NIJEC 5 (1) (2026) 13-39

NUML International Journal of Engineering and Computing

Journal Homepage: NIJEC.NUML.edu.pk

Print ISSN: 2788-9629

Virtual Compact Two-Thumbs the QWERTY Keyboard for Blind and Vision-impaired Users

[a]Abdul Ghafar*, Wasim Iqbal[b]

[a] Superior University, Department of Computer Science, Lahore, Pakistan
[b] University of Management & Technology, HSM, Lahore, Pakistan

ARTICLE INFO

Article History:
Received 5 March 2026
Accepted 30 May 2026
Available Online 10 June 2026

Keywords:
QWERTY Keyboard, Mobile Virtual Keyboard, Compact Keyword, Keyboard for blinds

Funding:
This research received no specific grant from any agency in public, commercial or non-for-profit sector

Conflict of Interest:
The author has declared no potential conflicts of interest and falsification/fabrication of data with respect to the research, authorship, and/or publication of this article.

ABSTRACT

A physical keyboard is the most favourite input device for text data, but for mobile devices and terminals, it becomes difficult to use a physical keyboard to input data. Therefore, virtual keyboards become the ultimate choice for such devices. Due to the small screen size, it is also not desirable to have a full-sized standard virtual keyboard, so a compact virtual keyboard with a few keys is an essential choice for mobile devices or terminals to improve typing as well as use the mobile screen efficiently. Using a compact-designed the QWERTY layout gives a more accurate typing experience as compared to other layouts. A compact design with the QWERTY layout is implemented with different key combinations and key events to improve the typing experience. In this proposed virtual compact keyboard design, the QWERTY layout is suggested with a limited number of keys with various key combinations and key events such as tapping, long press, and fling. It also manages the screen size effectively with a few keys to utilize the minimum of the screen size, as well as having enough size keys to reduce invalid key errors. It also offers sight-free typing and can be an effective solution for users who are blind or have vision impairments. Blinds and vision-impaired users can easily type text with less movement of the thumb and improved WPM rate. The results also show the expected improvement in WPM rate in the next iteration of the same task. The proposed virtual compact-keyboard design is tested for usability evaluation by measuring functionality, efficiency, and satisfaction using the After Scenario Questionnaire (ASR). It shows an overall effectiveness of 80.23% for vision-impaired users as compared to 72.08% for totally blind users. The overall satisfaction results show a score of 4.61 from 7-point Likert scale for operating the keyboard by users who are vision-impaired or blind.

1. Introduction

Input data or typing is one of the most crucial functions of a computer, without which one never gets the desired output after performing the required processing. A physical keyboard is one of the favourite input devices for this purpose. The structure of the physical keyboard is still under discussion to avoid hand injuries and other body health issues. Researchers suggest different variations by modifying the standard structure by reducing the number of keys to make it effective for health and to ease typing. These solutions are unable to provide the fastest ways to enter data

https://doi.org/10.52015/ygv86d22

[a] Corresponding Author : Abdul Ghafar abdul.ghafar@umt.edu.pk

[1]. Due to the limitations of physical keyboards, especially with mobile devices or terminals. Virtual keyboards have gained popularity, but it never eliminates the limitations of physical keyboards.

One of the most critical areas of keyboard design is to provide a solution to people with low vision, impaired vision, or even blind users, as well as handicapped or disabled people. They are unable to type by using a standard keyboard, so designing a keyboard that requires less finger and hand movement will encourage them to type with minimal typing errors [2]. Various solutions, such as a one-handed keyboard or keyboards with limited keys, are suggested to help people with low vision or who are blind.

With the increase in mobile users, the importance of virtual keyboard has increased. Due to small-sized mobile devices or terminals, the full standard virtual keyboard is not effective for the blind to type without error. They are unable to click the right button due to the small-sized keys. Such keyboards cover most of the screen, so tapping the right key is challenging even by a normal user [2]. To reduce the chance of tapping an inaccurate button and committing typing mistakes, virtual keyboards are designed with a limited set of keys. These keys may have multiple functions against a single key, but their performance and accuracy are low compared to a standard keyboard. Mobile virtual keyboards always give fewer WPM than a full-sized standard QWERTY keyboard [3].

Several keyboard layouts are implemented to enter text, with the QWERTY being the most common. It provides accuracy and speed of typing, and is considered one of the fastest methods of entering text data [4]. Different non-standard layouts are also common in mobile devices, but they are not effective for typing with accuracy and speed. They also require extensive training of users to achieve acceptable performance, so virtual keyboards with the QWERTY layout are still preferable than any other non-standard layout [5]. Entering text using a mobile is not an easy task due to its physical constraints. These constraints include screen size, keyboard layout, and key structure. Standard QWERTY keyboard in mobile devices causes a lot of difficulty due to its small-sized keys, and to navigate fingers or thumbs over half of the mobile screen. These will cause a high rate of typing mistakes and reduce the WPM rate [2]. While typing, the user needs to focus on keyboard keys and the content. All of these reduce words-per-minute speed by using these full-sized QWERTY keyboards [6]. So compact keywords are required with enough space to press the key without typing mistakes. Implementing the QWERTY layout with compact keyboards in mobile devices or terminal screens gives a reasonable typing experience. Various other non-standard keyboard layouts are suggested for mobile devices to utilize the small screen effectively. These solutions are inefficient as compared to the standard QWERTY layout. These also require extensive user training before replacing previous applications. These compact keyboards with fewer keys are also effective for the blind, vision-impaired, or disabled. These special users can easily memorize the limited set of keys to tap. Braille is one of the best solutions for the blind, but it is an expensive solution and needs much training to implement [7].

One of the most common problems with the standard QWERTY keyboard is health issues, such as hand injuries or other muscular pains, due to excessive muscular movements. Consistent extraordinary movement of fingers and the position of arms will cause such health issues [1]. It is also not a good choice for disabled or handicapped people with a single hand or having fewer fingers in one hand. To eliminate these muscular strains and to provide an opportunity for the special users to type data, a compact keyboard with fewer keys is a preferable option. The one-handed keyboard proposed in [8] is good, but it is good choice for a person with fewer fingers in one hand. It also becomes difficult to type with its virtual keyboard on a mobile, especially.

Using limited finger movements with a physical or virtual keyboard can help avoid physical health issues. Designing a keyboard with fewer keys is preferable for body health. It is also helpful in improving typing speed if embedded with the traditional QWERTY layout. The T9 keyboard improves typing speed by guessing dictionary words. It must require multiple clicks on the same key, or combine multiple keys for a single character. Chording is also a helpful feature to generate a letter with multiple keys in a compact keyboard, but these users need to learn the complex combinations of chording keys. [1]

The use of thumbs is more effective in most hand-held and sight-free devices, whereas other fingers, while holding the device, are unable to type properly [9]. The typing speed with thumbs is more reasonable than using fingers with a virtual mobile keyboard [10]. It can be an effective solution, but it also requires fewer on-screen keys to avoid typing errors, as pressing the keys with thumbs requires attention. It would also be an effective and supportive solution for blind, vision-impaired, or disabled users [6]. Typing with two thumbs also gives support to hold and fix mobile devices in the hands. Therefore, blind or vision-impaired users will be able to perform fixed stroke on the screen to type a specific character.

A modified keyboard designed in [11] can be an effective solution for physically handicapped users who are unable to use their hands or arms. These designs include several complex key structures that are entirely different from the QWERTY layout. Therefore, it takes extra time in key navigation and learning these complex key combinations.

For effective use of a small-sized mobile screen, a single-row keyboard using the QWERTY layout is designed in [5]. It facilitates typing with minimum finger movement. The size of keys becomes small enough to fit in a single row, which largely affects the accuracy. This solution is only effective with fixed terminal devices, where the user has free fingers to type. For handheld devices, it remains an infeasible and ineffective solution.

Feedback or hints are ways to provide error-free typing in mobile devices. These options provide an opportunity for sight-free typing, which helps to avoid eye strain with fewer eye movements. A virtual word-by-word feedback keyboard with the QWERTY layout is proposed in [12], to improve typing speed. But it is again difficult to type with small-sized keys on mobiles with a full-size QWERTY keyboard.

Specialized typing devices are also introduced to enter content by special users, but these input devices are not efficient. They also require extra training time and required skills before use. An input device based on pressure sensors on fingers using a two-step input method enables the blind or vision impaired users to enter data. It offers sight-free typing and is also effective for typing while walking, but it is not easy to hold and wear this device. It shows only 10 WPM even after extensive training. [13]

This research aims to propose a virtual compact keyboard for mobile devices that requires two thumbs to type text. This proposed solution also utilizes the screen effectively, using eight keys only in two rows to enter content with various combinations of character keys and control keys. Various key events, including tap, long press, or fling, are performed to type different character keys. Its design is implemented with the standard QWERTY layout with the hunt-and-peck technique to improve typing speed and accuracy [8]. Due to its compact design, by fitting character keys in a single row, it would be a better solution for the blind and vision-impaired users to enter data with a minimum error rate. In the future, more functionalities will be added to facilitate its use with one thumb by holding the mobile in one hand in a more convenient way.

Text typing using a virtual keyboard on a mobile device is one of the most difficult and complex tasks. Most of these virtual keyboards normally cover nearly half of the screen, as the standard Microsoft SwiftKey Keyboard [14] in the Android system, shown in Figure 1, covers half of the screen, so there would not be enough space for user applications. Some of these keyboards provide a non-standard keyboard instead of the most famous, the QWERTY layout. All of these keyboards give a speed of fewer words per minute on mobile devices. Few of these virtual keyboards offer the QWERTY layout to improve speed, but due to small mobile screens, it becomes difficult to operate with small-sized keys. They also cause a lot of typing errors due to difficulty in clicking the required key. They also do not facilitate sight-free typing. As these keyboards are more difficult for a normal user, they are totally infeasible for blind, vision-impaired, or disabled users. Due to excessive use of mobile devices by special users as well, there is an urgent need to design a compact QWERTY keyboard that facilitates error-free and sight-free typing. At the same time, this design is equally preferable for all types of users. The Android system provides TalkBack [15] as an accessibility feature for blind and vision-impaired users. It can read the screen to provide support to special users, confirming the action they want to perform with a second tap on the screen.

The objective of this research is to design a virtual and compact keyboard with the QWERTY layout with fixed keys in two rows, as shown in Figure 2, for blind and vision-impaired users. It is also used to eliminate the side effects of the conventional QWERTY layout on mobile devices that are not feasible for special users, as shown in Figure 1. Blind and vision-impaired users can enter data easily with the proposed keyboard. It needs less thumb movement and allows users to enter data without major typing errors. It is also effective to avoid hand and arm injuries with fewer thumb movements. Another objective of designing this virtual keyboard is to provide maximum mobile screen space for user applications and enable them to utilize the screen more effectively. It also provides an opportunity for sight-free typing by just focusing on the main area of the screen. It allows a user to enter data with less physical movement of thumbs.

The following research questions are formulated to design a virtual compact keyboard for blind and vision-impaired users to identify its feasibility for entering text without major critical typing mistakes.

a. Is a two-row compact keyboard used with two thumbs effective in typing for blind or vision-impaired users?
b. Does this proposed compact keyboard give improved typing compared to other virtual mobile keyboards?
c. How does the proposed virtual keyboard improve the speed and accuracy of sight-free typing by normal users?
d. Is a small-sized keyboard effective in improving screen usage?

The rest of the paper is organized as follows: Section 2 describes the past research in this field and identifies the research gap to initiate this proposed keyboard design. Section 3 describes the methodology to implement the proposed keyboard for the blind and vision-impaired users. And then Section 4 defines the experiment strategy for the proposed system. Section 5 describes the structure of usability evaluation to measure the effectiveness, efficiency, and user satisfaction, whereas Section 6 explains the results of experiments to evaluate usability. Section 7 concludes the topic and elaborates on the future work with the proposed system's limitations.

2. Literature review

A compact virtual mobile keyboard with a standard QWERTY layout is proposed to facilitate entering content with comfort and ease on mobile devices. A compact keyboard with a smaller number of keys, operating with two thumbs, is the true need for a mobile device with a limited screen size. It also facilitates blind and vision-impaired users to enter content. Several solutions with more precise designs are proposed and implemented by different researchers to facilitate blind and vision-impaired users in minimizing typing errors.

A compact one-handed-half QWERTY keyboard is designed in [3] for the disabled by implementing the standard QWERTY layout and hunt-and-peck technique in a physical half-size keyboard. A special key is used to switch to the other part of the keyboard. After practicing for 10 sessions, it shows a 33 WPM. One of the major limitations is its larger size, which covers most of the mobile screen in virtual mode, providing a limited view for written text. This solution is also not feasible for the blind due to the large number of keys on the virtual keyboard.

A virtual two-thumb keyboard, named Senorita, for mobile devices was proposed in [2]. It is an 11-key chorded keyboard in a single row. All of the chorded alphanumeric keys are arranged in a single row. It shows an average of 14 WPM as compared to the simple QWERTY keyboard. It shows reasonable outputs in terms of typing speed and accuracy after providing proper training to users. Its solution is also effective even for the blind and the disabled. One of the core difficulties of using it is learning chord keys compared to the standard QWERTY keyboard. A specific keyboard key is generated by memorizing different chords, sets, or combinations of keys. Any invalid combination of keys will cause a typing mistake.

A virtual one-handed keyboard, ThumbStroke, is proposed in [10] that helps to type without focusing the physical keyboard. The disabled can also perform sight-free typing by holding the device in their hand and entering text with their thumb. His design includes multi-function buttons, just modifying the simple T9 mobile keyboard, and recognizing the gesture of finger movement angle. The results are not satisfactory than the QWERTY keyboard, but it performs better than other mobile keyboard layouts. It is an effective solution for sight-free typing with limited thumb movement around the central character compared to the standard QWERTY keyboard. Pressing a single key multiple times within a short time causes slow typing speed, as inherited from the T9 keyboard layout. There are several typing errors if the thumb movement is not valid. Adjusting all 26 alpha keys in a small circular area is difficult to manage in a compact screen size.

A physical keyboard, named Badhon, is proposed in [11] to enter data with the thumb of the foot by the disabled or those with limited physical capabilities. It struggles to minimize the movement within keys by using a bigram and a trigram linguistic model to suggest typing words. It shows an average of nearly 29 wpm by physically handicapped persons. Learning new key arrangements is not easy to implement using the foot thumb, and the selection of proposed bigram and trigram words restricts typing speed and accuracy.

A one-row keyboard is implemented in [5] to minimize the number of keys and leave extra space on the mobile screen. It eliminates the extraordinary key movements using the QWERTY layout. Multiple combinations of keys are implemented to generate a specific character. Each row of the QWERTY layout is displayed using dead keys to facilitate typing with finger movements. Due to the limited number of keys, it becomes difficult to learn the key combinations to get the next row of keys. It becomes much slower to type words if they belong to different sets of keys.

Kim [12] explores a word-by-word dynamic feedback system to reduce eye movement while typing. It demonstrates high-performance sight-free typing compared to a normal character-by-character keyboard. It shows 11% typing improvement compared to other character-by-character keyboards, but it creates a lot of confusion in selecting the right and suitable word while typing. It becomes more difficult if the required word is not available in the suggestions.

Jiang [13] introduces one-handed wearable typing devices, called HiFinger, with sight-free typing. It gives quick, accurate, and comfortable typing by using thumbs and fingers. It is also suitable for typing while walking or moving, but it is not a natural and convenient way to type text. It also needs much practice to learn the location of keys on various fingers and thumbs. These devices are unable to provide an effective solution to the community that cannot afford such an expensive solution.

A clear research gap, after analysing previous studies, is identified to design a compact keyboard with fewer finger movements for blind and vision-impaired users. It would also be an effective and simple typing solution for normal mobile users. One of the major initiatives of this research is to determine the size of the virtual keyboard that reduces complexities for blind and vision-impaired users. Many virtual keyboards cover nearly 50% of the screen, as discussed in the literature review, so a very small portion of the screen is available for user applications. In this way, the user is unable to perform other available operations effectively. Another important aspect of already available keyboards is the small-sized keys that cannot be clicked without due care, as the size of the thumb or finger is larger than the size of the key, which creates an additional complexity in using them. Secondly, the user navigates the entire screen to search for the desired key. It ultimately limits typing speed and increases the complexity of finger movements for blind and vision-impaired users. Some virtual keyboards are compact and can be operated with fewer keys, but those ignore the standard QWERTY layout. These keyboards also require extra training time to achieve the same level of typing speed and accuracy as provided by the QWERTY layout. Normally, mobile virtual keyboards are operated with thumbs, and it becomes also difficult to move the thumb over a large area of the screen.

Another important fact about this study is the increased usage of mobile devices by special users such as the blind and the vision-impaired. They are almost unable to use these keyboards to enter data on mobile devices. Therefore, the proposed compact keyboard with limited thumb movements would be an effective and useful component in future mobile operating systems. In this way, the proposed keyboard design also gives sight-free typing, as there would be no need to focus on the keys while typing. It also facilitates blind and vision-impaired users to restrict finger or thumb movements excessively over the screen. It takes less time to learn thumb movement for a specific key movement compared to a braille device, which needs extensive training to read or write text. The proposed keyboard is implemented with fixed keys in two rows, accessible using a one-step thumb movement of both hands. The proposed solution is also effective for normal mobile users with improved typing speed and fewer thumb movements. It also provides maximum screen space to interact with the application. Mobiles have become easily accessible to everyone, including special users, so providing a solution to enter data with less effort is a major contribution in the field of research.

Keyboard design is important in analyzing the typing performance as well as the user's body health. Using a keyboard affects the body's health and the WPM typing score if excessive finger movements are required while typing. Brandon [1] studies the importance of keyboard design to determine its effects on body health and results in muscular injuries. He suggests a split keyboard in physical and virtual formats for long typing work. He also determines that it provides much comfort to the hands by maintaining a natural angle that strains less on the neck and shoulder. He explains that a keyboard with less finger movement is crucial to protect body health. While typing with a keyboard, the right muscle movement is essential to avoid hand injuries. This proposed keyboard is also designed to maintain user health by limiting thumb movements. In this way, there would be less chance of hand or arm injuries, as well as avoiding eye strains with sight-free typing.

3. Methodology

The purpose of this study is to design a virtual compact keyboard for blind and vision-impaired users to facilitate them to type without extraordinary movement of thumbs and with fewer typing errors.

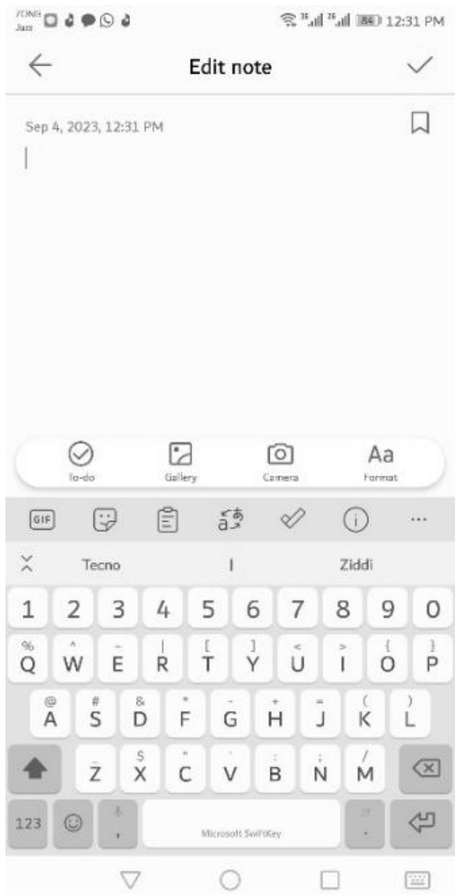

Figure 1: Virtual keyboard covers half of the screen

3.1. Proposed System

A two-thumb compact keyboard with five main keys, a space key, and two control keys is proposed for Android to achieve the objectives discussed in the previous section. The proposed keyboard design is based on the standard QWERTY layout, and its layout is divided into two rows. The first row contains the main keys, whereas the control keys and the space key are in the second row. The QWERTY layout is divided into two key sections with three keysets of six keys. These keysets can be accessed by using various combinations of tap, long press, and fling events with the CONTROL key. It also includes one keyset of numeric pad and two keysets of symbol keys, and these keypads can be accessed by using the same CONTROL key. The keyboard is automatically aligned to the bottom of the screen and can be operated with two thumbs only. The size of keys in the proposed keyboard is large enough to eliminate confusion or ambiguity in typing, especially for blind and vision-impaired users.

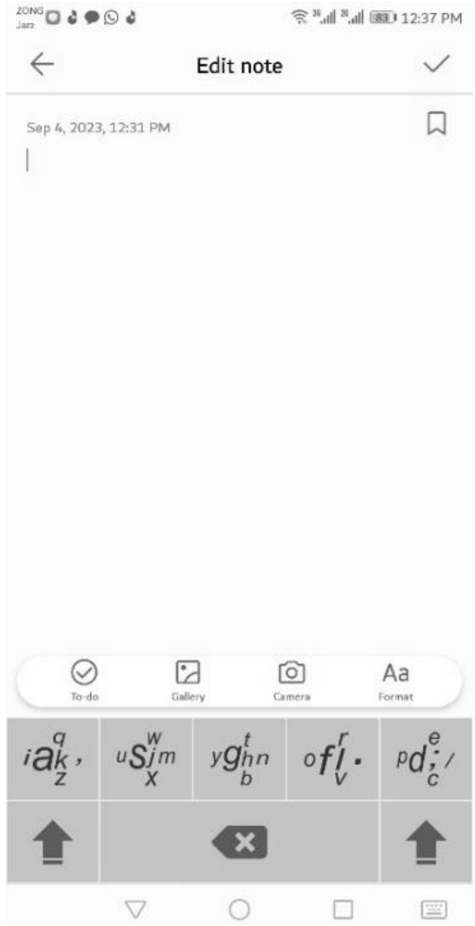

Figure 2: Proposed virtual compact keyboard

3.2. Proposed Keyboard Design

The proposed keyboard is purposefully designed to facilitate typing with limited thumb movements on the screen. It contains 8 keys organized in two rows, including five main keys, space key, and two duplicate control keys, as shown in Figure 2. The first row contains five main keys to type characters, whereas a duplicate control key with the Space key is available in the second row. Control keys are fixed at both corners to create different key combinations. It can be operated either with the right or left thumb. A large-sized space key is located between two control keys. It can be accessed with the left or right thumb. The operator can access any key with a single-step thumb movement. It provides a sight-free layout to enter data without focusing on the keyboard. A single thumb is used to press two or three main keys with two control keys.

By default, the proposed keyboard is set to the ASDFG keyset with lowercase. Left and Right sections of the QWERTY layout can be swapped by using the CONTROL key, whereas top and bottom keyset keys can be accessed with top and down fling events. Each keyset has five keys, which can be used by using the left or right thumb, A and S keys with the left thumb, and F, D, and G keys with the right thumb. In the case of the second section of the keyboard, the right thumb is used to press two keys, and the left thumb is used to tap three keys. In normal mode, the operator can type any letter from the ASDFG set of keys by using a simple tap event, where top fling and down fling events can be used to access keys from top and bottom keysets. Same events can be used to type characters from the second section of the keyboard by tapping the CONTROL key. Right section keysets can also be accessed directly with long-press, left fling, and right fling operations. The middle main key can be pressed either by the left or right thumb for user convenience.

3.3. Navigation to the QWERTY Keysets

The CONTROL key is the main navigation key in the proposed keyboard. Operator taps the CONTROL key to swap between the left and right sections of the QWERTY layout. As the user taps on the main or other key, it automatically swaps to the default keyset. The operator can press the CONTROL key at his convenience, either with his left or right thumb. To hold the right section of the keyboard, the CONTROL key is pressed for a long time. Fling and long-press operations can be performed to access keys in other keysets. The right and Left fling operation with the CONTROL key is used to navigate to numerical, symbol, and alpha keypads in a circular way. The top and down fling event, combined with the CONTROL key, is used to switch the caps lock and shift key. A single tap on the Control key resets the keyboard to the default keyset. Fling down the event on both CONTROL keys will reset all settings of the proposed keyboard.

3.4. Capital Keys

The CONTROL key converts the alpha keys to uppercase or lowercase. By default, available alpha keys are in lower case; top-fling operation on the CONTROL key will work as the SHIFT key to switch to CAPS mode. Normal mode is switched back as the user taps on any capital key. To activate the CAPS LOCK, the down-fling operation is performed in CAPS mode. Then the user will be able to type alpha letters in uppercase. Single-tap event on the CONTROL key will deactivate CAPS LOCK and restore keys to NORMAL mode.

3.5. Keypads in the proposed Keyboard

The proposed keyboard provides an option to type any alphabet, numbers, or symbols. It provides various keypads such as alpha, numeric, and symbol keypads. By default, alpha keys are available, whereas other keypads are available by using various fling events on CONTROL keys, as shown in Table 1. For example, the right-fling operation is used to access the numeric keypad.

Table 1: Key operations to access keysets

Keyset	Key Operation
Alpha Keys (default)	Tap Left Control + Tap Right Control
Capital keys	Top Fling Event on CONTROL key
Caps Lock	Top Fling + Down Fling on CONTROL key
Numeric Keys	Right Fling Event (one time)
Symbols	Right Fling event (two times)

3.5.1 Alpha Keyset

The Alpha keyset is based on the most popular QWERTY keyboard layout, and is divided into left and right sections. Each section is further divided into three rows of keys: top, middle, and bottom rows. The middle row of the left section is the default keyset, and other keysets can be accessed with various key combinations with the CONTROL key. All of these keysets are also presented with the capital case with the CAPS key and the CONTROL key event. Most common punctuation symbols are available in the SPACE key with fling action, whereas other symbols are available in the symbol keyset. The CONTROL key to shift to CAPS Key provides more punctuation symbols on the SPACE key.

A simple tap event on Alpha Keysets is used to get the middle row keys, whereas the top and down fling options give the top and bottom row of the QWERTY keyboard, as shown in Figure 3. For example, top fling returns the QWERTY keys, whereas down fling gives the zxcvb keyset. The CONTROL key or left/right fling and long press event are used to retrieve keys of the right section of the QWERTY keyboard. For example, to fetch k key, long-press the second alpha key, or with the CONTROL key, press a simple tap on the second key to access the same key.

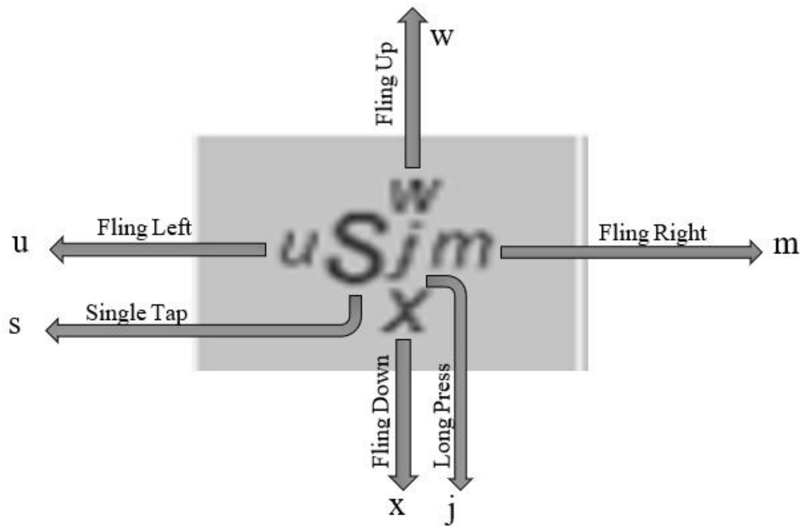

Figure 3: Key operations to access various characters

3.5.2 Number Keypad

The number keypad is just a single row of five keys with numbers as well as various symbols used to type numerical and mathematical equations. The same filing operations used with the Alpha keyset can be used to work with numerical characters. There is no use of the CONTROL key while typing numeric characters. The right two keys in the numerical keys give numerical symbols in a circular way using a long-press event. For example, pressing the rightmost key returns + sign, it returns -, *, /, and = signs on long press it again and again.

3.5.3 Symbol Keypad

All of the common symbols are available in the symbol keypad. There are two symbol keysets that can be accessed by using the CONTROL key. A simple tag event returns the main symbol, whereas other symbols can be accessed using fling and long-press events.

3.6 Thumbs Movement

The proposed keyboard required very little and simple movement of both thumbs. It requires only one step movement in any direction to press or type any character on a key, as shown in Figure 4. The left thumb is fixed on the S key by default, and the right thumb is fixed on the F key by default. The required movement of the thumb in the left or right direction to access any other character key. The left thumb is required to move on the left side to access the A key, whereas the right thumb is required to access the D key. To access the middle G key, it is optional to use left or right thumb.

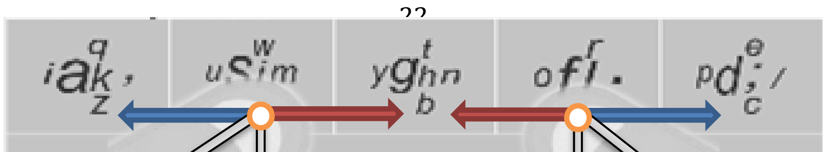

Figure 4: Thumb movement in the proposed keyboard

Control keys and the space key will be accessed by moving any of the thumbs to the bottom row. Either thumb can be moved toward to press the space key as per the convenience of the user. The right thumb is moved downward to access alternate keys of the left side of the keyboard, while the left thumb is normally used to access alternate keys of the right side of the keyboard. Still, it is on the discretion of the user to use either thumb to press the control key to get an alternate behaviour of the character keys.

3.7 Navigation within Keysets

All of the keys in any single keyset are composed of five keys, which can be used to type a letter with a single tap. Fling operation in a different direction is performed to fetch keys in the top and bottom keysets, whereas the right section is either accessed with the CONTROL key or long-press and top/down fling operations. The right two keys are accessed using the right thumb, and the left two keys are available with the left hand thumb, whereas the middle key can be accessed using either right or left thumb, with the convenience of the user.

3.8 Special and Short-cut Keys

To ease the typing process, some special and shortcut keys are also available to perform some specific functions, or to access some particular key from a specific keyset. These keys must be performed in some unique combination of MAIN Keys, CONTROL, or SPACE keys with a particular action on various keys as given in Table 2.

Table 2: Special and short-cut key operations

Action Key	**Short-cut key event**	**Key**
Enter key	Long Press	Space key
Backspace key	Fling Left	Space key
Backspace key	Simple Tap	Backspace key
Right Section middle keyset	Long Press	Alpha key
Right Section upper keyset	Fling Left	Alpha Key
Right Section lower keyset	Fling Right	Alpha Key
Keyboard Right Section	Simple Tap	Control Key
Left Section middle keyset	Single Tap	Alpha Key
Left Section lower keyset	Fling Up	Alpha Key

Action Key	Short-cut key event	Key
Left Section middle keyset	Fling Down	Alpha Key
CAPS Key	Fling Up	Control Key
CAPS Lock	Fling Up + Fling Down	Control Key
Alpha/Numeric/Symbol Keypad	Fling Left/Fling Right	Control Key

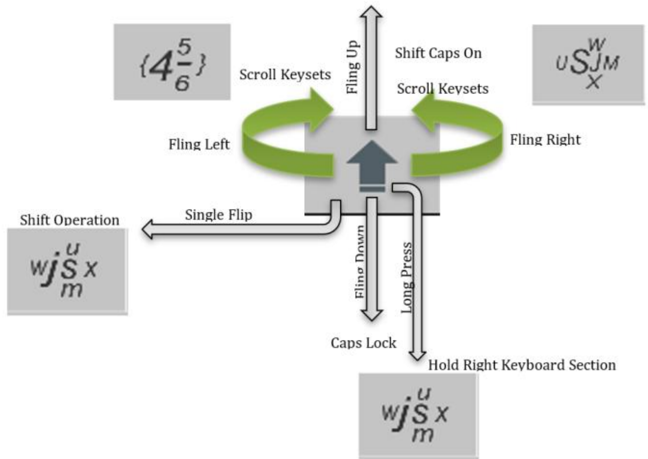

Figure 5: Control key operations in the proposed keyboard

3.9 CONTROL Key

There are two duplicate CONTROL keys to facilitate performing its operation by using the left or right thumb, as shown in Figure 5. The CONTROL key is a multi-function key that is used to perform multiple functions based on a particular action or event. Single-tap swaps the left or right section of the QWERTY layout. Flip event on the CONTROL key is used to access different types of keysets, such as alphabet, numeric, or symbol keypads. The single tap on the CONTROL key can be used to deactivate the previous action. Long-press action is used to activate the capital key option, whereas long-press action with the SPACE key is used to perform the backspace operation. Left and Right CONTROL keys can be used to reset the keyset to the default asdfg keyset row on the left section of the QWERTY layout, as shown in Figure 6.

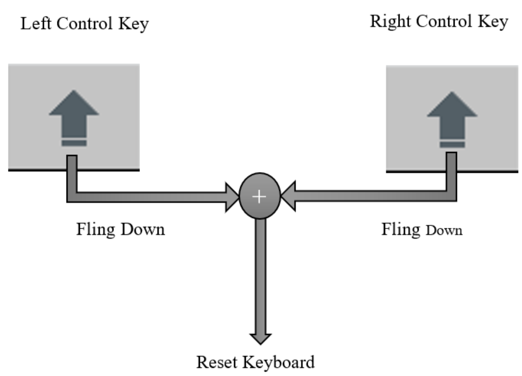

Figure 6: Reset the keyboard using control keys

3.10 SPACE Key

The SPACE key is also a special key that is required to type spaces or some common punctuation symbols and can be used for multiple operations, as shown in Figure 7. By default, it is used to type blank space using a simple tap event, but it can be used to perform multiple other useful functions.

Fling operations are required to type basic punctuation marks. It also acts as the ENTER key against a long-press event. Another important function of the SPACE key is the BACKSPACE key, which can be accessed using the left-fling operation. Once BACKSPACE is activated, the SPACE key behaves as the BACKSPACE unless another key or the CONTROL key is pressed. It is also used to type more punctuation symbols by activating the CAPS key with the CONTROL key.

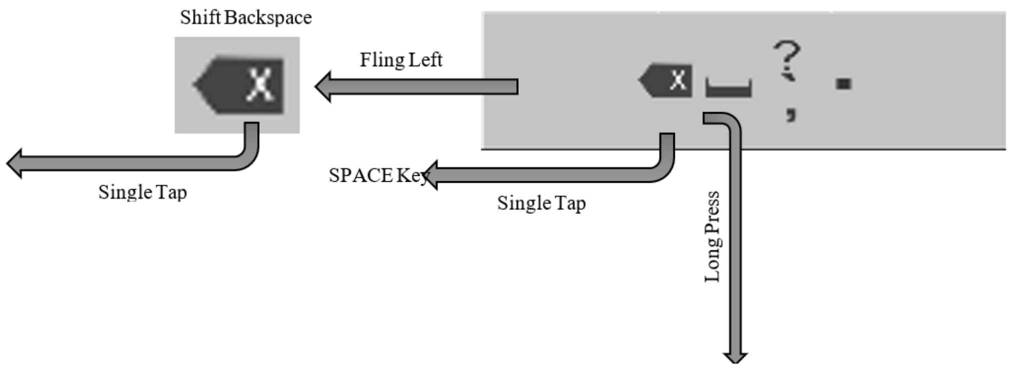

Figure 7: Operations using space key in the proposed keyboard

4. Experimentation

4.1. Proposed Keyboard Installation

The proposed mobile app was developed in Android Studio using Kotlin for the Android Operating system with a touchscreen. It is tested on a virtual emulator Pixel XL API 29 for testing purposes while developing. Then it is installed on Huawei Mate 10 Lite with a 5.9" screen with Android Operating System version 8.0 for testing on a physical device. The proposed keyboard is available in horizontal and vertical layouts, but vertical layout is only allowed for the proposed system testing.

4.2. Target population

The usability evaluation for the proposed solution needs a small group of users from the targeted population. The targeted population or domain is the number of blind and vision-impaired users. The proposed system is tested with 26 users in the targeted domain as described in [16] for specific vision impairments. The selected population includes people with vision impairment who have more than one year of experience using smartphones and laptops. There are 20 male participants, including 6 females, to maintain the gender ratio, but gender discrimination is ignored in this study. The participants are in the range of 21 to 30 years to ensure their learning ability in using smartphones. They are already familiar with the use of smart devices such as phones, tablets, etc. Participants are divided into two age groups: 21-25 years and 25-30 years.

For this study, all respondents are actively using smartphones, 50% also familiar with tablets, and 60% can also operate desktop systems. It is also observed that the usage of smartphones and tablets is more compared to desktop systems. More than 80% users can operate a mobile with the help of voice using voiceover functionality. To compare our proposed system performance, we have selected 10 respondents who are familiar with the Braille system to read and write. All participants are also familiar with the QWERTY layout as well as famous mobile virtual keyboards. All users prefer the QWERTY layout due to the ease of typing.

Blind and vision-impaired users are categorized in various grades as shown in Table 3. In this paper, we have categorized blinds who can read or write printed material, and low vision users who cannot read or write. The 40% of our target population cannot read or write the text, whereas 60% can read or write print. A large percentage of level 1 can read or write the printout, whereas level 3 can read and write the text. In our target population, 6 users belong to blindness level 1, 9 users have a disability of level 2, and the remaining are related to blindness levels 3 to 6. [17]

4.3. Experiment setting

Before starting the test session, all participants had attended 30-minute training sessions for three days to memorize the location of keys. On testing day, they had to listen to the soundtrack three times before starting to type using the proposed application. A smartphone set with the same Android operating system version is provided. A handfree is provided to listen to the same text to facilitate blind and vision-impaired users. Time is set before starting each task. They started to enter text while listening to the soundtrack for the specific task. They are allowed to edit the text if there are some typing mistakes, but not allowed to rewind or stop the soundtrack while typing. All tasks are attempted three times to measure the accuracy accurately. As users have vision impairment, there is a little pause after each word. There is no time limit to finish the task. It is not possible to arrange a test session for all respondents at the same time, so multiple test sessions based on the convenience of the respondents are arranged. Even a single respondent is allowed to perform the test individually.

The same setting is used to perform testing with the Microsoft SwiftKey keyboard without TalkBack and with the TalkBack option [15]. The Microsoft SwiftKey keyboard is the default keyboard in Android mobile phones and is normally used by normal users. For this test, the target population is informed about the Microsoft SwiftKey keyboard by aligning their hand on the screen and guiding the display of the QWERTY keyboard rows. First test is performed without TalkBack, and then the second test is performed using TalkBack. The TalkBack is a special application for the blind that translates every action of the user into voice, so the user can estimate the correct key to confirm the same key by pressing it a second time. It restricts the key, so there will be no typing mistakes. If the user cannot understand the voice properly, then there will be an error. One of the major drawbacks of TalkBack is the extra time required to understand the key before final selection.

Table 1: Blindness disability grades

Blindness Disability Grade	Visual Acuity
Level 1	0.02 and less
Level 2	Less than 0.02 to 0.04
Level 3	Less than 0.04 to 0.06

4.4. Experiment tasks

The proposed system is tested in a controlled environment and divided into three tasks. There will be three rounds for all of the tasks to measure performance accuracy. These tasks are basically three sets of English language words or paragraphs with an increasing level of difficulty. The first task includes simple words as shown below without any punctuation marks. Use of initial capital is not required for test purposes. There are fifteen words in task 1, and these words are divided into three lines. First row belongs to the Left section of the keyboard, second row contains words from the Right section of the keyboard, and third row contains mixed letters from both sections to use the right and left sides of the keyboard simultaneously.

Cat wear fat deer sad

Moon, You Lip Noon look
Petrol Verb Teen Like walk

The second task includes eight simple English phrases and sentences with basic punctuation marks as given below. Use of capital letters and punctuation marks is also not required for this test. There is a total of 30 words in all of these sentences or phrases.

Good Morning
Coffee Beans
Red Shoes
Dark Night
Stay there
I am a boy.
Are you there?
What a nice car!
My color book is here.
I can do it.

The third task includes a simple English paragraph with simple punctuation marks. It includes long and short sentences, including some phrases. Capitalization of words, as well as punctuation where needed, is required for this task. This task includes a total of 32 English language words.

Here is your receipt of hundred rupee and you have to pay it before end of this month. It also includes sales tax as well as service charges.
Thank you so much!

The purpose of these tasks is simply to test the usability of the proposed keyboard using the alphabet keypad to type English language words and paragraphs with simple punctuation and capitalization. There would be a one-minute break between each task. Each round is performed consecutively. The user also has an option to listen to the entire audio of the text before starting that task. All of the tasks are just designed to type the alphabet and basic punctuation marks. The use of numeric and symbol keypads is not included in the test due to the initial phase of implementation and difficulty in memorizing the position of number and symbol keys.

4.5. Usability Evaluation

For the purpose of evaluating the usability of the proposed application, a timer is started before starting each round, and the ending time is recorded when the user finishes the task. A moderator will perform all such activities to record the starting and ending times. He will also align the mobile before starting the next round of that task. There would be a 15-second pause between each round for a short break. In this way, all of the tasks are performed in the same session. The moderator records the number of typing mistakes at the end of each task.

ASQ is required to be filled at the end of the test session to measure the user satisfaction after using the proposed system [18]. All of the filled ASQs and above test reports are compiled in an Excel worksheet at the end of the training session. The test reports include the task number, the number of words in each task, the number of typing mistakes, the total time for the task, and the task completion time.

It is performed by using the ISO 9241-11 standard by measuring factors such as effectiveness, efficiency, and satisfaction in performing the task while testing the proposed keyboard [19]. These factors are measured by using the formulas given below.

4.5.1. Effectiveness

The effectiveness is the percentage of tasks completed successfully, and is calculated by dividing the number of successfully completed tasks by the total number of tasks to be performed, as shown in formula 1 [20]. To evaluate the effectiveness, the following dependent variables, WPM, and Number of typing errors, are recorded during the test. Text entry can be categorized into Correct, Incorrect but fixed, Incorrect but not fixed, and fixed [20]. It is difficult to observe the correct and incorrect if fixed by key strokes, so we have considered them as correct words. The incorrect words not fixed and missed words are considered typing errors. If the required WPM rate is achieved, then that task will be considered successful. WPM rate is the typing speed, i.e., the number of words per minute. It is calculated by using the formula given in equation 2. The acceptable WPM rate for blinds is around 8-9, so for testing the proposed application in its initial phase of development, we have set above half of the given WPM in [21], that is, 6 words/minute. If any participant achieves 7 WPM after subtracting the mistakes, then the task will be performed successfully; otherwise, it will not.

Effectiveness = Total tasks completed / Total tasks undertaken * 100 (1)

WPM Rate = (Total Words – Mistakes) / (Total Time (secs)/60) (2)

4.5.2 Efficiency

Efficiency is another important factor to evaluate the usability of the proposed system. It is based on various resources required to complete tasks, such as time, money, and physical effort, to type the required paragraph within specified constraints. In this study, efficiency is calculated in terms of time using formula 3 given below, where N is the number of tasks and R is the number of Participants [22].

$$Time\ based\ Efficiency = \frac{\sum_{j=1}^{R}\sum_{i=1}^{N} {n_{ij}}/{t_{ij}}}{N x\ R} \qquad (3)$$

Result of each task either successfully completed or not, $n_{ij} = 1$ if task is completed successfully, otherwise $n_{ij} = 0$. Time taken by participant *j* to perform task *i* is represented by t_{ij}.

4.5.3 After Scenario Questionnaire (ASQ)

ASQ is implemented to measure the user satisfaction after using the proposed system. It is easy to evaluate user satisfaction by asking about the usage and user experience of typing with the proposed keyboard. The questionnaire is divided into three sections. The first section is related to ease of typing with the proposed keyboard. The second section is about performing the required typing task within the specified time constraints. Lastly, the third section is related to the level of satisfaction while typing with the proposed keyboard. These questions are ranked on a 7-point Likert scale from 0 (strongly disagree) to 6 (strongly agree) to evaluate the question in the questionnaire [22].

5. Results and Discussion:

The section describes the results of the performance of the proposed keyboard, Microsoft SwiftKey [14], without TalkBack and with the TalkBack option. The proposed keyboard is evaluated in terms of effectiveness, efficiency, and satisfaction for blind and vision-impaired users. The usability of the proposed system is tested for total blindness (Level 1), partial blindness (Level 2), and vision-impaired (Level 3) users by measuring the effectiveness, efficiency, and satisfaction. The results are calculated based on test results. The proposed keyboard performance is compared with other selected keyboard options for an Android mobile phone. To compare the performance of the proposed system, data entry speed WPM and Error Rate are calculated by using the time to finish the tasks, typing errors, and correct entries.

Average WPM speed of the proposed keyboard, Microsoft SwiftKey without TalkBack, and Microsoft SwiftKey with TalkBack, were 7.41 wpm (SD = 1.73), 2.33 wpm (SD = 1.76), and 4.73 wpm (SD = 2.17), respectively, as shown in Figure 8. ANOVA [23] shows a significant effect of WPM speed of the proposed keyboard with single-factor ANOVA as ($F_{2,231}$ = 139.89, P = 0). ANOVA is a statistical technique to compare means of different groups of values to determine significant differences among them. P-value and F-value are determined to identify the significance of group means by comparing threshold values.

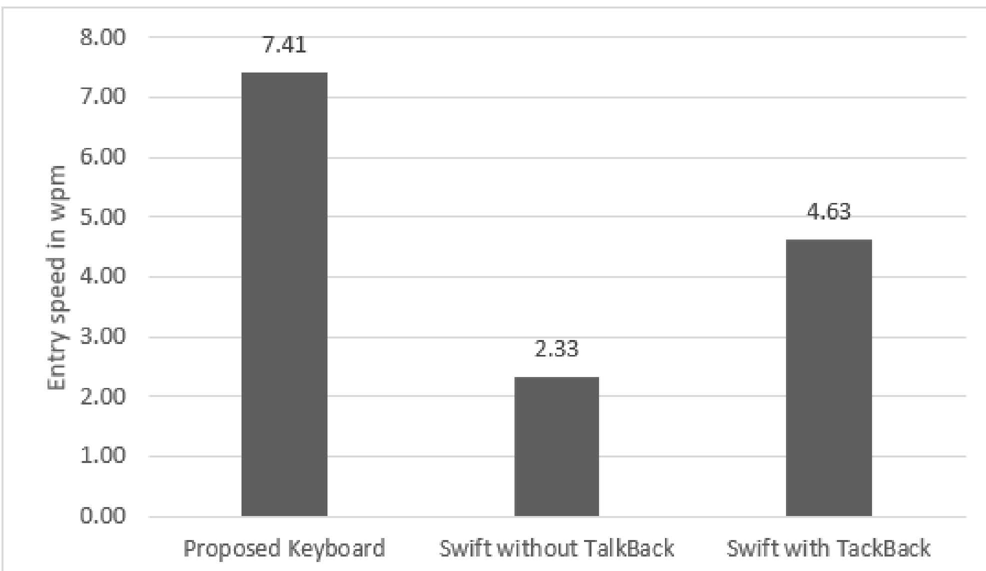

Figure 8: Comparison of typing speed with Microsoft SwiftKey without TalkBack and wihtout TalkBack

The average error rate of the proposed keyboard, Microsoft SwiftKey without TalkBack, and Microsoft SwiftKey with TalkBack, are 24.69% (SD = 8.65), 74.83% (SD = 16.34), and 55.76% (SD = 13.57), respectively. ANOVA shows a significant effect of Error Rate of the proposed keyboard with single factor ANOVA as ($F_{2,231}$ = 285.14, P = 0).

5.1. Effectiveness

The overall effectiveness of the proposed system is measured in terms of time, is 75.31%(SD=8.64) using formula 3, compared to 25.17%(SD=16.34) and 55.76%(SD=13.57) for Microsoft SwiftKey without TalkBack and with TalkBack option. ANOVA shows a significant effect of the effectiveness of the proposed keyboard with single-factor ANOVA ($F_{2,231}$ = 284.31, P = 0). It also shows that the use of the Microsoft SwiftKey keyboard is not feasible for blind and vision-impaired users due to the smaller-sized keys, as well as being unable to align the thumbs with the keys. The TalkBack option gives some support, but again, due to smaller-sized keys, sometimes, the user cannot understand the voice properly before finally pressing the key. The proposed keyboard performs better than the Microsoft SwiftKey without TalkBack and with the TalkBack option. Figure 9 shows the effectiveness of the proposed system for various types of tasks is 73.85%(SD=7.81), 77.74%(SD=7.67), and 74.36%(SD=10.06). It shows the level of difficulty in various tasks. As tasks 1 and 2 contain simple words and phrases without any punctuation marks, there will be no high complexity in typing those words. Task 3 includes some complexity due to long sentences with basic punctuation marks. In the next iteration, the user performs much better compared to the last drill, so they would be able to move their thumbs easily, and overall efficiency is improved in the next iteration of the same task. ANOVA does not show any significant effect of effectiveness for types of tasks performed by the proposed keyboard with the single-factor ANOVA ($F_{2,75} = 1.57$, $P = 0.21$), so the proposed keyboard performs irrespective of content type.

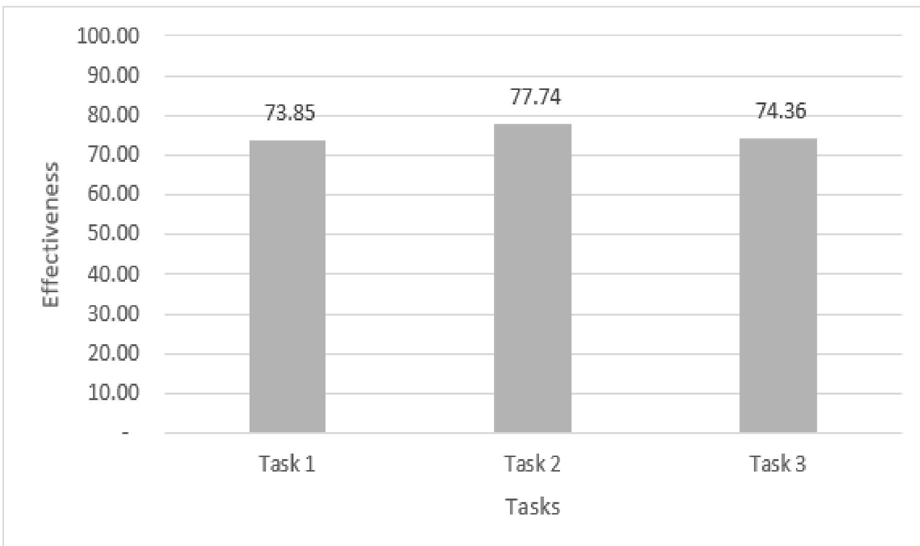

Figure 9: Effectiveness of the proposed system in terms of tasks

The effectiveness of the proposed system in terms of age group is 77.35%(SD=7.49) and 73.28%(SD=9.32), as shown in Figure 10. Younger users are more comfortable with the proposed keyboard and easily learn to apply various key combinations. They are quick to learn and follow the instructions given before the test, as well as their familiarity with using mobile devices for their different routine tasks. Therefore, they also showed more efficiency compared to the higher age group users. The ANOVA shows some significant effect of the effectiveness of using the proposed keyboard by age group with single-factor ANOVA ($F_{1,75} = 4.52$, $P = 0.036$). These younger users have better skills compared to older people, which affects the effectiveness of the proposed keyboard while using it.

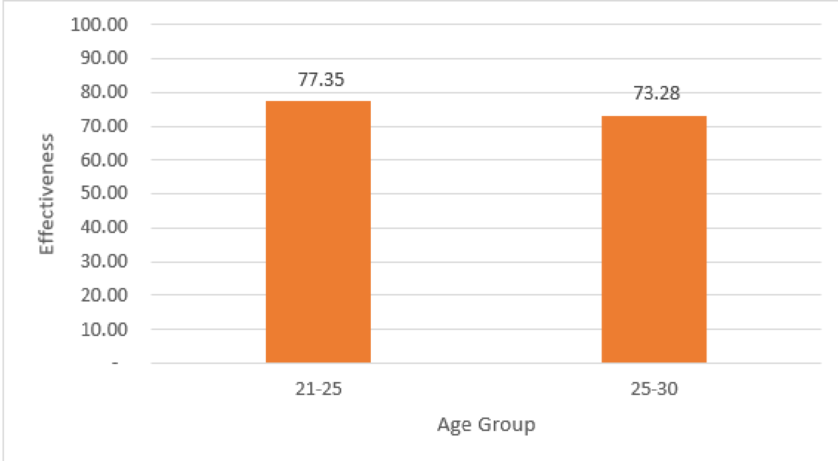

Figure 10: Effectiveness of the proposed system in terms of age group

The most important results for the effectiveness of using the proposed keyboard with different blindness level is 72.08%(SD=8.88), 73.96%(SD=7.55), and 80.23%(SD=7.82), as shown in Figure 11. It is natural, as totally blind users face much difficulty in using any new application. They have to face difficulty adjusting their thumbs to the keys to type properly. They have to pause to listen to words carefully before typing. On the other side, the level 3 blinds are much more comfortable with the proposed system and can align and move their thumbs easily and without much difficulty. The ANOVA shows a very significant effect of effectiveness for different levels of blindness in using the proposed keyboard with single-factor ANOVA ($F_{2,75} = 6.92$, $P = 0.0019$). It means the effectiveness of the proposed keyboard is affected by different types of users, so vision-impaired users performed well compared to blind users. It will become easy for level 1 users after extensive practice, as their results improve in the next round of the same tasks.

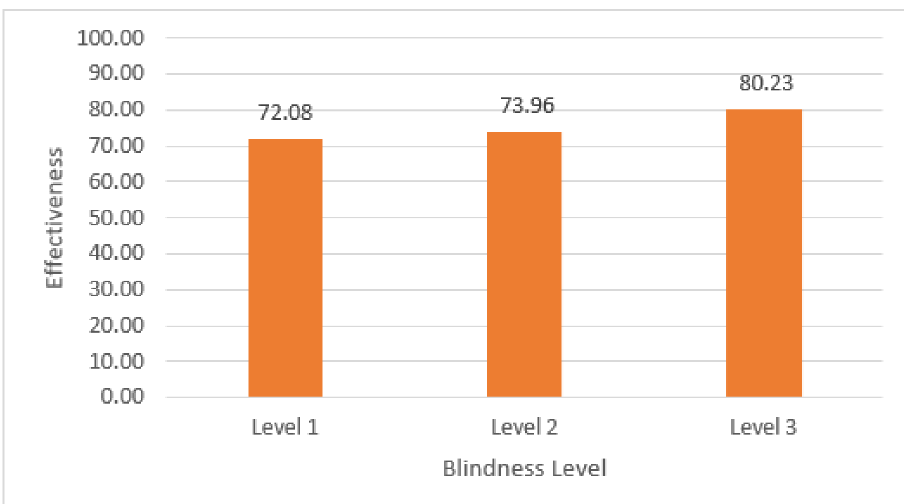

Figure 11: Effectiveness of the proposed system in terms of blindness level

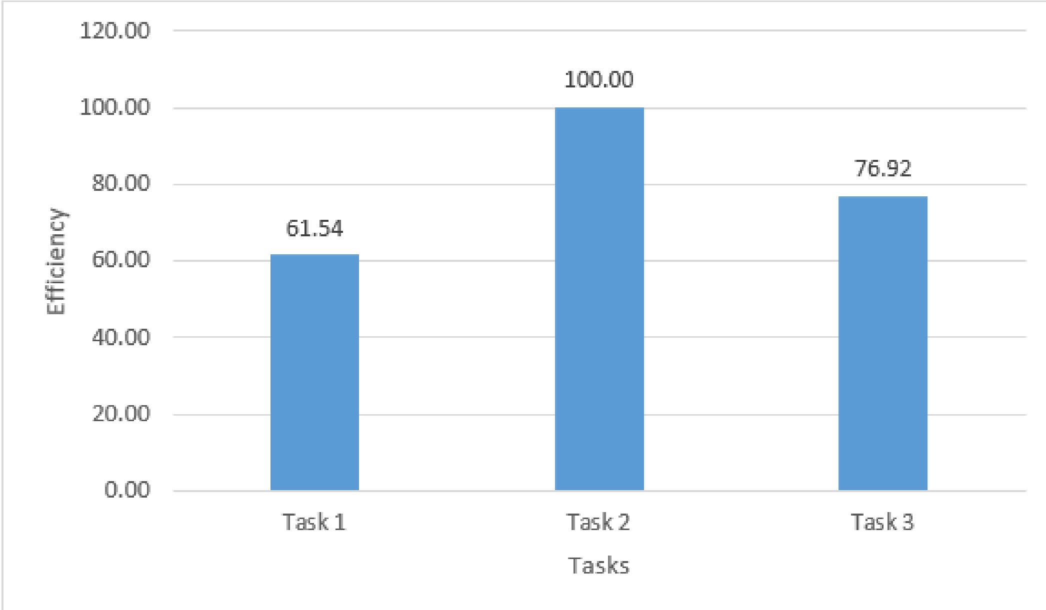

Figure 2: Efficiency of the proposed system in terms of tasks

5.2. Efficiency

The proposed keyboard efficiency can be measured for blindness levels, age groups, and tasks. The overall efficiency of the proposed system is 79.49%(SD=40.64) compared to 11.54%(SD=32.16) and 24.36%(SD=43.20) of the Microsoft SwiftKey without TalkBack and with TalkBack option, respectively. The ANOVA shows a significant effect of the efficiency of the proposed keyboard with single-factor ANOVA ($F_{2,231} = 67.0$, $P = 0$). It clearly demonstrates that the proposed keyboard performs desirable results compared to Microsoft SwiftKey without TalkBack. The efficiency of the Microsoft SwiftKey keyboard is much lower due to the smaller size of the keys. So, they are not suitable for blind and vision-impaired users. TalkBack option provides extra support, but it takes a long time to understand the voice before tapping the final key, or tapping the wrong key. The efficiency of the proposed system in terms of tasks is 61.54%(SD=49.61), 100.0%(SD=0), and 76.92%(SD=42.97) in terms of tasks performed, as shown in Figure 12. The efficiency of the proposed keyboard will be improved with usage and by keeping it in practice. Efficiency improves in the subsequent task as users have learned various key combinations in previous tasks. The efficiency of task 2 is more than that of task 3 because it contains short phrases and words without punctuation marks. Punctuation marks and longer sentences increase the difficulty level for blind and vision-impaired users. The ANOVA shows a significant effect on efficiency of the proposed keyboard with different types of content, with single-factor ANOVA ($F_{2,75} = 6.79$, $P = 0.0019$). The efficiency of the proposed keyboard is higher for simple content than for complex content containing punctuation marks and long sentences.

The efficiency of the proposed system for the age group is 82.05%(SD=38.88) and 76.92%(SD=42.68), as shown in Figure 13. It shows users in the age group 21-25 are more active and have more skills in using mobile devices compared to users belonging to the age group 26-30.

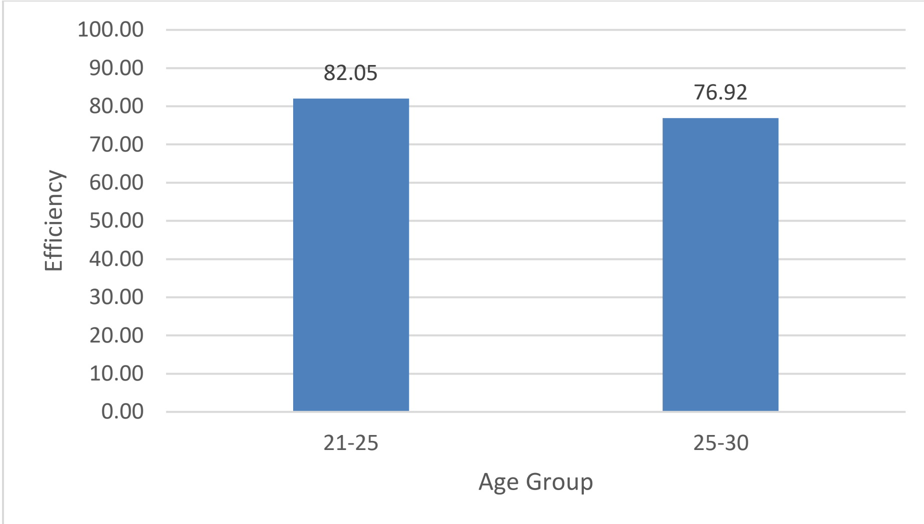

Figure 3: Efficiency of the proposed system in term of age group

So, they quickly adapt to the proposed system and give better results. ANOVA does not show any significant effect of efficiency of using the proposed keyboard by using age group users with single factor ANOVA as ($F_{1,76} = 0.31$, $P = 0.58$), so the same efficiency can be achieved by each age group user with proper training.

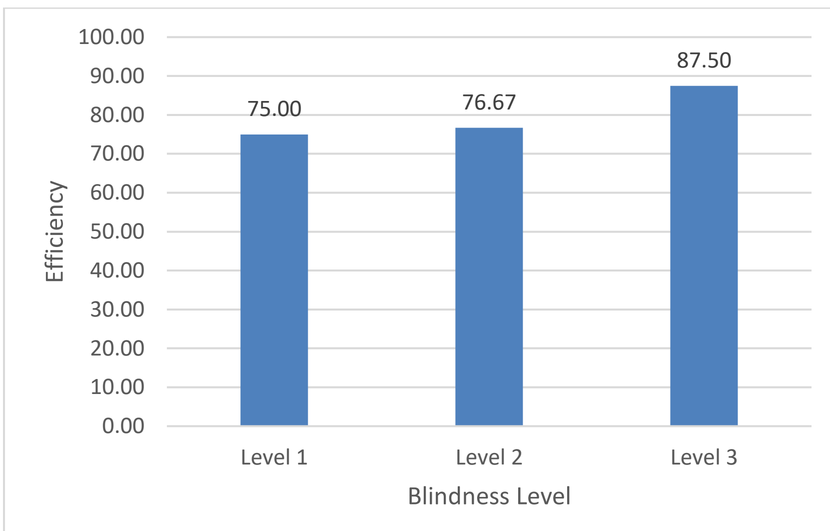

Figure 4: Efficiency of the Proposed System in Term of Blindness Level

The results of efficiency for various levels of blindness are shown in Figure 14, 75.00%(SD=44.23), 76.67%(SD=43.02), and 87.50%(SD=33.78) for level 1, level 2, and level 3, respectively. It shows the difficulty level and complexity for totally blinds with the proposed system, and they need to align themselves to the keys every time before typing new word and they have to listen first and then type, whereas for Level 3 blindness, they can easily align their thumbs to the keys and type as they listen with comfort as compare to other level of blindness. The ANOVA does not show any significant impact of blindness on the efficiency of the proposed keyboard with single-factor ANOVA ($F_{2,75} = 0.68$, $P = 0.51$).

5.3. Satisfaction

The ASR questionnaire shows the results of usability of the proposed system in terms of age group and blindness levels after each task performed by the user. The overall average ASR score of 4.61 shows that users are willing to adopt the proposed system and show much satisfaction with the proposed keyboard. ASR score of 4.86 for the age group of 21-25 and 4.36 for the age group of 25-30. It shows younger users are willing to accept the proposed keyboard and are much more satisfied compared to the age group of 25-30. As they are facing some complexity in learning the required skills before adopting the new proposed keyboard.

The average ASR score in terms of blindness level is 4.02, 4.67, and 5.13, respectively. Users with blindness level 3 are more satisfied than users with blindness level 1. It is due to the level of difficulty in aligning their thumbs on the keys for blindness level 1 and level 2 users. On the other hand, users with blindness level 3 perform the tasks easily, as their ability to adjust to the proposed keyboard. They operate the proposed keyboard easily and without much difficulty compared to totally blind users.

A Wilcoxon Signed-Rank test [24] is performed to identify the impact of ASR score after conducting each task. It shows a much more significant z-value after the second task (z = 4.22) and after the third task (z = 4.46) by comparing with 1.96 as the standard z-value. It shows that the level of user satisfaction increased with improved skills in the next tasks. With more training and learning, users could perform better in the next task, as shown in Table 5. This test is used to compare multiple groups of inputs to determine if the values of one group are significantly different than the values of another group.

Table 2: Results of task performed with multiple rounds (only three respondent data is shown)

Sr.	User	Tasks	Round	Time	Words	Mistakes	WPM	Error Rate	Effectiveness	Efficiency
1	1	Task 1	1	120	15	4	5.5	0.27	73.33	0
2			2	125	15	5	4.8	0.33	66.67	0
3			3	138	15	3	5.22	0.2	80	0
4	2	Task 2	1	170	30	6	8.47	0.2	80	1
5			2	168	30	4	9.29	0.13	86.67	1
6			3	172	30	5	8.72	0.17	83.33	1
7	3	Task 3	1	200	30	6	7.2	0.2	80	1
8			2	195	30	7	7.08	0.23	76.67	1
9			3	210	30	4	7.43	0.13	86.67	1

5.4. Comparison with Available Options

Due to high familiarity with the QWERTY layout, users can achieve better WPM and show satisfaction in using the proposed keyboard design. The overall satisfaction can be increased with extensive training. Most of the users performed better in the next round of the same task, as shown in Table 5, which contains information related to the first three users only. After practice and training, they can type without focusing on the screen and show a lower error rate compared to a full-sized QWERTY keyboard. It is observed that there is a slight decrease in typing speed due to extensive use of the control key and various key combinations. Users with impaired vision report greater satisfaction than those who are totally blind with the proposed keyboard, but they prefer less thumb movement compared to other available virtual keyboards. Another major issue with the blind users is synchronization of thumbs with the keys in the proposed keyboard. Totally blind users have to face it a lot of time during the testing, whereas partially blinds feel comfort in aligning their thumbs with the proposed keyboard keys.

The average entry speed of the proposed keyboard is 7.41 wpm as compared to 5.5 wpm of Senorita in [2], as shown in Figure 15. Further, the proposed keyboard shows an average speed of 7.16 wpm for blind and low vision users, including level 1 blindness, as compared to 2.52 wpm in Senorita, and 7.30 wpm for vision-impaired users having level 2 and 3 blindness, as compared to Senorita, which shows 3.69 wpm. Our proposed keyboard is slightly behind ThumbStroke [10], which shows 7.83 wpm with small-sized phones, which is a single-hand device to type with one thumb only, but it is not suitable for long jobs, as picking up the mobile in one hand is not feasible for a long period. HiFinger [13] shows an average entry speed of 9.82 wpm as compared to 7.41 wpm of our proposed keyboard. It is due to HiFinger, a wearable hand device, where the alignment of keys is perfect as compared to our proposed system. It is a simple mobile application that aligns the thumbs on the screen by picking up the device in the hand. If the hand grip becomes loose, then there is a chance of unalignment of thumbs on the screen that generates typing errors. The ANOVA does not show any significant impact of blindness on entry speed using the proposed keyboard with single-factor ANOVA ($F_{2,76}$ = 1.84, P = 0.165), which indicates the use of the proposed keyboard would be a perfect solution for all users, irrespective of blindness level.

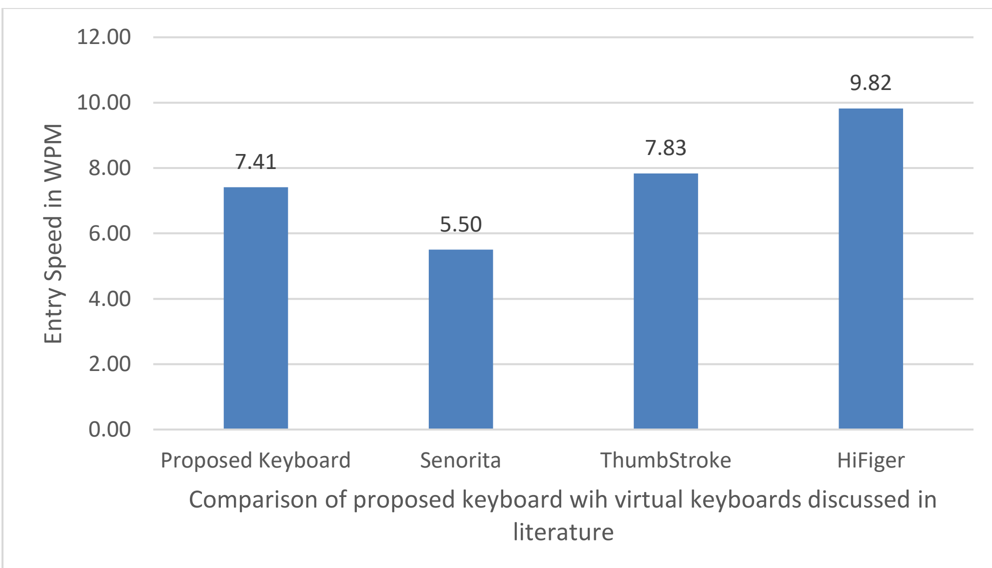

Figure 15: Comparison of proposed keyboard with other virtual keyboards discussed in literature

The proposed keyboard shows 25.99% error rate for blindness level 1 and 21.76% error rate for blindness levels 2 and 3. Whereas Senorita [2] shows 4.2% error rate for blind users and 13.24% for vision-impaired users. In HiFinger [13], the rate is 7.48% as compare to the average 24.39% error rate of the proposed keyboard. ThumbStroke [10] shows 1.43% error rate, as the usage of a single thumb reduces the alignment issue for a short period. The huge difference in error rate is due to a lack of proper training and skills, and users have to face the problem of thumb alignment on the keyboard screen. Many users are unable to realign their thumbs themselves due to blindness or low vision. The ANOVA also shows a significant impact of blindness on error rate with single-factor ANOVA ($F_{2,75}$ = 6.82, P = 0.0019), using the proposed keyboard. It indicates that the totally blind users have a problem aligning the thumb on the keyboard screen as compared to vision-impaired users, who can easily adjust their thumbs to press the right keyboard keys, but proper training and learning about using keyboard keys to type different characters are essential to reduce the error rate.

The overall results are satisfactory at the initial phase of the proposed keyboard development, so it can be improved with more practice and more training. On the basis of these, training and practice are the crucial and important requirements. After proper training, the effectiveness, efficiency, and satisfaction in terms of usability of the proposed keyboard will be enhanced for blind and vision-impaired users.

6. Conclusions

One of the main elements of information technology is not its availability, but rather its accessibility to different users, especially disabled or handicapped. Most of the information and communication technologies have accessibility issues for blind and vision-impaired users. Interfaces to these input devices cannot adjust for these special users, so there is a need for time to design interfaces that adjust to the limitations of special user contexts. In this paper, a proposed virtual keyboard is designed to facilitate the entry of data for the blind and vision-impaired with sight-free typing. The results of the proposed keyboard show 80.23% effectiveness for vision-impaired users and 72.08% for totally blind users. Vision-impaired users can set their thumbs easily without third-party support, whereas the totally blind need third-party support to adjust their thumbs on the keyboard interface. For totally blind users and vision-impaired users, the efficiency of the proposed keyboard is 75.0% and 87.50%, respectively. The difference is also due to the difficulty level for totally blinds to set their thumb for the right key after listening to the words, so it takes more time to type the same text for totally blind users compared to vision-impaired users. The test scores show the improvement in the next rounds of tasks in most cases. It indicates that with practice and training, overall effectiveness will be improved positively. If they adopt this application for a long time, its effectiveness and efficiency will surpass other virtual keyboards. The overall effectiveness of the proposed keyboard is 75.31% as compare 25.17% and 55.76% for Microsoft SwiftKey without TalkBack and with TalkBack, respectively. The proposed keyboard shows a satisfactory result of efficiency of 79.49% as compared to 11.54% and 24.36% of Microsoft SwiftKey without TalkBack and with TalkBack option, respectively.

The overall satisfaction ASR score is 4.61 out of 7.0, which shows the proposed keyboard provides satisfactory outcomes. It also shows a better experience for users in adopting the proposed keyboard. The overall performance of the proposed keyboard in terms of WPM and error rate is compared with Senorita, HiFinger, and ThumbStroke. The proposed keyboard outperforms Senorita by achieving 7.41 wpm typing speed as compare to 5.5 wpm. It is just behind 7.83 wpm and 9.82 wpm of ThumbStroke and HiFinger, respectively. In terms of error rate, our proposed keyboard is struggling behind Senorita, HiFinger, and ThumbStroke, but this is due to a lack of sufficient training and learning of the key combinations. Instead of these, the proposed keyboard would be an effective typing solution with proper training and experience of blind and vision-impaired users.

7. Limitations

The proposed keyboard provides a lot of benefits compared to existing virtual keyboards. Instead of that, there are a few limitations in its initial development. The effectiveness of the system can be achieved with two thumbs of both hands, but it is difficult to operate for physically disabled users with a single hand. Built-in systems such as voiceover in the Android operating system are available, but they are not easy to use by blind and vision-impaired users. These take much time to enter data due to the extra operations required. The proposed keyboard design has only been tested with a vertical mobile screen. There are some issues in horizontal mode that still need more clarification on thumb movements to access the middle key. It is also difficult to align the thumbs in the right position for blind users on different-sized screens. It can be controlled by using sound beeps, but the movement of thumbs on various keys is still an issue. For more clarity for blinds, it can be used with the TalkBack option, but still, research is required to overcome the complexities and extra time consumed there. There is no proper mechanism to adjust the movement along the keys; users just set an arbitrary distance to jump over the next key, so some mechanism is required to guess the position of the next key. Another important limitation is the keyboard status while typing; the user must remember the last action performed if there is a pause during typing, otherwise, they need to reset the keyboard to its initial position.

7.1. Future Work

The proposed research can be used to enhance the design for physically disabled users who have only one hand or fewer fingers on one hand. This compact keyboard can be designed in a vertical layout that would be easy to enter text with one hand, or with a single thumb only. The proposed system can be analysed to test the adoptability by the normal users to gain a lot of expected benefits. The alignment of thumbs in the right position is one of the main issues in the proposed system that can be further analysed to position automatically by sensing the thumbs on the screen.

Data Fabrication/Falsification Statement

The author(s) declare that no data have been fabricated, falsified, or manipulated in this study.

Participant Consent

The authors confirm that Informed consent was obtained from all participants, and confidentiality was duly maintained.

8. References

[1] M.-I. B. Lin, R.-H. Hong and Y.-P. Huang, "Influence of virtual keyboard design and usage posture on typing performance and muscle activity during tablet interaction," *Ergonomics,* vol. 63, no. 10, pp. 1312-1328, 2020.

[2] G. rakhmetulla and A. Arif, "Senorita: A Chorded Keyboard for Sighted, Low Vision, and Blind Mobile Users," in *CHI 2020*, Honolulu, HI, USA, 2020.

[3] E. Matias, I. S. MacKenzie and W. Buxion, "One-Handed Touch Typing on a the QWERTY Keyboard," *Human Computer Interaction,* vol. 11, pp. 1-27, 1996.

[4] T. Starner, "Keyboards redux: Fast mobile text entry," in *IEEE Pervasive Computing*.

[5] R. Puka, P. Łebkowski and J. Duda, "One Row Keyboard: The Concept of Designing a Common Layout for Physical and Virtual Keyboards," *Electronic,* vol. 10, no. 6, p. 663, 2021.

[6] R. Mathur, A. Sheth, P. Vyas and D. Bolchini, “Typing Slowly but Screen-Free: Exploring Navigation over Entirely Auditory Keyboards,” in *In Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, Pittsburgh, PA, USA, 2019.

[7] R. Sawant, Prabhav, P. Shrivastava, P. Shahane and H. R, “Text to Braille Conversion System,” in *International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) IEEE*, 2021.

[8] E. Matias, I. S. Mackenzie and W. Buxton, “Half-QWERTY: A One-handed Keyboard Facilitating Skill Transfer from the QWERTY,” in *In Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, 1993.

[9] J. Son, S. Ahn, S. Kim and G. Lee, “Improving two-thumb touchpad typing in virtual reality,” in *In Extended abstracts of the 2019 CHI conference on human factors in computing systems*.

[10] J. Lai, D. Zhang, S. Wang, I. D. Y. Kilic and L. Zhou, “ThumbStroke: A Virtual Keyboard in Support of Sight-Free and One-Handed Text Entry on Touchscreen Mobile Devices,” *ACM Transactions on Management Information Systems,* vol. 10, no. 3, pp. 1-19, 2019.

[11] J. Protim, A. T. M. Abdullah, A. I. Pritom and M. Z. Chowdhury, “BADHON: A High Performing Keyboard Layout for Physically Impaired People,” in *In 2019 23rd International Computer Science and Engineering Conference (ICSEC) IEEE*, 2019.

[12] H. Kim, S. Yi and S.-Y. Yoon, “Exploring touch feedback display of virtual keyboards for reduced eye movement,” *Displays,* vol. 56, pp. 38-48, 2019.

[13] H. Jiang, D. Weng, Z. Zhang and F. Chen, “HiFinger: One-Handed Text Entry Technique for Virtual Environments Based on Touches between Fingers,” *Sensors,* vol. 19, no. 14, p. 3063, 2019.

[14] Microsoft, “Microsoft SwiftKey,” Microsoft, [Online]. Available: https://www.microsoft.com/en-us/swiftkey. [Accessed 2 4 2026].

[15] Google.com, “Get started on Android with TalkBack,” Google, [Online]. Available: https://support.google.com/accessibility/android/answer/6283677?hl=en. [Accessed 3 4 2026].

[16] J. Lazar, J. H. Feng and H. Hochheiser, Research methods in human-computer interaction, Morgan Kaufmann, Wiley, 2017.

[17] T. Watanabe, H. Kaga, M. Kobayashi and K. Minatani, “Touchscreen Text Entry Methods Used by Blind and Low Vision Users,” *ITE Transactions on Media Technology and Applications (MTA),* vol. 7, no. 3, pp. 134-141, 2019.

[18] M. W. Iqbal, N. Ahmad, S. K. Shahzad, M. R. Naqvi and I. Feroz, “Usability Aspects of Adaptive Mobile Interfaces for Colour-Blind and Vision Deficent Users,” *IJCSNS International Journal of Computer Science and Network Security,* vol. 18, no. 10, pp. 179-189, 2018.

[19] B. Nigel, “International standards for HCI and usability,” *International journal of human-computer studies,* vol. 55, no. 4, pp. 533-552, 2001.

[20] L. J. R., “IBM Computer Usability Satisfaction Questionnaires: Psychometric evaluation and Instructions for Use,” *International journal of Human-Computer Interaction,* vol. 7, no. 1, pp. 57-78, 1995.

[21] R. W. Soukoreff and I. S. MacKenzie, “Recent Developments in Text-Entry Error Rate Measurement,” in *In CHI'04 extended abstracts on Human factors in computing systems*, 2004.

[22] D. LOTTRIDGE, C. YOON, D. BURTON, C. WANG and J. KAYE, “Ally: Understanding Text Messaging to Build a Better Onscreen Keyboard for Blind People,” *ACM Transactions on Accessible Computing,* vol. 15, no. 4, p. 22, 2022.

[23] R. N. Cardinal and M. R. Aitken, ANOVA for the behavioural sciences researcher, New York: Psychology Press, 2006.

[24] S. Kotz, Encyclopedia of Statistical Sciences, vol. 3, New York: Wiley, 1982.

[25] g. Yfantidis and G. Evreinov, “Blind Text Entry for Mobile Devices,” in *In proc. of AAATE*, 2005.