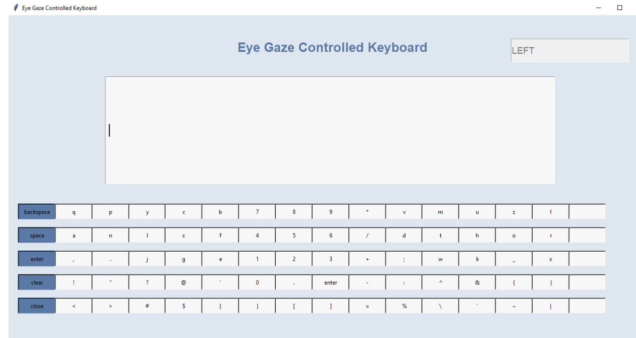

Fig. 9. GUI for Maltron Keyboard layout

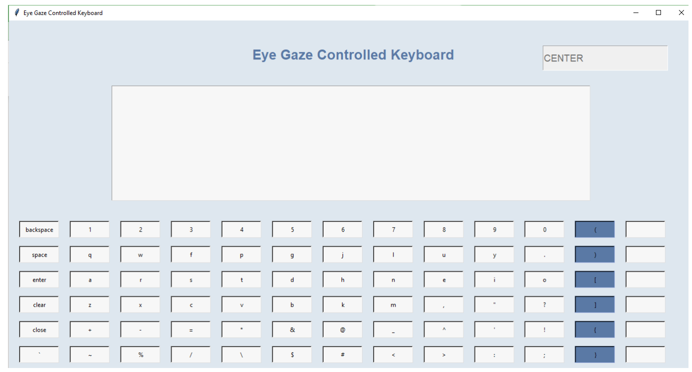

Fig. 10. GUI for Colemark keyboard layout

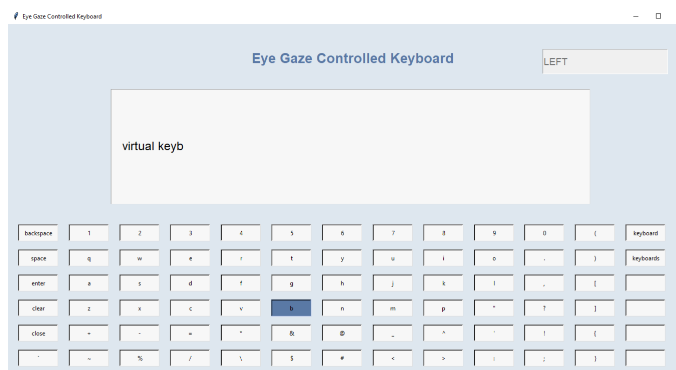

Fig. 11. Word Suggestion on right for the word 'Keyboard'

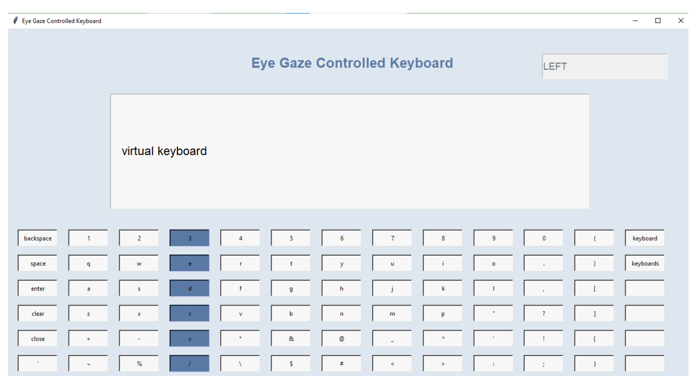

Fig. 12. Complete Output of the word "virtual keyboard"

TABLE II
AVERAGE CPM AND ER

Layout	cpm	er
QWERTY	8.1	0.023
Maltron	8	0.022
Colemak	6.75	0.013

ceptions, the empirical approach relies on real-world facts, measures, and outcomes.The analysis of user behavior gives a better visualization of the application. We asked people to try our application and asked them to answer a few questions based on their experience. The following questions are asked to around 30 people, the results are recorded and analyzed well.

1) Which keyboard layout do you prefer?
   Majority of users preferred COLEMAK keyboard layout (fig 13). Also Colemak has the least error rate as compared to other two layouts.
2) How many characters per minute are you able to print?
   The average cpm is 8.1, 8 and 6.75 (as seen in Table II) for the three layouts respectively.
3) What is the error rate for the given text?
   The average er is 0.31, 0.30 and 0.20 (as seen in Table II) for the three layouts respectively.
4) How long did it take you to get familiar with the layout? (Answer in number of trails)
   As per user reviews, this application takes time to adapt. Once, user adopts, they can make best use of it. On average, an ideal user takes around 6 trails to get familiar with the layout.
5) How to improve the performance of our application?
   Decreasing latency rate of keyboard part highlight, might increase the efficiency of selecting key. This application is not suitable for people wearing spectacles. Takes time to adopt and error rate is high for cold start users. However, error rate is controlled as the number of trails of user increase. This application is limited to python users, extending it to mobile and web users will be appreciable.
6) How do you like to rate our application?
   The average user rating for this application is 3.6 out of 5 (Fig. 14).

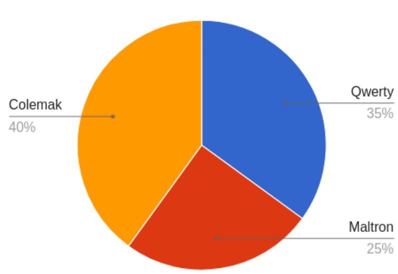

Fig. 13. User preference

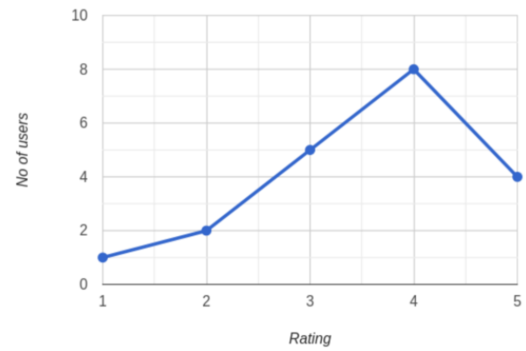

Fig. 14. User Rating

ANOVA Results

The results obtained for character per minute and error rate is given in figure 15 and figure 16 respectively. Taking alpha value of 0.05, f critical of value 3.16. For both er and cpm it is seen that f value is lower than f critical and also p value is greater than 0.05. Hence the null hypothesis is not rejected.

Data Summary

Groups	N	Mean	Std. Dev.	Std. Error
Group 1	20	8.1	4.2907	0.9594
Group 2	20	8	3.5094	0.7847
Group 3	20	6.75	2.8631	0.6402

ANOVA Summary

Source	Degrees of Freedom DF	Sum of Squares SS	Mean Square MS	F-Stat	P-Value
Between Groups	2	22.6333	11.3167	0.8722	0.4235
Within Groups	57	739.5434	12.9744		
Total:	59	762.1767			

Fig. 15. Anova results of cpm for qwerty, maltron and colmak respectively

Data Summary

Groups	N	Mean	Std. Dev.	Std. Error
Group 1	20	0.3196	0.2705	0.0605
Group 2	20	0.308	0.1476	0.033
Group 3	20	0.206	0.1543	0.0345

ANOVA Summary

Source	Degrees of Freedom DF	Sum of Squares SS	Mean Square MS	F-Stat	P-Value
Between Groups	2	0.1563	0.0781	1.974	0.1483
Within Groups	57	2.2565	0.0396		
Total:	59	2.4128			

Fig. 16. Anova results of er for qwerty, maltron and colmak respectively

GANTT CHART

Fig 17 shows Gantt chart of the implementation of the application.

INDIVIDUAL CONTRIBUTION

Patchipulusu Sindhu: Qwerty Keyboard Layout and Gaze Detection.

Alina Melony Pinto: Maltron Keyboard Layout, Blink Detection and Word Suggestion.

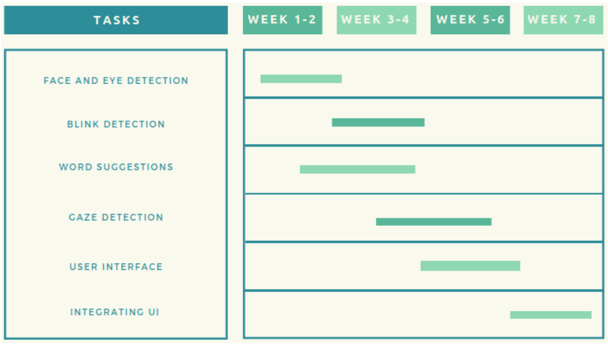

Fig. 17. Gantt Chart

Sanjeevi Meghana: Colemak Keyboard Layout and Face Detection.

Three of us worked together to integrate keyboard layouts with GUI.

VII. Conclusion and Future Work

This paper attempts to create an eye gaze controlled virtual keyboard where based on user eye gaze movement, columns are highlighted and using eye blink a particular key in the keyboard is selected. This application is then integrated with Tkinter, a python based GUI. Thereby, proving Human Computer Interaction. This paper also includes real time feedback from 20 users and based on the users feedback character per minute(cpm) and error rate(er) of all the three keyboard layouts are calculated. Most of the users prefer Colemak keyboard as it provides a more familiar transition for users who are used to the standard layout. Also, users claimed that it takes around 6 trails to get used to the application and so error rate is high for cold start user and eventually this value decreases as user gets familiar. Users also claimed that Auto complete feature of keyboard helped them to print letters faster. This application is not suitable to people wearing spectacles. Decreasing latency of keyboard highlight might increase efficiency. This work can future extended to mobile application as well.

References

[1] D. R. S. Dhananjay Theckedath, "Study of affect computing techniques in human computer interface." [Online]. Available: https://www.iosrjournals.org/iosr-jce/papers/Conf.17014-2017/Volume-2/9.%2038-44.pdf?id=7557

[2] B. Flores, "Quadriplegia - science topic."

[3] U. Garg, K. K. Ghanshala, R. Joshi, and R. Chauhan, "Design and implementation of smart wheelchair for quadriplegia patients using iot," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018, pp. 106–110.

[4] R. Islam, S. Rahman, and A. Sarkar, "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 (ACMI)*, 2021, pp. 1–6.

[5] A. Z. Attiah and E. F. Khairullah, "Eye-blink detection system for virtual keyboard," in *2021 National Computing Colleges Conference (NCCC)*, 2021, pp. 1–6.

[6] P. Chakraborty, D. Roy, Z. Zahid, and S. Rahman, "Eye gaze controlled virtual keyboard," *International Journal of Recent Technology and Engineering*, vol. 8, pp. 3264–3269, 11 2019.

[7] V. I. Saraswati, R. Sigit, and T. Harsono, "Eye gaze system to operate virtual keyboard," in *2016 International Electronics Symposium (IES)*, 2016, pp. 175–179.

[8] A. P. UG Students, "Eyeblink controlled virtual keyboard using raspberry pi." [Online]. Available: https://www.irjet.net/archives/V7/i3/IRJET-V7I3269.pdf