Computer Systems Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.045981

ARTICLE

Virtual Keyboard: A Real-Time Hand Gesture Recognition-Based Character Input System Using LSTM and Mediapipe Holistic

Bijon Mallik[1], Md Abdur Rahim[1], Abu Saleh Musa Miah[2], Keun Soo Yun[3,*] and Jungpil Shin[2]

[1]Department of Computer Science and Engineering, Pabna University of Science and Technology, Pabna, 6600, Bangladesh

[2]School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu, Fukushima, 965-8580, Japan

[3]School of Computer and Information Technology, Ulsan College, Dong-gu, Ulsan, 44610, South Korea

*Corresponding Author: Keun Soo Yun. Email: ksyun@uc.ac.kr

Received: 13 September 2023 Accepted: 12 December 2023 Published: 19 March 2024

ABSTRACT

In the digital age, non-touch communication technologies are reshaping human-device interactions and raising security concerns. A major challenge in current technology is the misinterpretation of gestures by sensors and cameras, often caused by environmental factors. This issue has spurred the need for advanced data processing methods to achieve more accurate gesture recognition and predictions. Our study presents a novel virtual keyboard allowing character input via distinct hand gestures, focusing on two key aspects: hand gesture recognition and character input mechanisms. We developed a novel model with LSTM and fully connected layers for enhanced sequential data processing and hand gesture recognition. We also integrated CNN, max-pooling, and dropout layers for improved spatial feature extraction. This model architecture processes both temporal and spatial aspects of hand gestures, using LSTM to extract complex patterns from frame sequences for a comprehensive understanding of input data. Our unique dataset, essential for training the model, includes 1,662 landmarks from dynamic hand gestures, 33 postures, and 468 face landmarks, all captured in real-time using advanced pose estimation. The model demonstrated high accuracy, achieving 98.52% in hand gesture recognition and over 97% in character input across different scenarios. Its excellent performance in real-time testing underlines its practicality and effectiveness, marking a significant advancement in enhancing human-device interactions in the digital age.

KEYWORDS

Hand gesture recognition; M.P. holistic; open CV; virtual keyboard; LSTM; human-computer interaction

1 Introduction

Gesture recognition enhances the natural and intuitive interaction between humans and computers, contributing to smoother and more immersive virtual environments [1]. This technology holds the potential to significantly amplify user experience across various domains, such as virtual reality (V.R.), augmented reality (A.R.), touchless interfaces, air writing, and gaming. In recent times, interactions between humans and computers have evolved to become increasingly sophisticated, employing various communication means, like physical movements of hands, fingers, arms, and other body parts, to facilitate interaction [2–4]. Hand gesture recognition, a crucial method that enables users to control

devices without physical contact or traditional input devices like keyboards and mice, has notably made significant contributions to A.R., non-touch systems, sign language recognition, and beyond.

Researchers have investigated numerous methods to recognize and interpret gestures, including sensor-based, computer vision-based, and hybrid approaches that combine both data types. Computer vision-based techniques utilize video or image analysis to identify distinctive gesture characteristics, such as hand shape, trajectory, and movement [5]. In contrast, sensor-based methods capture gesture data directly using sensors like flex sensors, accelerometers, or gyroscopes, often utilizing data gloves or motion sensors. Although sensor-based approaches offer precise data capture, they can be limited by environmental interference, restricted range, and the bulky nature of the equipment. Furthermore, this method mandates the use of a data-capture hand glove with embedded sensors, making the device burdensome to carry [6].

Existing literature unveils common challenges and complications in gesture recognition approach -es, such as high sensitivity to background noise and lighting conditions, which can compromise their accuracy. Additionally, the limited range and costly devices present further challenges for sensor-based methods, causing difficulties in precisely identifying complex and subtle movements. To overcome the problem, Rahim et al. proposed a CNN-based feature extraction and SVM machine learning algorithm to improve the performance accuracy [7]. Also, their system achieved good performance accuracy but still faced problems in terms of computational complexity and efficiency because of the redundant background, light illumination, and partial occlusion. To overcome the problem, Shin et al. utilized the MediaPipe system to isolate 21 crucial hand points from an American Sign Language dataset, subsequently employing an SVM for recognition after deriving the distance and angular features [8]. Sole reliance on hand skeleton data occasionally falls short of accurately conveying a sign's precise meaning, given the absence of emotional and physical expression.

Consequently, recent scholarly efforts have pondered whether incorporating a complete body skeleton might enhance the efficacy of SLR systems [9]. However, most of the research work for gesture recognition has been developed for the general hand gesture recognition or sign language recognition (SLR) application. However, very few research works have been done to recognize hand gestures for operating a virtual keyboard through pointing and touching with the fingertips because of the unavailability of the dataset for this. In addition, for the virtual Keyboard, it is crucial to collect exact gesture information from the participant to avoid confusion about the exact command. Moreover, most of the previous hand gesture recognition tasks based on the skeleton information used only hand skeleton datasets, and few works have been done that considered whole body information as gesture information, but their performance accuracy is not satisfactory. To overcome the issue, we proposed a hand gesture recognition for operating the virtual Keyboard by taking leverage of image processing and deep learning methods.

The main contributions of this paper are highlighted as follows:

1. **Novel Hand Pose Dataset Creation:** A new hand pose dataset was created specifically for virtual keyboards, addressing the limitations of existing datasets. It features a broad array of hand gestures, static and dynamic, covering diverse hand movements. The dataset uses MediaPipe to capture detailed joint skeletons, extracting 1662 landmarks (468 faces, body, and 21 per hand), improving precision and enhancing hand gesture recognition research.
2. **Innovative Use of Deep Learning:** In the study, we introduce a new model combining LSTM and fully connected layers for a better understanding of sequential data. It also includes CNN layers, max-pooling, and dropout layers, allowing it to process both temporal and spatial information, leading to improved performance in sequential data analysis. This approach

allows for a more comprehensive understanding of input data and superior performance in sequential data analysis.

3. **Comprehensive Evaluation:** The model underwent extensive evaluation in various scenarios, including real-time applications, demonstrating robustness and effectiveness. The code and dataset are available at https://github.com/musaru/Virtual_Keyboard. Promoting transparency and facilitating replication of experiments, thus encouraging broader adoption of the model.

The remainder of this paper is organized as follows: Section 2 summarizes the existing research work and related problems. Section 3 describes the descriptions of the datasets, and Section 4 describes the architecture of the proposed system. Section 5 details the evaluation performed, including a comparison with a state-of-the-art approach. In Section 6, our conclusions and directions for future work are discussed.

2 Literature Review

Our investigations examined the vision-based strategy, including recognizing and categorizing hand motions. Hand recognition uses neural networks and algorithms to determine hand motion and then detect recurring behavioral activity patterns [10–12]. A two-step approach for classifying hand gestures made by various subjects under significantly variable illumination conditions. Using trained 3D Convolutional neural networks, each hand gesture is first classified [13–15]. Then, utilizing the honed 3D Convolutional neural network, automated learning of spatiotemporal information for lengthy short-term memory takes place [16,17]. Another study of particular gestures that would initiate hand detection, tracking, and segmentation utilizing motion and color signals. Finally, scale-space feature detection was incorporated into gesture identification to overcome the aspect ratio limitation experienced in the majority of learning-based hand gesture systems [18,19]. Another system uses 24 gestures, including 13 static and 11 dynamic gestures that are relevant to the environment, to address problems with gesture detection. The suggested deep learning architecture was used to collect, prepare, and train the RGB and depth picture data set. A three-dimensional convolutional neural network (3DCNN) [4,20–22] and a long short-term memory (LSTM) model were integrated in order to extract the spatiotemporal information [23].

We have organized new approach based on these features stated in this paper according to the environment. Another study continuously detected and recognized hand gestures using a frequency-modulated continuous wave (FMCW) radar. They initially estimated the range and Doppler parameters of the hand gesture raw data using the 2-Dimensional Fast Fourier Transform (2D-FFT), and then they constructed the range-time map (RTM) and Doppler-time map (DTM). Second, a hand gesture detection technique using a decision threshold was suggested to partition the continuous hand movements. Thirdly, following the clustering of each hand gesture's spectrogram using the k-means method, the Fusion Dynamic Time Warping (FDTW) technique was developed to detect the hand gestures [24]. We have acknowledged the heaviness of this approach, trading off with accuracy. We then chose a light approach with adjustable accuracy in a sense. A study investigated the application of data mining techniques for hand motion recognition. They used data mining to first identify bandwidth hand gestures [25]. They were also excellent for data mining because of their high temporal resolution.

A close-pulse, short-range broadcast signal was usually required [26]. One paper stated that for continuous hand gesture identification, electromyography (EMG), strain/pressure sensing, multi-source data sensing, data gathering and processing, and wireless communication, a soft wrist-worn sensor system (SWSS) was created [27–29]. Our work has overcome the drawbacks of these works

efficiently. Another proposal of a neural network architecture based on R-FCN is suggested; to be more precise, they initially created an adaptive template selection technique for a variety of hand-raising gesture-detecting movements. Second, they created a feature pyramid to concurrently collect the detail and highly semantic features for better detection of small-size hands [30]. We took the idea of feature extraction from this paper, and we have tried to implement it like them, but after rigorous testing, we changed the technique and approached our new one. A hand gesture recognition (HGR) algorithm was put forth, which is capable of handling time-dependent input from an inertial measurement unit (IMU) sensor and supporting real-time learning for a variety of human-machine interface (HMI) applications [6,31].

We have already discussed sensors and how they work in this field. As stated earlier, we are not into sensors, but this paper gave us basic ideas to compare with our approach. A study proposed low-cost consumer radar integrated circuits with recent improvements in machine learning have created several brand-new opportunities for smart sensing. They trained a deep convolutional neural network (DCNN) to classify the Doppler signatures of 14 different hand motions using a small radar sensor [27]. They made use of two receiving antennas from a continuous-wave Doppler radar that could generate the beat signals' in-phase and quadrature components. We have experimented with the result, and compared to the single-channel Doppler approach, it does not perform so well. Our approach covers the drawbacks it has and provides a simple and easiest solution for some specific tasks. Another study offered a system for 3D skeletal tracking combined with deep learning 133 for real-time hand motion detection and recognition [32]. In a study, a virtual keyboard enables users to text to any gadget from any plane. The customized virtual Keyboard is printed on simple paper so that it may be affixed to a wall or placed on any oblique 136 planes. Then, based on the location of the fingertip and hand skin tone, the device camera is employed for key recognition [33]. Another study regarding the virtual Keyboard was proposed as the computer's camera records the user's hand movements in various gestures, and the mouse or cursor moves in accordance with those movements, including the left and right clicks utilizing various gestures. An algorithm is created to map the mouse and keyboard functions with the convex hull flaws that the system first creates [34].

In our virtual Keyboard, we have all three options, e.g., clicking like a normal keyboard and non-touch approach like tapping and gestures-based key pressed. Also, the approaches stated above in their paper lack efficiency because of the blurry background and this noisy environment, dropping their expected accuracy. As we have worked with landmarks that do not vary from person to person, it reduces the risk of dropping the accuracy. A non-touch character writing system suggested system is composed of two primary components: one is a virtual keyboard for gestural flick input and hand gesture recognition. For feature extraction of the gestures, the system employs a deep learning technique utilizing convolutional neural networks (CNN). Color segmentation using specific HSV and threshold masking is used to detect the hand, and then a support vector machine is used to classify the gesture [35]. This paper was inspired by the basic technique of gestural operation, and we developed the concept to expand the work area in the near future.

3 Dataset Description

Our dataset accommodates both static and dynamic gestures. In our methodology, we distinguish between static gestures, which correspond to individual frames similar to traditional image-based training, and dynamic gestures, which are represented as sequences of frames within videos. We assign labels to selected frames within these videos and train our model to recognize dynamic gestures. For hand gesture recognition, we capture data in the form of landmark coordinates using OpenCV and a 1080P YOCO full-HD web camera. To ensure dataset diversity, we collect data from multiple

participants and under various lighting conditions. We meticulously label the data by assigning distinct class names, such as “Hello” or “Bye,” along with corresponding descriptions. This labelling process associates hand movements with their intended meanings, laying the foundation for training and testing hand gesture recognition models. Our dataset focuses on ten gestures carefully chosen to operate our virtual keyboard effectively.

These gestures include “hello,” “ok,” “bye,” “victory,” “show,” “space,” “thanks,” “clear,” “backspace,” and “tab.” Fig. 1 visually illustrates our data collection procedure, which entails using a web camera. For dynamic hand gestures, such as those requiring a sequence of frames to convey their meaning effectively, we capture 30 frames from each video. In contrast, static hand gestures can be adequately represented by a single frame, and we collect 30 such frames for each static gesture. From each frame, we meticulously extract 1,662 pose landmarks, including those from the hand, face, and body. These landmarks provide rich and detailed information about hand gestures.

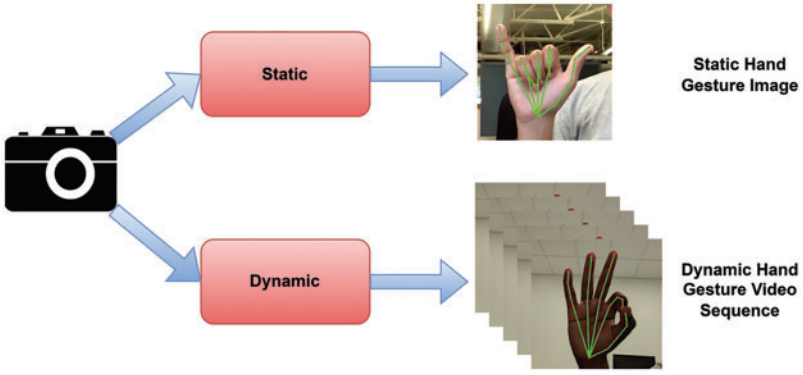

Figure 1: Data collection procedure

Fig. 2 demonstrates the dataset sample of our newly created hand gesture dataset for the virtual keyboard. Also, we considered 30 sequential frames for the dynamic gestures, but we demonstrated five sequential frames for each gesture here.

Figure 2: Sample of the dynamic dataset

4 Methodology

Fig. 3 illustrates the architecture of our proposed method. We used an RGB image as the input and identified pose landmarks through a precise position estimation approach. Subsequently, we leverage a deep learning-based method for feature extraction and classification. In our study, we introduce a novel integration of several LSTM layers along with a set of fully connected layers, as depicted in Fig. 4a. This integration empowers our model with the ability to capture long-range dependencies and temporal patterns, making it particularly effective in tasks involving sequential data. Furthermore, to enhance the model's ability to extract spatial features, we incorporate multiple CNN, max-pooling, and dropout layers in conjunction with the LSTM mechanism, as depicted in Fig. 4b. This combined architecture allows us to simultaneously capture both temporal and spatial information, providing a comprehensive understanding of the input data. The utilization of a long-short-term memory (LSTM) approach in both model configurations enables the extraction of intricate patterns and relationships from sequences of frames, leading to superior performance in tasks requiring sequential data analysis. Finally, it is worth noting that our approach culminates in the creation of a pre-trained model, which we rigorously test in real-time across various real-world datasets. This approach showcases the versatility and robustness of our model, making it a valuable tool in a wide range of applications.

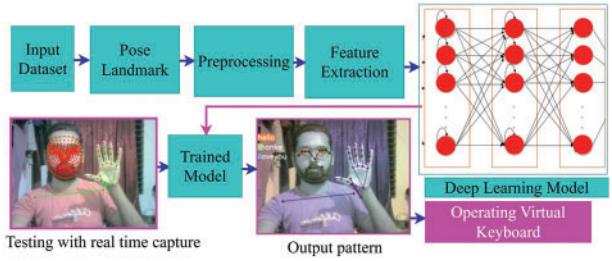

Figure 3: The architecture of our proposed approach

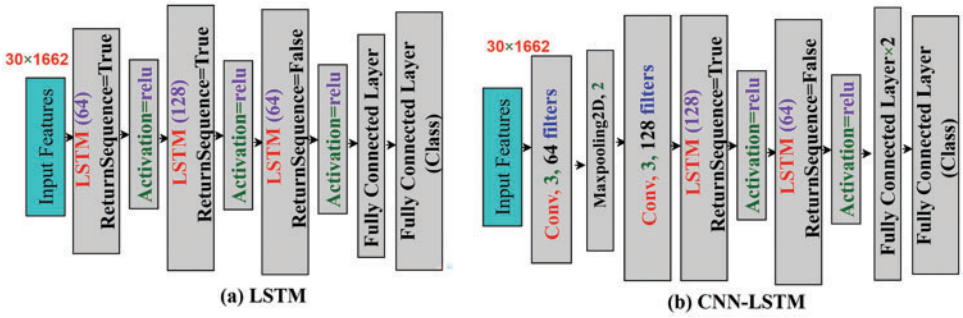

Figure 4: Proposed deep learning-based model (a) LSTM (b) CNN-LSTM

4.1 Data Preprocessing

The MediaPipe Holistic API is used to acquire the landmark points of the frame (image), which are then saved to a folder as a NumPy array. We compiled 30 videos for each gesture from every action on the Virtual Keyboard. The landmark points can be extracted using one of two methods. One method involves reading captured gestures for a virtual keyboard and then extracting the landmarks from them. The alternative technique entails swiftly extracting the landmark locations from a real-time video recording. Due to the limited control we have over the accuracy of recorded input data—especially

when it originates from various sources and uses a variety of frame settings. However, the first approach is simpler to use but less effective in controlling the accuracy of landmark extraction. The second method is more accurate, although it necessitates longer collection times.

It was used to precisely determine the locations of landmarks in real time, verifying that the neural network had received enough training. Additionally, the virtual Keyboard was exclusively used for specific operations. After gathering landmarks from the previous stage, compile them into a NumPy array. Each of the 21 points on each hand within the hand landmarks has three-dimensional coordinates (x, y, and z). The total number of landmarks on both hands is determined by multiplying 21 by three by 2, resulting in 126 landmarks ($21 \times 3 \times 2 = 126$) being considered. The face landmarks consist of 468 points that share the exact three-dimensional coordinates as those in the hand landmarks. The total facial landmarks are 1404. The body landmarks consist of 33 points in four dimensions (x, y, z, and visibility). Therefore, the total number of landmarks on the body can be calculated as 33 times four, or 132, landmarks. As a result, each frame's landmarks have a total of 1662 points. We used seven hand gestures for the experimental evaluation. Each gesture is composed of 30 videos, each with 30 frames. The total number of landmarks for each gesture (word) is $1662 \times 30 \times 30 \times 3 = 4{,}487{,}400$.

4.2 Implementation of LSTM

The study introduces a novel model configuration comprising multiple LSTM layers [36] and a set of fully connected layers, as illustrated in Fig. 4a. This integration empowers our model to effectively capture long-range dependencies and temporal patterns, especially in sequential data analysis tasks. To further enhance our model's ability to extract spatial features, we incorporate multiple CNN layers alongside max-pooling and dropout layers, complementing the LSTM mechanism, as shown in Fig. 4b. This cohesive architecture enables the concurrent extraction of both temporal and spatial information, providing a holistic grasp of the input data. Employing the long-short-term memory (LSTM) approach in both model configurations enables us to extract intricate patterns and relationships from sequences of frames, resulting in superior performance when handling tasks that necessitate sequential data analysis.

Fig. 5 shows the detailed internal architecture of the LSTM module. We utilized a Long Short-Term Memory (LSTM) layer with 64 units and integrated multiple modules. In each model, we used the Rectified Linear Unit (ReLU) activation function. The LSTM layer returns the entire sequence of outputs for each input sequence, as the return sequence parameter is set to True. Our system took the input shape (30, 1662), where 30 represents the number of frames, and 1662 represents the body, face and hand skeleton key points. There are efficient numbers of videos, where each video contains 30 frames. A second LSTM layer with 128 units and the ReLU function is incorporated into the model as the second layer. Once again, the argument return sequence is set to True. Moving on, a third LSTM layer with 64 units and the ReLU activation function is added as the third layer of the model. As the parameter return sequence is set to False, this LSTM layer will solely provide the final result for each input sequence. Subsequently, the model includes two fully connected layers comprising 64 and 32 units, respectively, utilizing the ReLU activation function in both layers.

The final layer of our model introduces a dense layer with a number of units equal to the total count of possible actions. This layer employs the Softmax activation function, commonly used to establish a probabilistic model across potential classes in classification problems. We employed this model to recognize three actions, as the remaining processes will follow similar procedures. The model consists of a total of six layers, which are outlined in Table 1.

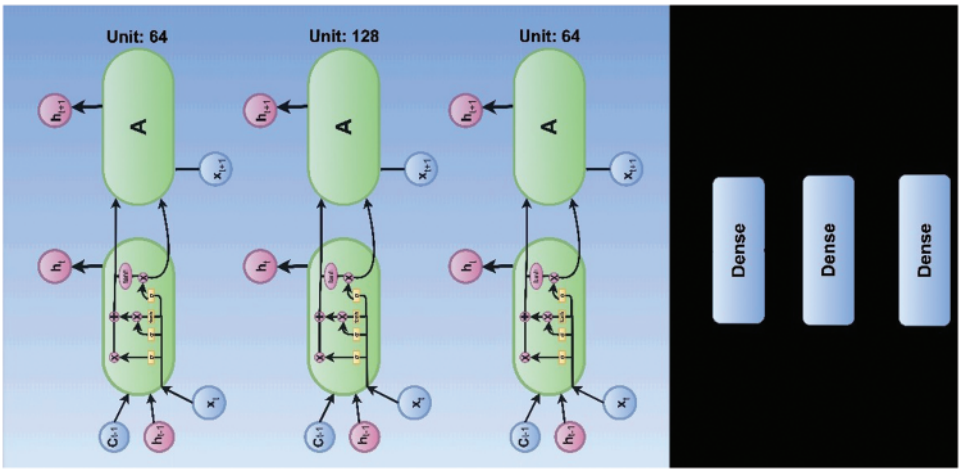

Figure 5: The internal structure of the LSTM module

Table 1: Parameters of our LSTM model

Layer	Output shape	Units	Parameters
LSTM 1	(30, 64)	64	442112
LSTM 2	(30, 128)	128	98816
LSTM 3	64	64	49408
Dense 1	64	64	4160
Dense 2	32	64	2080
Dense 3	3	3	99

4.3 Virtual Keyboard Implementation

We created a virtual keyboard using hand gesture recognition. Hand gestures are recognized through the aforementioned steps, and the predicted scores (weights) for various gestures are saved to control the Keyboard. The virtual Keyboard is displayed on the screen using the OpenCV library, and key presses are simulated using the Pynput library. To ensure practical interaction, we kept the system as simple as feasible. Based on the proposed approach, a hand-tracking module was developed using the MediaPipe solution, which includes 21 hand landmarks. These landmarks were retrieved and used to operate the virtual Keyboard. To accomplish keyboard activities, the user needs to use their thumb and index finger to click or tap. For this, our system uses a camera to capture the user's hand movements, and the video data is evaluated to identify the clicks.

The user has to click to activate the virtual Keyboard by inputting the specified individual key of the virtual Keyboard. For example, entering each character requires a single click, whereas pressing the spacebar requires a double click. The system can recognize these hand movements and assist with suitable key inputs by using machine learning technologies such as convolutional neural networks. Additionally, we determine the coordinates of important locations that were retrieved using the MediaPipe hand detection solution in order to detect tapping. Where the thumb and index finger landmarks are located by tracking the hand landmarks in the webcam images that have been collected. Our system determines which key on the virtual Keyboard is being pressed based on its positioning.

We can calculate the X, Y, and Z coordinates of these two fingers to determine their corresponding X and Y axes as well as their distance from the camera. Through testing, we discovered that a single tap gesture can be recognized if the distance between the thumb tip (top I.D. point) and the index fingertip is less than 30. The virtual Keyboard is thereby mapped to these tapped locations, which are then used to manipulate characters and perform special operations like "space," "backspace," and "clear."

Finally, we mainly focus on a non-touch method of controlling the virtual Keyboard. As previously indicated, LSTM is used to recognize movements, and each recognized gesture is associated with a unique label that is presented on the screen. The proposed virtual Keyboard uses these labels, which were assigned using Label Encoder, to replace the requirement for character input. The 'Clear' action in the tapping approach employs an empty character, whereas, in the gesture recognition method, the predicted scores of the associated gestures are used as input values. Table 2 represents the description of a graphic representation of keyboard operations.

Table 2: Descriptions of character input execution approach

Operations	String manipulation in tapping	String manipulation by gestures
SPACE	Passing a string with a space	Trained video sequence of 'hello' gesture labelled as 0
BACKSPACE	Displaying string without Last value	Trained video sequence of 'thanks' gesture labelled as 1
CLEAR	An empty string	Trained video sequence of 'right_to_left' gesture labelled as 3

5 Experimental Result

This section uses experimental training and testing principles to conduct experimental evaluations of gesture recognition and character input based on the proposed approach. Furthermore, end-users evaluate the trained model in real time.

5.1 Environmental Setup

To conduct our experiment, we employed the Python programming language within the Spyder Python environment. We leveraged various Python packages, including Cv2, NumPy, Pickle, TensorFlow, Keras, and Matplotlib, to support the implementation. Our deep learning architecture was configured with an Adam optimizer and a learning rate of 0.001. During training, we executed 100 epochs to train and evaluate the model's performance. This section is divided into two parts: identifying the model's output and determining how frequently it provides the appropriate output using the proposed virtual Keyboard. After extracting the landmarks, we divided the data into 85% for training and 15% for evaluation. As a result, once the processing processes were completed, we labelled the key information for gesture recognition and character input.

5.2 Performance of Our Model

We proposed an LSTM model combined with M.P. Holistic to recognize hand motions. In that situation, the training dataset's size and quality are determined by several factors, including the proposed LSTM model, the hyperparameter used, and the evaluation measure. The LSTM model achieves excellent accuracy in hand gesture identification by utilizing long-term dependency learning

capabilities on sequential input, whilst the M.P. holistic gives hand detection and tracking capabilities. The training and testing accuracy and loss of our proposed model are depicted in Fig. 6. Fig. 7 shows that the average accuracy of the label's performance of each gesture is 98.52%. The results show that the proposed model achieved an average recognition accuracy of 98.52%. Among them, the highest accuracy is 100% for the 'Victory' gesture, and the lowest accuracy is 97.6% for the 'Bye' gesture.

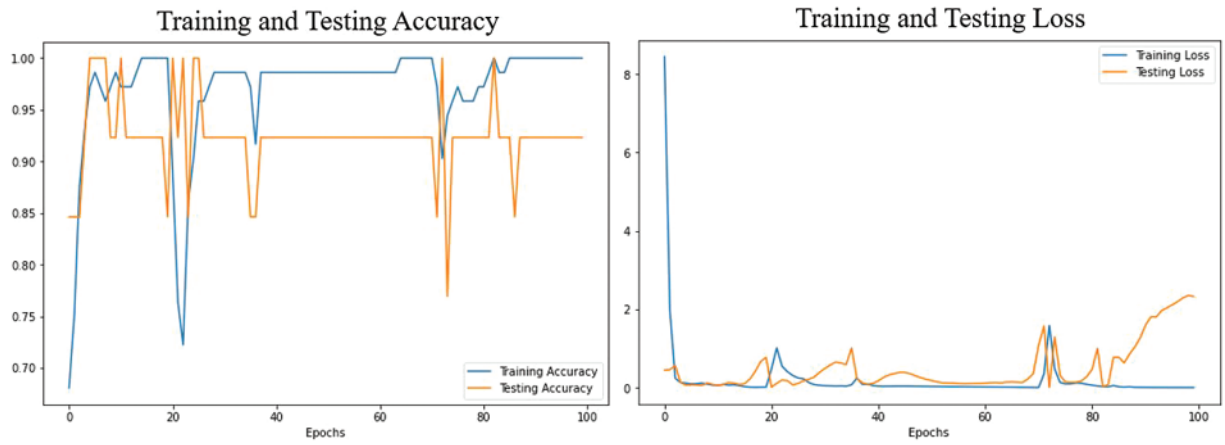

Figure 6: Accuracy and loss of the proposed model

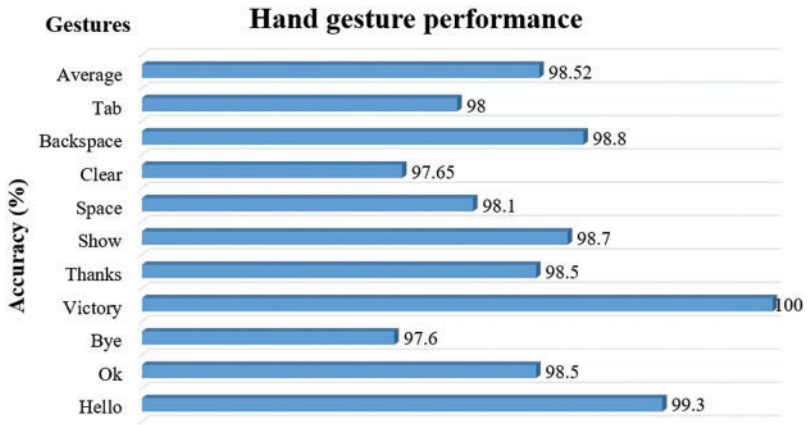

Figure 7: Label-wise performance for each gesture

5.3 Performance on the Virtual Keyboard with Trained Gestures

In this system, trained hand gestures can effectively recognize both static and dynamic (motion) gestures, and the user can rapidly input characters with these gestures on the proposed virtual Keyboard. We used string manipulation to evaluate our gestures here. The user inputs characters by viewing the virtual Keyboard on the display screen. Various motion functions such as "Clear," "Backspace," and "Tab" are performed using dynamic motion gestures. However, these results indicate a high level of accuracy. We conducted real-time user tests to evaluate the accuracy of the proposed virtual Keyboard. The results of this study are summarized in Table 3. Seven participants performed various random gestures to evaluate the proposed virtual Keyboard. Each volunteer repeated each gesture several times. In this system, character input can be accomplished in one of two ways.

Table 3: Subject independent corresponding action and detected accuracy

Subject	No. of times	Accurately performed	Mis-classification	Accuracy	Average accuracy (LSTM)	Average accuracy (CNN-LSTM)
Subject 1	Hello: 7 Ok: 6 Space: 5 Show: 4	17	1	94.44%		
Subject 2	Victory: 11 Backspace: 3 Hello: 9	18	0	100%		
Subject 3	Clear: 3 Tab: 5 Thanks: 2	19	1	94.74%		
Subject 4	Victory: 4 Bye: 17 Hello: 6	21	1	95.24%	96.55%	80.00%
Subject 5	Clear: 3 Bye: 13 Hello: 14	22	0	100%		
Subject 6	Show: 11 Thanks: 7 Bye: 8 Hello: 11	40	2	95%		
Subject 7	Victory: 13 Space: 7	31	1	96.77%		

The user can enter a single character by tapping the index and thumb fingers. Furthermore, different dynamic gestures can give diverse word input formats via it. Table 3 illustrates that each user executed distinct gestures many times. The results showed that Subject 2, Subject 5, and Subject 7 operated perfectly as inputs to all motions. The incidence of incorrect classification is 3.45%. The average recognition accuracy of all subjects was 96.55%. The example of the keyboard operation is demonstrated in Figs. 8 and 9.

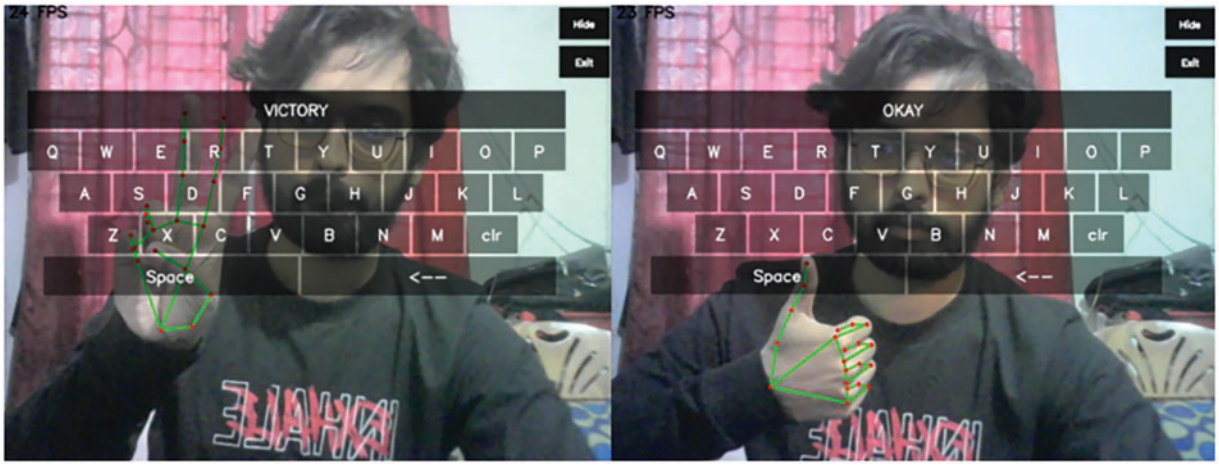

Figure 8: Operating keyboard with static gestures (a: victory and b: okay)

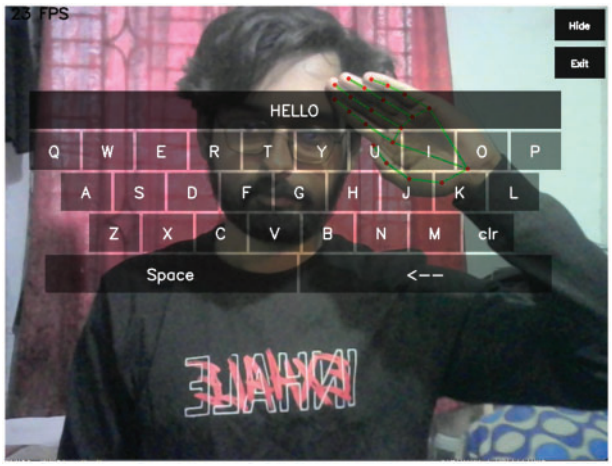

Figure 9: Operating keyboard with dynamic gestures (Hello)

Moreover, users are able to enter character input using hand-tapping gestures 303. The users are asked to enter sentences like "This is first program." Table 1 shows the 304 summary of character input.

From Table 4, we can see that the average character input is 97.86%. However, each user inputs 21 characters (including spaces), averaging 34.43 s. From this, it can be said that the user can input an average of 37 characters per minute. Reviewing the overall results, our proposed virtual keyboard system is very efficient and provides better performance in hand gesture recognition and character input. Furthermore, we compared the accuracy of character input and hand gesture recognition with state-of-the-art methods, as shown in Table 5. Table 5 shows that the recognition accuracy of hand gestures is comparable to that of state-of-the-art techniques. The recognition accuracy in this study is 98.52% in the testing environment, 96.55% in real-time operation with the LSTM model and 80.00% with the CNN-LSTM Model. Furthermore, the character input achieves 97.86% accuracy when utilizing the virtual Keyboard to insert the character.

Table 4: Character input statistics

Users	Total character	Corrected character	Accuracy (%)	Time (in s)
Subject 1	21	21	100.00	35
Subject 2	21	21	100.00	36
Subject 3	21	20	95.00	30
Subject 4	21	21	100.00	36
Subject 5	21	20	95.00	32
Subject 6	21	21	100.00	32
Subject 7	21	20	95.00	40
Average			97.86	34.43

Table 5: Comparison of recognition accuracy with state-of-the-art approaches

References	Character input/minute	Character input accuracy (%)	Methods	Reported avg. accuracy (%)
Ref. [5]	N/A	–	YOLOv3	97.68
Ref. [23]	N/A	–	Dynamic time warping	96.17
Ref. [30]	N/A	–	R-FCN	90.00
Ref. [35]	N/A	–	CNN	97.93
Ref. [4]	N/A	–	Attention-GCN	97.01
Ablation study	N/A	–	CNN-LSTM	80.00
Proposed method	21/34.43 s	97.86	LSTM & MediaPipe	98.52

5.4 Discussion

In the proposed study, we focused on enhancing non-touch communication technology, particularly in virtual keyboard input, through advanced hand gesture recognition. Our method, combining the various deep learning layers and evaluating with hand gestures, achieved a high recognition accuracy of 98.52% for the word and over 96% in character input accuracy. This opens up possibilities for improving user experience in virtual interfaces, especially for the physically impaired or in situations where touch-based interactions are not feasible. Our extensive newly created dataset, with 1,662 landmarks including detailed hand, face, and posture landmarks, forms a robust base for improving gesture-based communication technologies. This dataset can be instrumental for future research in non-touch technologies and virtual communication platforms.

Future research will focus on applying our model to fields like immersive gaming, virtual reality, and assistive technologies, using adaptive algorithms to personalize user experience. A key priority is ensuring secure and private interactions through robust cybersecurity measures for data protection. We aim to expand the model's applicability by including a wider range of gestures and linguistic characters from various cultures, enhancing its universality and inclusivity. Our findings pave the way for further exploration of non-touch technologies and virtual communication platforms. Assessing the model's effectiveness across diverse populations with unique hand gestures and cultural nuances is crucial. The research also plans to delve deeper into security and privacy aspects, which are crucial

for digital communication technologies. By leveraging our comprehensive dataset, we will explore broader applications, enhancing user interaction. Integrating adaptive algorithms for personalization and focusing on cybersecurity and privacy will help build trust in these technologies. Our goal is to make the model more inclusive by considering a broader array of cultural gestures and linguistic characters.

6 Conclusion and Future Work

This research presented an innovative character input system based on dynamic hand gestures, harnessing the power of the MediaPipe technique combined with the LSTM model. We integrated ten distinct hand gestures (Hello, Ok, Bye, Victory, Show, Thanks, Space, Clear, Backspace, Tab) to facilitate character input. Notably, these gestures were determined through the extraction of unique landmark values, a feat achieved via MediaPipe. For classification and training, the LSTM model proved instrumental. A significant contribution of our work is the introduction of a virtual keyboard, displayed onscreen. Users can intuitively perform character inputs by simply tapping their thumb and index finger. Furthermore, specialized words can be entered using specific hand gestures, enhancing the system's versatility. Our evaluations highlighted an impressive average hand gesture recognition accuracy of 98.52%. When tested in real-time across diverse users, the system maintained a commendable performance of 96.55%. For character inputs specifically, we achieved a recognition accuracy of 97.86%.

These results, when juxtaposed with existing methodologies, underline the superiority and efficiency of our proposed system. Looking forward, we aim to enrich our dataset by incorporating a wider variety of dynamic gestures, thereby enhancing our model's precision. A pivotal direction for future endeavours is adapting this system for mobile devices, enabling users to leverage their cameras for real-time gesture recognition, ensuring accessibility and convenience. While we have taken hand gestures into account for operating the virtual Keyboard, we have incorporated a limited number of commands. Moving forward, we plan to expand the gesture repertoire to enhance real-time application utility.

Acknowledgement: We would like to express our gratitude to Pabna University of Science and Technology (PUST) for their invaluable contribution in providing the dataset essential for our research. Additionally, we acknowledge the assistance of ChatGPT 3.5 in reviewing and refining the grammatical structure of the content across various sections of this paper.

Funding Statement: This work has been supported by the 2023 Research Fund of Ulsan College.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: B. Mallik, M.A. Rahim, A.S.M. Miah and J. Shin; data collection: B. Mallik and M.A. Rahim; analysis and interpretation of results: B. Mallik, M.A. Rahim and A.S.M. Miah; draft manuscript preparation: B. Mallik, M.A. Rahim, A.S.M. Miah and J. Shin; supervision: K.S. Yun and J. Shin. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The code and dataset are available at https://github.com/musaru/Virtual_Keyboard.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Zihan, N. Sang and C. Tan, "Deep learning based hand gesture recognition in complex scenes," in *MIPPR 2017: Pattern Recognition and Computer Vision*, vol. 10609, Xiangyang, China, pp. 194–200, 2017.

[2] A. S. M. Miah, J. Shin, M. A. M. Hasan and M. A. Rahim, "BenSignNet: Bengali sign language alphabet recognition using concatenated segmentation and convolutional neural network," *Applied Sciences*, vol. 12, pp. 3933, 2022.

[3] R. Regmi, S. Burns and Y. S. Song, "Humans modulate arm stiffness to facilitate motor communication during overground physical human-robot interaction," *Scientific Reports*, vol. 12, no. 1, pp. 18767, 2022.

[4] N. A. Aljalaud and H. M. Alkhaldi, "Comparative anatomical studies on the hand of human and two mammalian species," *Asian Journal of Biology*, vol. 16, no. 4, pp. 40–58, 2022.

[5] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius *et al.,* "Real-time hand gesture recognition based on deep learning YOLOv3 model," *Applied Sciences*, vol. 11, pp. 4164, 2021.

[6] M. Kim, J. Cho, S. Lee and Y. Jung, "IMU sensor-based hand gesture recognition for human-machine interfaces," *Sensors*, vol. 19, no. 18, pp. 3827, 2019.

[7] A. M. Rahim, M. R. Islam and J. Shin, "Non-touch sign word recognition based on dynamic hand gesture using hybrid segmentation and CNN feature fusion," *Applied Sciences*, vol. 9, no. 18, pp. 3790, 2019.

[8] J. Shin, A. Matsuoka, M. A. M. Hasan and A. Y. Srizon, "American sign language alphabet recognition by extracting feature from hand pose estimation," *Sensors*, vol. 21, no. 17, pp. 5856, 2021.

[9] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu *et al.,* "Whole-body human pose estimation in the wild," in *Proc. of the Computer Vision–ECCV 2020: 16th European Conf.*, Glasgow, UK, Berlin, Heidelberg, Germany, Springer, pp. 196–214, 2020.

[10] A. S. M. Miah, M. A. M. Hasan, S. W. Jang, H. S. Lee and J. Shin, "Multi-stream general and graph-based deep neural networks for skeleton-based sign language recognition," *Electronics*, vol. 12, no. 13, pp. 2841, 2023.

[11] R. E. Nogales and M. E. Benalcázar, "Hand gesture recognition using automatic feature extraction and deep learning algorithms with memory," *Big Data and Cognitive Computing*, vol. 7, no. 2, pp. 102, 2023.

[12] E. Valarezo Añazco, E. Seung, J. Han, K. Kim, P. R. Lopez *et al.,* "Hand gesture recognition using single patchable six-axis inertial measurement unit via recurrent neural networks," *Sensors*, vol. 21, no. 4, pp. 1404, 2021.

[13] M. A. Rahim, A. S. M. Miah, A. Sayeed and J. Shin, "Hand gesture recognition based on optimal segmentation in human-computer interaction," in *2020 3rd IEEE Int. Conf. on Knowledge Innovation and Invention (ICKII)*, Kaohsiung, Taiwan, pp. 163–166, 2020. https://doi.org/10.1109/ICKII50300.2020.9318870

[14] J. Xu, L. Lu and G. K. Byung, "Gesture recognition and hand tracking for anti-counterfeit palmvein recognition," *Applied Sciences*, vol. 13, no. 21, pp. 11795, 2023.

[15] W. J. Lin, J. Chu, L. Leng, J. Miao and L. F. Wang, "Feature disentanglement in one-stage object detection," *Pattern Recognition*, vol. 45, pp. 109878, 2024.

[16] S. Jiang and Y. Chen, "Hand gesture recognition by using 3DCNN and LSTM with adam optimizer," in *Proc. of Pacific Rim Conf. on Multimedia*, Hefei, China, Springer, pp. 743–753, 2017.

[17] T. L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 4489–4497, 2015.

[18] Y. Fang, K. Wang, J. Cheng and H. Lu, "A real-time hand gesture recognition method," in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, Beijing, China, IEEE, pp. 995–998, 2007.

[19] M. Oudah, A. Al-Naji and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *Journal of Imaging*, vol. 6, no. 8, pp. 73, 2020.

[20] I. Merino, J. Azpiazu, A. Remazeilles and B. Sierra, "3D convolutional neural networks initialized from pretrained 2D convolutional neural networks for classification of industrial parts," *Sensors*, vol. 21, no. 4, pp. 1078, 2021.

[21] S. Adhikari, T. K. Gangopadhayay, S. Pal, D. Akila, M. Humayun *et al.,* "A novel machine learning-based hand gesture recognition using HCI on IoT assisted cloud platform," *Computer Systems Science and Engineering*, vol. 46, no. 2, pp. 2123–2140, 2023.
[22] M. Ullah, M. M. Yamin, A. Mohammed, S. D. Khan, H. Ullah *et al.,* "Attention-based LSTM network for action recognition in sports," *Electronic Imaging*, vol. 33, pp. 1–6, 2021.
[23] Y. Wang, A. Ren, M. Zhou, W. Wang and X. Yang, "A novel detection and recognition method for continuous hand gesture using FMCW radar," *IEEE Access*, vol. 8, pp. 167264–167275, 2020.
[24] L. I. Khalaf, S. A. Aswad, S. R. Ahmed, B. Makki and M. R. Ahmed, "Survey on recognition hand gesture by using data mining algorithms," in *Proc. of Int. Cong. on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara Turkey, IEEE, pp. 1–4, 2022.
[25] S. Chioccarello, A. Sluÿters, A. Testolin, J. Vanderdonckt and S. Lambot, "FORTE: Few samples for recognizing hand gestures with a smartphone-attached radar," in *Proc. of the ACM on Human-Computer Interaction*, vol. 7, no. 1, pp. 1–25, 2023.
[26] W. Dong, L. Yang, R. Gravina and G. Fortino, "Soft wrist-worn multi-functional sensor array for real-time hand gesture recognition," *IEEE Sensors Journal*, vol. 22, pp. 17505–17514, 2021.
[27] S. Skaria, A. Al-Hourani, M. Lech and R. J. Evans, "Hand-gesture recognition using two-antenna Doppler radar with deep convolutional neural networks," *IEEE Sensors Journal*, vol. 19, pp. 3041–3048, 2019.
[28] W. Dong, L. Yang, R. Gravina and G. Fortino, "Soft wrist-worn multi-functional sensor array for real-time hand gesture recognition," *IEEE Sensors Journal*, vol. 22, no. 18, pp. 17505–17514, 2021.
[29] Y. Sugiura, F. Nakamura, W. Kawai, T. Kikuchi and M. Sugimoto, "Behind the palm: Hand gesture recognition through measuring skin deformation on the back of the hand by using optical sensors," in *2017 56th Annual Conf. of the Society of Instrument and Control Engineers of Japan (SICE)*, Kanazawa, Japan, IEEE, pp. 1082–1087, 2017.
[30] J. Si, J. Lin, F. Jiang and R. Shen, "Hand-raising gesture detection in real classrooms using improved R-FCN," *Neurocomputing*, vol. 359, pp. 69–76, 2019.
[31] M. M. H. Joy, M. Hasan, A. Ahmed, S. A. Tohfa, M. F. I. Bhuaiyan *et al.,* "Multiclass mi-task classification using logistic regression and filter bank common spatial patterns," in *COMS2 2020: Computing Science, Communication and Security*, Gujarat, India, Springer, pp. 160–170, 2020.
[32] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, pp. 3592–3607, 2011.
[33] Y. Zhang, W. Yan and A. Narayanan, "A virtual keyboard implementation based on finger recognition," in *Proc. of the 2017 Int. Conf. on Image and Vision Computing New Zealand (IVCNZ)*, Christchurch, New Zealand, IEEE, pp. 1–6, 2017.
[34] S. R. Chowdhury, S. Pathak and M. A. Praveena, "Gesture recognition based virtual mouse and keyboard," in *Proc. of the 2020 4th Int. Conf. on Trends in Electronics and Informatics (ICOEI)*, IEEE, pp. 585–589, 2020.
[35] M. A. Rahim, J. Shin and M. R. Islam, "Hand gesture recognition-based non-touch character writing system on a virtual keyboard," *Multimedia Tools and Applications*, vol. 79, pp. 11813–11836, 2020.
[36] S. Hochreiter and S. Jürgen, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 2010.