

Dynamic Hand Gesture to Text using Leap Motion

Nur Aliah Nadzirah Jamaludin¹

Faculty of Computing, University Malaysia of Computer
Science and Engineering, Cyberjaya, Selangor, Malaysia

Ong Huey Fang^{*2}

School of Information Technology, Monash University
Malaysia, Bandar Sunway, Selangor, Malaysia

Abstract—This paper presents a prototype for converting dynamic hand gestures to text by using a device called Leap Motion. It is one of the motion tracking technologies, which could be used for recognising hand gestures without the need of wearing any external devices or capturing any images and videos. In this study, five custom dynamic hand gestures of American Sign Language were created with Leap Motion to measure the recognition accuracy of the proposed prototype using the Geometric Template Matching, Artificial Neural Network, and Cross-Correlation algorithms. The experimental results showed that the prototype achieved recognition accuracy of more than 90% in the training phase and about 60% in the testing phase.

Keywords—Dynamic hand gesture; leap motion; American sign language; artificial neural network; cross-correlation; geometric template matching

I. INTRODUCTION

According to the Cambridge Dictionary, a sign language is a system of hand and body movements representing words, used by and to people who cannot hear or talk. However, not everyone knows and understand sign languages, and they are normally limited to the deaf and mute people and their families. Therefore, assistive devices or systems are required to aid their communication with others or even with machines. Furthermore, sign language is considered a powerful way for human-computer interaction as it is natural, ubiquitous, and meaningful [1, 2].

In general, hand gestures fall into two categories: 1) static; and 2) dynamic. A static hand gesture refers to a still and stable shape of the hand, while a dynamic hand gesture is movements of the hand to express a specific interaction [3]. Although dynamic hand gestures are more practical for real-time application, they require more computational complexity in processing input signals and in building recognition algorithms [4, 5]. The main problems in developing a dynamic gesture recognition system are in detecting and tracking hand shapes and hand movements, which may be complicated with other challenges such as occlusion between fingers, various trajectory, speed and amplitude of hand movements, background environment, and different styles of sign languages [5, 6].

Three technologies can be used to assist in the acquisition of hand gestures: 1) gloved-based; 2) vision-based; and 3) depth-based [3, 7]. Gloved-based technology or data glove uses wearable sensors that typically mounted on fingers and wrist to capture the position and movement of the hand. The PowerGlove is the first commercially available data glove used by the Nintendo Entertainment System as a gaming controller [8, 9]. Other examples of data gloves are such as the

Sayre Glove, CyberGloves, and Polhemus. Nonetheless, the use of data gloves is costly and inconvenience for users to wear, as they are typically wired and connected to computers [10]. In contrast, vision-based technology is more prevalent in hand gesture recognition with lower cost, noncontact, natural and intuitive [10, 11, 17]. Vision-based approaches record hand gestures by using a camera, and the captured image or video is processed and analysed with computer vision techniques. However, the remaining challenges of vision-based approaches are mainly in background, lighting, noise, and camera [11]. Besides, it usually captured a large amount of data, thus depends heavily on features preprocessing to provide better accuracy and computation rate [5].

The introduction of low-cost depth-based technology, such as Microsoft's Kinect and Leap Motion, has opened the possibility for informative and accurate dynamic gesture recognition. This approach can provide 3D information of the real world with depth information. A depth image contains not only the human body region but also the background. The user is supposed to stay close to the camera and occupies a significant portion of the camera's field of view [12]. Microsoft's Kinect is a depth camera that acquires gestures with rich depth information and has demonstrated a notable performance of light sensitivity in gesture recognition [18]. However, there are still unresolved issues regarding gesture finding and segmenting with Kinect sensors [10]. Different from the Kinect, the Leap Motion controller is a device explicitly targeted to hand gesture recognition and can directly compute the position of the fingertips and the hand orientation with better accuracy [13]. The device is smaller in size and comes with built-in recognition algorithms.

Although Leap Motion has been adopted in previous works on sign language recognition, most of them focused on static gestures instead of dynamic gestures [14, 15, 16]. The study in [16] showed that the results in static alphabet recognition were low using the built-in algorithms, namely the Geometric Template Matching (GTM), Artificial Neural Network (ANN), and Cross-Correlation (CC). Where the highest average accuracy achieved was only 52.56% with the GTM algorithm. Nevertheless, this study attempted to test further the ability of Leap Motion in recognising and converting dynamic hand gestures to text with by proposing a dynamic hand gesture recognition prototype for American Sign Language (ASL). The proposed prototype created five custom dynamic hand gestures of ASL and tested their accuracy with Leap Motion's built-in recognition algorithms.

The remaining content of this paper is organised as follows: Section II introduces the Leap Motion Controller and Leap Trainer; Section III discusses the design for the proposed

*Corresponding Author.

prototype; Section IV discusses on the experimental results, and finally, the conclusion was given in Section V.

II. LEAP MOTION

This section explains the overall process in building a dynamic hand gestures recogniser by using Leap Motion. It includes the illustration of workflow for Leap Motion Controller and Leap Trainer.

A. Leap Motion Controller

Leap Motion is a small device connected to USB and can recognise hand motions or gesture accurately when no occlusion happens without the need of wearing any gloves or instruments. The device consists of three Infrared (IR) Light emitters and two cameras. The cameras received the IR lights and track hand gesture up to 200 frames per second and 150° field of view with approximately 8 cubic feet of interactive 3D space [19].

The Leap Motion Controller has a wide range of application. For example, this device had been used for stoking rehabilitation by people from The Intelligent Computer Tutoring Group at the University of Canterbury [20]. The tracking data of the Leap Motion Controller, which contains the palm and fingers' position, direction, velocity can be accessed using its Software Development Kit.

B. Leap Trainer

Leap trainer is one of the main application used to create a custom gesture for Leap Motion Controller. As for now, Leap Motion only supports four gestures, which are a swipe, circle, screen tap and key tap. Circle and swipe gestures are continuous, and these objects can have a state of start, update, and stop. While the screen tap gesture is a discrete gesture [21], and it always has a stop state. However, these four gestures are not enough to cover all the hand gestures available in the ASL. Each sign language has a different sign to convey the same meaning in different languages. There are two ways of creating custom gestures using Leap Motion device, which are: 1) writing code to recognise a custom gesture, and 2) allowing the device to learn a gesture which can be recognised.

Fig. 1 below shows the workflow of how a Leap Trainer trained the custom gestures by using the Leap Motion Controller. The description for each component in the workflow is as below:

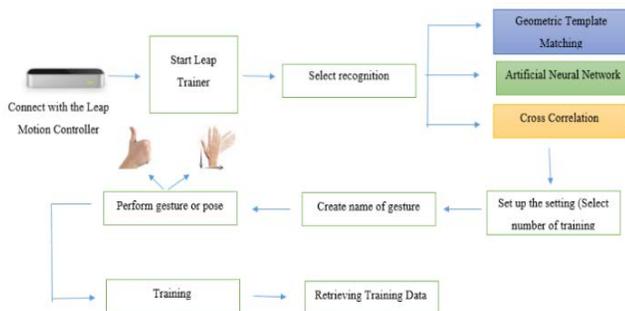


Fig. 1. The Workflow of a Leap Motion Controller.

- Connect with the Leap Motion Controller: To run the Leap Trainer, the user must connect to the Leap Motion Controller.
- Start Leap Trainer: If the device is connected, it detects user hand movements on top of the Leap Motion device.
- Select recognition: Before start creating custom gesture, the user may need to select an algorithm they want to use. The choices are Geometric Template Matching, Artificial Neural Network and Cross-Correlation.
- Set up the setting: When the user clicks the Options button, they can modify settings that have been provided. User is advised to change the number of training gesture since it helps to increase the recognition of the custom hand gestures.
- Create name of the gesture: After done with the setting part, the user can put a name at the textbox available at the top left to start creating custom hand gestures.
- Perform gesture or pose: User can perform any motions, either gesture or pose. A gesture means the hand gesture is intended to change over a period of time, whereas pose is observed at the spurt of time.
- Training: After a few times of training gesture (according to the number of training gesture that has been set up), the application now try to learn the gesture and user can start repeating the gesture for training the algorithm to learn the hand gesture.
- Retrieving training data: Lastly, after the algorithm had learned the hand gesture, the user can retrieve the data and save it into the database.

III. PROPOSED PROTOTYPE

This section presents designs of the proposed prototype, including the hand gesture recognition process, the user interface design and the database design.

A. Design Workflow

Fig. 2 shows the design workflow used in this paper with two main phases. The first phase is the training phase. The training phase is part of the process to train the data, extract data and save it into the database. The data need to be trained to get the best result during the translation of the motion of hand gesture to text. The second phase is the real-time phase, where the testing of hand gesture recognition is done real-time with the techniques of GTM, ANN, and CC. This phase will try to match a gesture with the most suitable text and give an output via the prototype.

B. Sign Trainer

A Sign Trainer page allows the user to create a custom gesture, to train hand gestures and store data into the database. The creation of custom gestures can be recognised using three different techniques which are the GTM, ANN, and CC. This study used several specific programming languages and scripting, which are JavaScript, HTML, CSS and image files. Python files also created to store custom gesture data into the

database. Fig. 3 shows the workflow and architecture of a Sign Trainer. The main parts are:

- Connect with Leap Motion Controller
- Start creating hand gestures
- Store custom gesture data
- Load hand gesture in the database.

The user interface for Sign Trainer page is shown in Fig. 4, which allows users to add custom gestures. This custom gesture can be created according to three different classifiers which are the GTM, ANN, and CC. Before doing the customisation of hand gestures, users are allowed to change the settings listed and choose a recognition algorithm when they clicked on the Options button. The custom gestures data will produce in a JSON file.

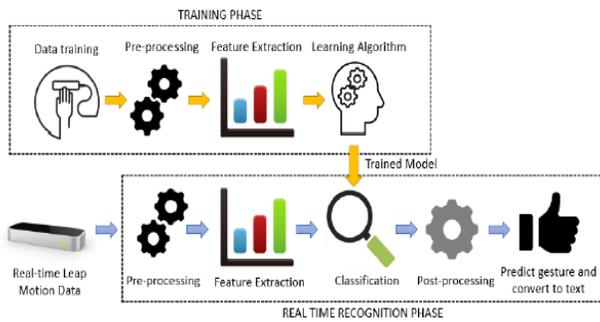


Fig. 2. The Workflow of Hand Gesture Recognition.

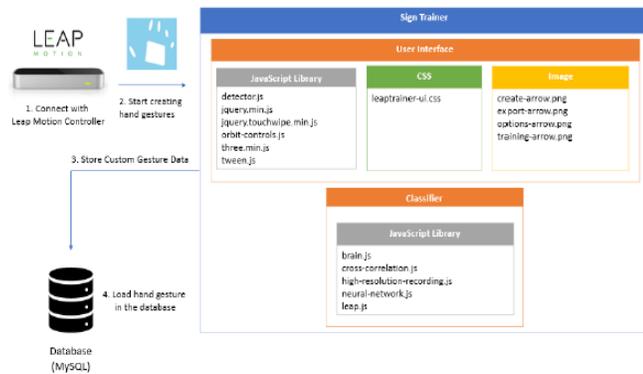


Fig. 3. Architecture and Workflow of Sign Trainer.

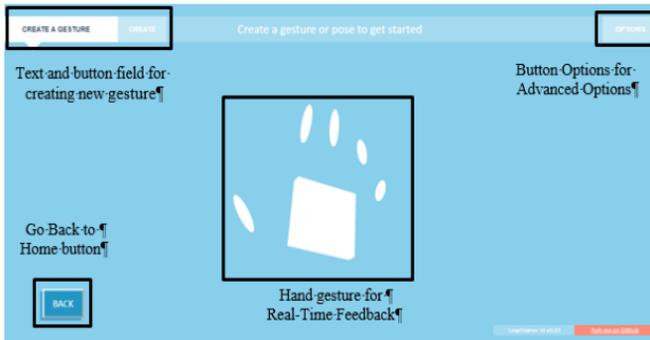


Fig. 4. Sign Trainer Page.

There are some other options provided on this page, as shown in Fig. 5. These options are the default implementations of an algorithm and may be removed or redundant in sub-classes of the Sign Trainer controller. The Option button also let the user choose one of the techniques as the recognition strategy, which are the GM, ANN or CC.

C. Sign Translator

Sign Translator is the page to convert a hand gesture into corresponding text for ASL. This page first connects to the MySQL database all the gesture data available. If the user wants to use this application, they first need to choose the three main algorithms that have been implemented in the Sign Trainer. The workflow and architecture, as shown in Fig. 6, and the main parts of Sign Translator are:

- Connect with Leap Motion Controller
- Retrieve hand gesture data from the database
- Load the stored data
- Output result

Fig. 7 shows the page that intended to convert the sign language into text. The user can perform any sign while connecting to the Leap Motion Controller. Then, after one of the algorithms been chosen, a pop-up alert will appear and shows that the stored data already loaded to the application. Automatically, the application will later look for a match of the current hand gesture with the stored data. If a match occurs, the result with percentage of matching will appear on top of the screen, as shown in Fig. 8.



Fig. 5. Options in Sign Trainer Page



Fig. 6. Architecture and Workflow of Sign Translator Page.

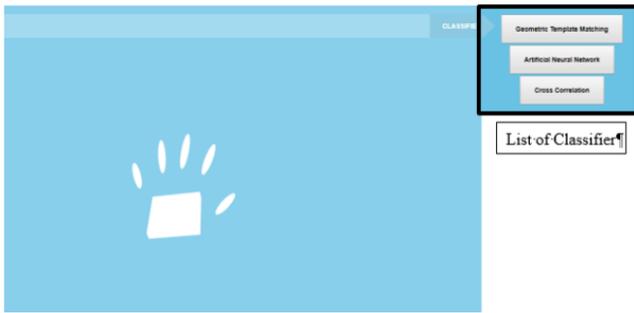


Fig. 7. List of Classifiers in Sign Translator Page.



Fig. 8. List of Hand Gestures in Database.

IV. EXPERIMENT

This section presents the experiments conducted using the proposed prototype that runs on the Leap Motion Controller. The results of in recognising dynamic hand gesture into text are also discussed.

A. Experiment Setup

This study performed two experiments to test the accuracy of the training phase and the testing phase. The tests were held to find out the performance of the proposed prototype. The accuracy value is calculated by using the following formula:

$$Accuracy = \frac{\text{(Number of correct signs)}}{\text{(Number of signs performed)}} \quad (1)$$

Five datasets were used in this study. For each gesture, the number of tests conducted was two times, and the average accuracy was calculated. Fig. 9 shows the dynamic hand gestures that were used in this study, namely the sign language of 'YES', 'NO', 'CIRCLE', 'BOOK', and 'HOUSE'.

B. Training Results

Table I to Table III shows the results of the training phase that had been conducted on the Sign Training page. The results show that the ANN technique achieved the highest accuracy in recognising the five dynamic sign language.

The bar chart shown in Fig. 10 compared the results obtained from three different algorithms, which are GTM, ANN, and CC. The comparison shows that ANN always gets the highest accuracy in every type of gesture. Besides, the results suggest all the three techniques struggle in recognising particular signs such as 'HOUSE', 'NO', and 'BOOK'. The CC technique was not able to detect the hand gesture of 'HOUSE', although given several trials.

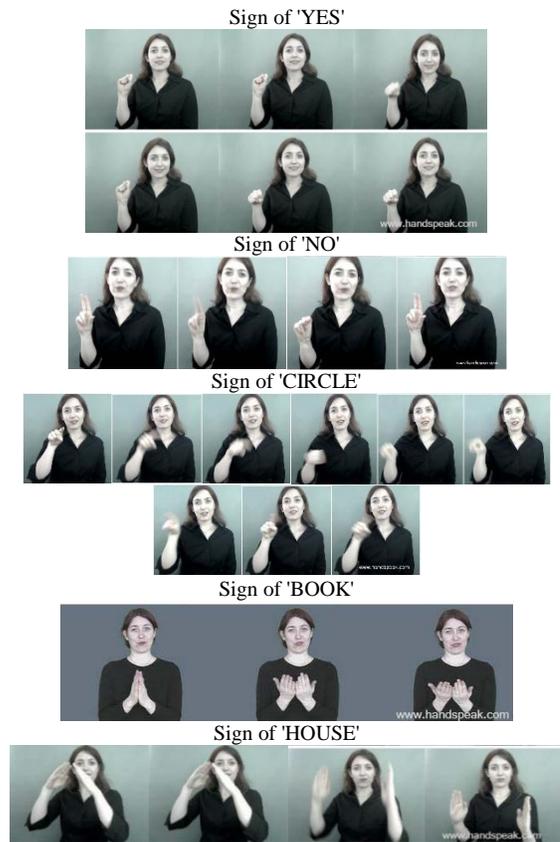


Fig. 9. Dynamic Hand Gestures Tested in this Study.

TABLE. I. RESULTS OF DYNAMIC HAND GESTURES RECOGNITION USING GEOMETRIC TEMPLATE MATCHING

Gesture	No. of training gesture	Threshold	Accuracy (%)
YES	2	0.65	96.33
NO	2	0.65	67.33
CIRCLE	2	0.65	48.33
BOOK	2	0.65	72.00
HOUSE	2	0.65	23.33

TABLE. II. RESULTS OF DYNAMIC HAND GESTURES RECOGNITION USING ARTIFICIAL NEURAL NETWORK

Gesture	No. of training gesture	Threshold	Accuracy (%)
YES	2	0.65	98.00
NO	2	0.65	96.00
CIRCLE	2	0.65	91.33
BOOK	2	0.65	95.00
HOUSE	2	0.65	95.33

TABLE. III. RESULTS OF DYNAMIC HAND GESTURES RECOGNITION USING CROSS-CORRELATION

Gesture	No. of training gesture	Threshold	Accuracy (%)
YES	2	0.65	83.67
NO	2	0.65	23.33
CIRCLE	2	0.65	70.67
BOOK	2	0.65	43.33
HOUSE	2	0.65	0.00

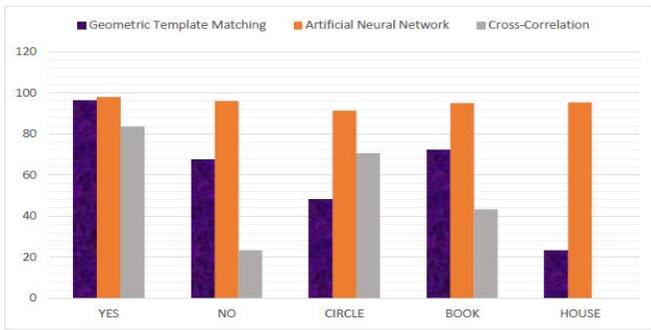


Fig. 10. Comparison of Recognition Training Results for Geometric Template Matching, Artificial Neural Network and Cross-Correlation.

C. Testing Results

For the testing results, only GMM was implemented in the experiment. Table IV shows the results of GMM in converting hand gesture to text. The results of the cumulative average accuracy of translating hand gesture to a text are 60.00%, as shown in Fig. 11. This accuracy can be a good start for more improvement in future work.

TABLE IV. RESULTS OF TESTING USING GEOMETRIC TEMPLATE MATCHING

Gesture	Test #1	Test #2	Test #3	Average
YES	1	1	1	100.00%
NO	1	1	0	66.67%
CIRCLE	1	0	1	66.67%
BOOK	0	1	0	33.33%
HOUSE	1	0	0	33.33%
Cumulative				60.00%

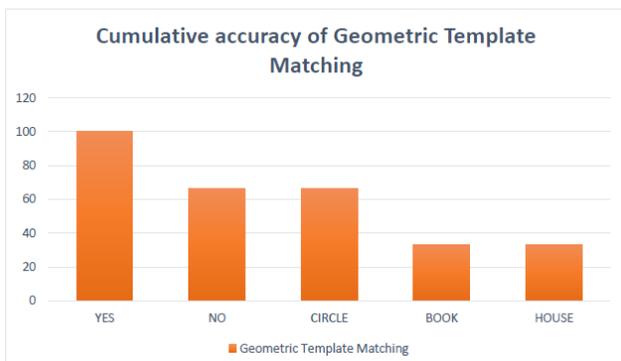


Fig. 11. Cumulative Accuracy of Geometric Template Matching.

V. CONCLUSION

This paper proposed a prototype that can convert dynamic hand gestures into text by using one of the depth-based technologies known as Leap Motion. The objective of the prototype is to be a medium of communication between deaf and mute by using a lightweight device that is capable of detecting and tracking hand motions. However, the experimental results suggest that Leap Motion was unable to track finger position precisely, especially when using both hands to do the 'HOUSE' gesture, and when fingers are in the

first position. Most of the time, the hand gesture of 'YES' is the fastest and easiest to be detected. This gives hints on the tracking capability of Leap Motion and the recognition algorithms required for improvement in tracking similar dynamic hand gestures more precisely and accurately.

Nonetheless, with Leap Motion technology, it has become possible for converting hand gesture to text to recognise various actions that can be interpreted into words and understandable for the deaf and mute people. For future work, improvement can be made in term of performance and accuracy, where better gesture recognition algorithms are required not only to recognise dynamic gestures that can be translated into a word, but also to recognise continuous dynamic gestures that can be translated into sentences. Moreover, better ways of capturing tracking data, creating custom gestures and recognising gestures are needed for Leap Motion in building an effective dynamic hand gesture to text system.

REFERENCES

- [1] P. Premaratne, S. Yang, Z. Zhou, and N. Bandara, "Dynamic Hand Gesture Recognition Framework," in *Intelligent Computing Methodologies*, Cham, 2014, vol. 8589, pp. 834-845: Springer International Publishing.
- [2] H. Cheng, L. Yang, and Z. Liu, "Survey on 3D Hand Gesture Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1659-1673, 2016.
- [3] J. S. Sonkusare, N. B. Chopade, R. Sor, and S. L. Tade, "A Review on Hand Gesture Recognition System," in *2015 International Conference on Computing Communication Control and Automation*, 2015, pp. 790-794.
- [4] J. Jia, G. Tu, X. Deng, C. Zhao, and W. Yi, "Real-time hand gestures system based on leap motion," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 10, pp. e4898, 2019.
- [5] Z.-j. Liang, S.-b. Liao, and B.-z. Hu, "3D Convolutional Neural Networks for Dynamic Sign Language Recognition," *The Computer Journal*, vol. 61, no. 11, pp. 1724-1736, 2018.
- [6] X. Zhang, J. Wang, X. Wang, and X. Ma, "Improvement of Dynamic Hand Gesture Recognition Based on HMM Algorithm," in *2016 International Conference on Information System and Artificial Intelligence (ISAI)*, 2016, pp. 401-406.
- [7] R. R. Itkarkar and A. V. Nandi, "A survey of 2D and 3D imaging used in hand gesture recognition for human-computer interaction (HCI)," in *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2016, pp. 188-193.
- [8] M. Kadous, "Machine Recognition of Auslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language," 1970.
- [9] C. Preetham, G. Ramakrishnan, S. Kumar, A. Tamse, and N. Krishnapura, "Hand Talk-Implementation of a Gesture Recognising Glove," in *2013 Texas Instruments India Educators' Conference*, 2013, pp. 328-331.
- [10] X. Ma and J. Peng, "Kinect Sensor-Based Long-Distance Hand Gesture Recognition and Fingertip Detection with Depth Information," *Journal of Sensors*, vol. 2018, pp. 9, 2018.
- [11] Y. Zhu, Z. Yang, and B. Yuan, "Vision Based Hand Gesture Recognition," in *2013 International Conference on Service Sciences (ICSS)*, 2013, pp. 260-265.
- [12] C. Wu and C. H. Lin, "Depth-based hand gesture recognition for home appliance control," in *2013 IEEE International Symposium on Consumer Electronics (ISCE)*, 2013, pp. 279-280.
- [13] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 1565-1569.

- [14] A. Vaitkevičius, M. Taroza, T. Blazauskas, R. Damasevicius, R. Maskeliunas, and M. Woźniak, "Recognition of American Sign Language Gestures in a Virtual Reality Using Leap Motion," *Applied Sciences*, vol. 9, p. 445, 2019.
- [15] T. W. Chong and B.-G. Lee, "American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach," (in eng), *Sensors (Basel, Switzerland)*, vol. 18, no. 10, p. 3554, 2018.
- [16] F. Khan, H. Ong, and N. Bahar, "A Sign Language to Text Converter Using Leap Motion," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, p. 1089, 2016.
- [17] B. Garcia, S. A. Viesca (2014) Real-time American Sign Language Recognition with Convolutional Neural Networks.
- [18] D. H. Vo, H. H. Huynh, P. M. Doan and J. Meunier, "Dynamic Gesture Classification for Vietnamese Sign Language Recognition" *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 3, p 2017.
- [19] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," (in eng), *Sensors (Basel, Switzerland)*, vol. 13, no. 5, pp. 6380-6393, 2013.
- [20] A. Bracegirdle, T. Mitrovic, and M. Mathews (2014). Investigating the usability of the leap motion controller. Gesture-based interaction with a 3d virtual environment.
- [21] R. O'Leary (2013). Leaptrainer.js. Retrieved from Github: <https://github.com/roboleary/LeapTrainer.js>.