

A Sign Language to Text Converter Using Leap Motion

Fazlur Rahman Khan[#], Huey Fang Ong^{*}, Nurhidayah Bahar[#]

[#]School of Information Technology, UCSI University, Kuala Lumpur, Malaysia
E-mail: f.r.khan.trk@gmail.com, nurhidayah@ucsiuniversity.edu.my

^{*}Faculty of Computing, University Malaysia of Computer Science & Engineering, Malaysia
E-mail: onghueyfang@unimy.edu.my

Abstract— This paper presents a prototype that can convert sign language into text. A Leap Motion controller was utilised as an interface for hand motion tracking without the need of wearing any external instruments. Three recognition techniques were employed to measure the performance of the prototype, namely the Geometric Template Matching, Artificial Neural Network and Cross Correlation. 26 alphabets from American Sign Language were chosen for training and testing the proposed prototype. The experimental results showed that Geometric Template Matching achieved the highest recognition accuracy compared to the other recognition techniques.

Keywords— sign language to text; Leap Motion; Geometric Template Matching; Artificial Neural Network; Cross Correlation; American sign language

I. INTRODUCTION

Hearing is one of the most essential human senses, yet not everyone possesses this gift. According to the World Health Organization, approximately 360 millions of people worldwide are affected by disabling hearing loss [1]. Sign language is typically used to aid communication for deaf and hearing loss people. However, not everyone knows sign language, thus, it has become harder for deaf people to communicate with others. To fill in the gap, researchers have been working on building automatic sign language translators, which are capable of converting sign language into text or speech.

Sign Language to Text Converter (SLTC) is a type of application that captures sign language performed by a user and translates it into texts with the assistance of virtual technologies and devices. There are few promising technologies that can aid in sign language to text conversions, such as glove technologies, wrist sensors, two-dimensional (2D) cameras, three-dimensional (3D) cameras and kinetic platforms [2]. Along with that, there are various gesture recognition techniques that have been introduced in the literature for building SLTC. Gesture recognition can be defined as the identification of meaningful expression or motion performed by a human being, which involves the movement of hand, arm, face, head and body [3]. The

applications of gesture recognition can be ranging from sign language through medical rehabilitation to virtual reality.

PowerGlove from the Nintendo Entertainment System was used with machine learning techniques for recognising Australian Sign Language (Auslan) [4]. The proposed recognition system achieved an accuracy of approximately 80%, but the remaining limitation is found lies in the glove itself. Similarly, [5] had proposed a sign language recognition system known as SLARTI. The system employed neural network and nearest-neighbour algorithms to recognise gestures from Auslan. It works by using CyberGloves and Polhemus IsoTrak to track hand gestures. With these devices, users can press a switch on their hand to indicate the starting and end of a gesture. SLARTI had shown a promising recognition result by achieving 94% of accuracy for trained signers, and 85% of accuracy for non-trained signers.

In another work, [6] developed the Hand Motion Understanding (HMU) system that incorporated hand tracking aided by a color-coded glove and recognition of static and dynamic signs by an adaptive fuzzy expert system for Auslan. The SLTC achieved 91% of accuracy without training, and 95% of accuracy with training, but with only 20 signs selected. While for another regional sign language, [7] had classified static and dynamic gestures with 40 words vocabulary of American Sign Language (ASL). They employed Hidden Markov Model (HMM) technique and achieved an accuracy of 99.2%. Besides that, [8] used HMM

to classify dynamic gestures for Japanese Sign Language and achieved the accuracy of 98.4% for 65 words vocabulary.

Currently, there are very few products available in the market to facilitate deaf people in communicating with normal hearing people. Most of the existing products do not include a real time functionality to convert sign language into text. For instance, ASL Translator [9] and iCommunicator [10] only allow the conversion of English words into video representations of sign language. Moreover, the available products so far are meant to recognise ASL, thus unable to facilitate people around the world who uses different languages other than English.

Apart from systems and products described above, studies have been conducted to employ Leap Motion controller in building SLTC. Leap Motion is a small device connected via USB and is capable of tracking hand motions or gestures without the need of wearing any gloves or instruments. This study aims to develop a prototype of SLTC by using a Leap Motion controller. Three gesture recognition techniques, namely the Geometric Template Matching, Artificial Neural Networks (ANN) and Cross-Correlation were used to recognise 26 static hand gestures. The development of the proposed prototype has shown that Leap Motion is a promising technology for developing an SLTC.

Gesture recognition is a mathematical interpretation of a human motion by a computing device. Different techniques have been proposed to acquire necessary information for gesture recognition systems. Some techniques use additional hardware devices such as data glove devices and colour markers to easily extract a comprehensive description of gesture features. While other techniques extract necessary features based on the appearance of the hand, such as capturing the skin colour. These methods are considered easy, natural and less costly compared to other methods.

Several applications have been developed so far in the field of sign language to text conversion. Different applications use different technologies and techniques to develop a fully functional SLTC system. Few of the most promising works are discussed here.

A. Sign Language Recognition (SLARTI)

SLARTI was developed by [5], and ANN technique was employed to recognise Auslan. The system comprised of hand glove and two wrist sensors. The glove is known as Cyber-Glove. A switch was placed on user's left hand to indicate the start of a gesture. The system showed a promising result, where the accuracy for trained signers was 94% and the accuracy for non-trained signers was 85%. Trained signers are those who were trained to use the system, and their data were collected to train the gesture recogniser. Whereas non-trained signers are those who have never been trained to use the system and their data were not used to train the recogniser.

B. Hand Motion Understanding (HMU)

HMU system was developed by [6] to recognise Auslan. It comprised of color-coded glove and fuzzy logic to track and recognise hand gestures. Fuzzy set theory allows the description of sign data in natural and imprecise manners. The system's classifier has an adaptive fuzzy inference

engine that trains the system to be adaptive to errors caused by tracker or motion variations exhibited among signers. The HMU system was evaluated with 22 static and dynamic Auslan signs, and it can recognise 20 signs before training, and 21 signs after training of the classifier. Although the system had shown promising accuracies for simple gestures, it was unable to identify complex gestures. A gesture can be static (posture or sign), which require less computational complexity, or dynamic (sequence of postures), which are more complex and suitable for real time environments.

C. Enable Talk

Enable Talk provides two gloves, which are equipped with sensors and can recognise sign language and translate it into text on a smart phone. The text can be translated to spoken words as well. However, the development of this project is still in progress and the device needs further development. Therefore, currently, there is no commercially available product in the market for SLTC yet [18].

D. ASL Translator

ASL Translator is a text to sign language converter that is developed by Sigtel Inc. This application translates English words into equivalent video representation of ASL. It has a huge library of videos for all the alphabet and various words of English language. Two main features of this application are Text-to-Sign Generator and ASL Phrases. The Text-to-Sign Generator is capable of translating English texts into corresponding ASL and video representation. ASL Translator can take up to 50 words at a time and has a library of videos for over 30000 words of English language. The ASL Phrases feature allows normal hearing people to learn ASL idioms and phrases. The application provides over 1400 idioms and phrases [9].

E. iCommunicator

iCommunicator promotes independent communication for a person who is deaf or hard-of-hearing and improves literacy by "translating" English in a number of ways. It has three main features, which are Speech to Text, Speech/Text to Video Sign-Language, and Speech/Text to Computer Generated Voice. Like ASL Translator, this application does not provide any way to convert sign language into text. It provides one-way communication by showing ASL video for corresponding English word [10].

The rest of the paper is organised as follows. First, the next section discusses the material and method where Leap Motion was employed for this experiment. Then, the proposed SLTC's design are presented. The following section discusses the experiment conducted and its result. Finally, the last section concludes this paper.

II. MATERIAL AND METHOD

This study employed a Leap Motion controller to track hand gestures and implemented a gesture recognition framework known as Leap Trainer. This section provides brief descriptions of how Leap Motion controller and Leap Trainer works.

A. Leap Motion Controller

Leap Motion controller is a small USB device and was first introduced to the market in 2013 by Leap Motion Inc. It can track hand up to 200 frames per second and 150° field of view with approximately 8 cubic feet of interactive 3D space [11]. It allows actions like pinching, crossing fingers, moving one hand over another, and other kinds of hand-to-hand interactions [12]. Using two monochromatic infrared (IR) cameras and three LED lights, the device generates 3D patterns of IR lights, which is sent to the host computer via a USB cable. The data captured are then analysed using "complex math", where in some way synthesising 3D position data by comparing with 2D frames generated by the two cameras.

A Leap-enabled application can receive the motion tracking data by accessing the Leap Motion application program interface (API). The Leap Motion's software development kit (SDK) provides two varieties of API to get tracking data from the Leap Motion service, namely the native interface and the WebSocket interface [13].

This study utilised Leap Motion's WebSocket API to receive tracking data. According to [13], the Leap Motion service runs a WebSocket server on a localhost domain at port 6437. The WebSocket server provides tracking data in the form of JSON messages to the web application. A JavaScript client library is used to establish a connection to the server and to consume the JSON messages. Then the tracking data is presented as regular JavaScript object. The workflow of a Leap Motion and WebSocket is shown in Fig. 1.

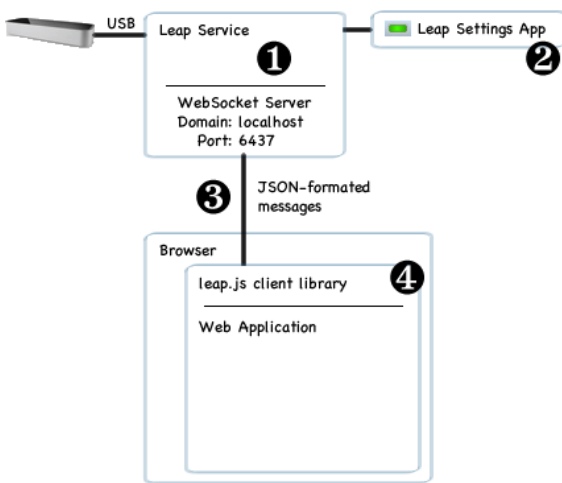


Fig. 1 Workflow of Leap Motion and WebSocket

B. Leap Trainer

Leap Trainer classifies all types of gesture into two categories, namely pose and gesture. The pose is a static gesture that does not involve any motion. On the contrary, the gesture is a hand movement with a recognisable start and end, such as a wave or a swipe from the left to right direction [14]. Leap Trainer is used to create custom gestures for Leap Motion as the current SDK only accepts four basic gestures, which are swipe, circle, screen tap and key tap. In SLTC, each sign language has a different sign to convey the same meaning in different language. Therefore, it is necessary to have a feature that enables developers to

create custom gestures. Fig. 2 depicts the workflow of a Leap Trainer.

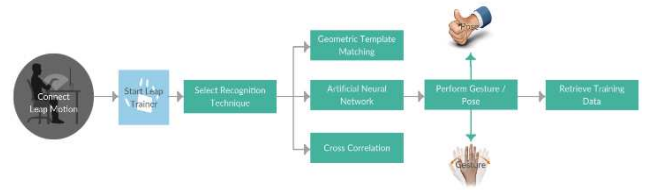


Fig. 2 Workflow of Leap Trainer

Leap Trainer allows the use of three different gesture recognition techniques, which are ANN, Cross Correlation and Geometric Template Matching. ANN is relatively crude electronic models based on the neural structure of the brain and it promises a less technical way to develop machine solutions. Cross Correlation is a mathematical model to measure the similarity of two series as a function of the lag of one relative to the other. It is a standard method of estimating the degree to which two series are correlated [15].

Template Matching is a technique in digital image processing for finding small parts of an image that match a template image [16]. Template Matching requires two sets of images, source image and template image. The Geometric Template Matching implementation for Leap Trainer is based on \$P\$ Point-Cloud Recognizer. According to [17], \$P\$ Point-Cloud Recognizer is a 2D gesture recogniser designed for rapid prototyping of gesture-based user interface.

III. RESULT AND DISCUSSION

The proposed SLTC prototype was developed with the Leap Motion controller and gesture recognition techniques used in Leap Trainer. The prototype has three modules, namely Sign to Text Converter, Sign Viewer and Sign Trainer. The SLTC runs on web browser and uses the MySQL database to store the gesture data. The architecture of the proposed SLTC prototype is shown in Fig. 3. The functionality of each module is explained in the following sections.

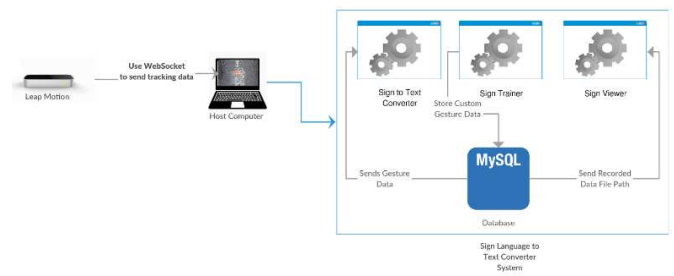


Fig. 3 Architecture of the proposed SLTC prototype

A. Sign to Text Converter

In Sign to Text module, a gesture is converted into its corresponding ASL. This module connects to a MySQL database that contains all gesture data. It also fetches custom gesture data created by users and receives tracking data from the Leap Motion controller. It will look for matching between the received gestures and the stored data. If any matches occurred, the output will be shown on the screen

instantly. This module comprises of JavaScript, CSS, HTML and PHP files and the workflow is shown in Fig. 4.

The corresponding web page design for this module is illustrated in Figure 5. At this page, the prototype will keep listening for any gestures that match the gestures in the database. If it found matches, it will show the text instantly on the field located the right side of the 3D hand.

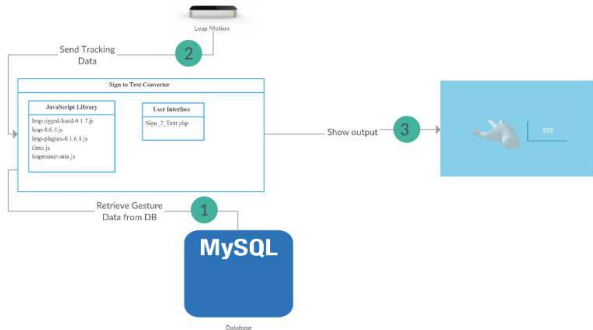


Fig. 4 Workflow of Sign to Text Module

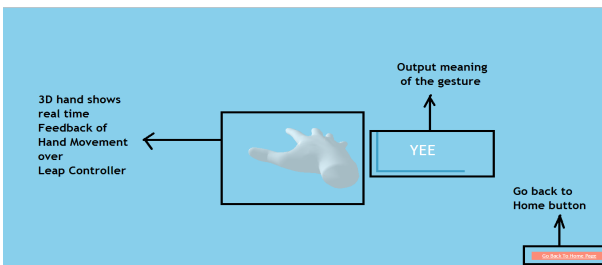


Fig. 5 Sign to Text Page

B. Sign Trainer

Sign Trainer module allows users to create custom gestures and store data into the database. It allows users to train and create custom gestures by changing the parameters of gesture recognition technique used. This module consists of several JavaScript, HTML, CSS and image files. Fig. 6 depicts the workflow of Sign Trainer. While the corresponding web page design for this module is illustrated in Fig. 7. At this page, the user can add custom gestures. To store the data of the gestures created, the save button is included to transfer SQL statement and JSON data to a PHP file, which establishes a connection to the database and store the data into designated table.

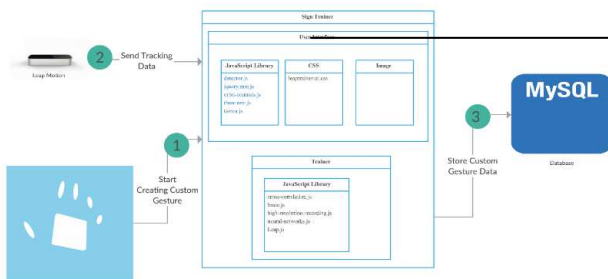


Fig. 6 Workflow of Sign Trainer Module

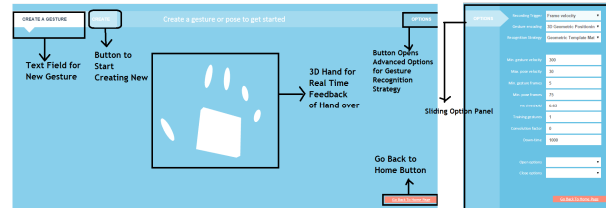


Fig. 7 Sign Trainer Page

C. Sign Viewer

Sign Viewer module shows the sign of the corresponding alphabet or word selected by users. It also consists of JavaScript, HTML, CSS and PHP files. The workflow of this module is shown in Fig. 8 and the web page design is shown in Fig. 9. Users are allowed to choose an alphabet or a word from a dropdown list and render the corresponding sign using 3D hand shown on the screen.

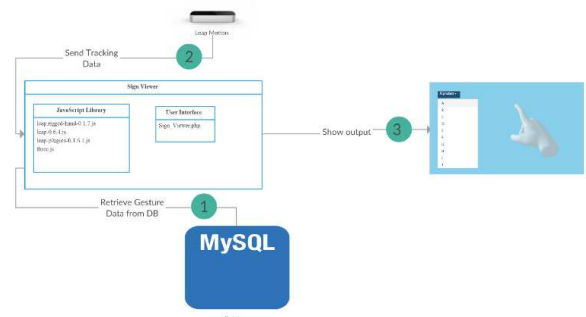


Fig. 8 Workflow of Sign Viewer Module

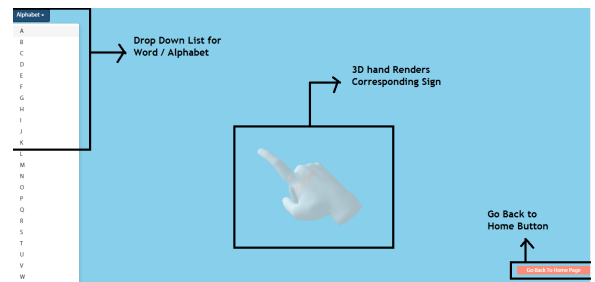


Fig. 9 Sign Viewer Page

This study performed an accuracy test to find out the performance of the proposed prototype. The accuracy value indicates how many gestures can be correctly recognised by the prototype and is calculated by using the formula in (1).

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

26 alphabets from ASL were used to calculate the accuracy using three gesture recognition techniques, namely Artificial Neural Network, Cross Correlation and Geometric Template Matching. Fig. 10 shows the alphabet chart for ASL, where the alphabet with an asterisk (*L left or *R right) is shown from the side view.

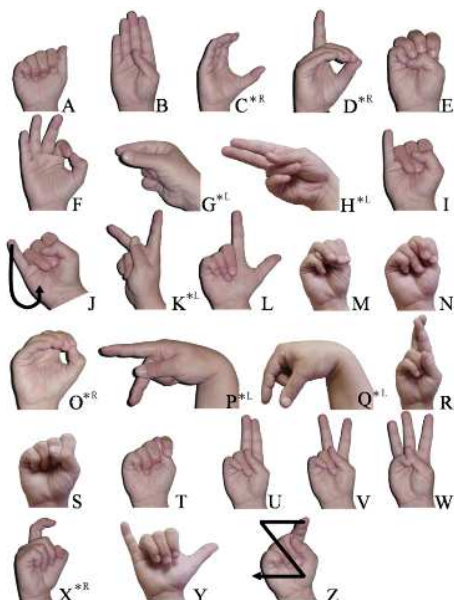


Fig. 10 Alphabets for ASL

Table 1, Table 2, and Table 3 shows the results obtained for these three recognition techniques. For each alphabet, the number of tests conducted was three times and the average accuracy was calculated. While cumulative accuracy was calculated from all the alphabets tested. Among these three recognition techniques, Geometric Template Matching achieved the highest accuracy of approximately 52.56% for alphabet recognition. While ANN and Cross-Correlation achieved 44.87% and 35.90% of accuracy respectively.

The experimental results suggest that Leap Motion unable to track finger position precisely especially when fingers are in the first position. As shown in Fig. 10, both signs for A and E are quite similar. For A, the thumb is kept straight, whereas, for E, the thumb is curled and set under four fingers. Only Geometric Template Matching is able to recognize sign E twice out of three tries. However, the other two techniques failed completely.

Sign for alphabet M, N, S and T also look similar. To perform these signs, all four fingers of the users are curled into the palm which is similar to the fist position, except that the thumb position is different. As a result, often when sign M is performed, Leap Motion is unable to track it properly, and often gesture recognition techniques mismatch M with either N, S or T. This gives a hint that tracking capability of Leap Motion and the recognition algorithms require further improvement in order to track similar signs more precisely and accurately.

Based on the experiment conducted, different techniques serve the best result for different gestures, and none of these algorithms is well enough to cover all types of static and dynamic gestures as shown in Fig. 10. Fig. 11 shows a comparison of the three techniques used. The bar chart suggests all the three techniques struggle to recognise particular signs such as E, N, P, R and S. While none of the techniques successfully recognised the sign of T, although given several trials.

TABLE I
RESULTS OF ALPHABET RECOGNITION USING
GEOMETRIC TEMPLATE MATCHING

Alphabet	Test#1	Test#2	Test#3	Average
A	1	1	1	100.00%
B	1	1	1	100.00%
C	1	0	0	33.33%
D	1	1	1	100.00%
E	1	1	0	66.67%
F	1	1	0	66.67%
G	1	0	0	33.33%
H	1	1	0	66.67%
I	1	1	1	100.00%
J	1	0	1	66.67%
K	1	0	0	33.33%
L	1	1	0	66.67%
M	1	0	0	33.33%
N	1	0	0	33.33%
O	1	1	0	66.67%
P	0	0	0	0.00%
Q	1	1	0	66.67%
R	0	0	0	0.00%
S	0	1	0	33.33%
T	0	0	0	0.00%
U	0	0	1	33.33%
V	1	1	0	66.67%
W	1	1	0	66.67%
X	1	0	0	33.33%
Y	1	1	0	66.67%
Z	1	0	0	33.33%
Cumulative Average				52.56%

TABLE II
RESULTS OF ALPHABET RECOGNITION USING
ARTIFICIAL NEURAL NETWORK

Alphabet	Test#1	Test#2	Test#3	Average
A	1	1	1	100.00%
B	1	1	1	100.00%
C	0	0	0	0.00%
D	1	1	0	66.67%
E	0	0	0	0.00%
F	1	0	0	33.33%
G	1	0	0	33.33%
H	1	0	0	33.33%
I	1	1	1	100.00%
J	1	0	0	33.33%
K	1	1	0	66.67%
L	1	0	0	33.33%
M	1	1	0	66.67%
N	0	0	0	0.00%
O	1	1	0	66.67%
P	1	0	0	33.33%
Q	1	1	0	66.67%
R	1	0	0	33.33%
S	0	0	0	0.00%
T	0	0	0	0.00%

U	1	0	1	66.67%
V	1	1	0	66.67%
W	1	1	0	66.67%
X	1	0	0	33.33%
Y	1	0	0	33.33%
Z	1	0	0	33.33%
Cumulative Average				44.87%

TABLE III
RESULTS OF ALPHABET RECOGNITION USING
CROSS CORRELATION

Alphabet	Test#1	Test#2	Test#3	Average
A	1	1	0	66.67%
B	1	0	1	66.67%
C	1	0	0	33.33%
D	0	1	0	33.33%
E	0	0	0	0.00%
F	1	0	0	33.33%
G	0	0	0	0.00%
H	1	0	0	33.33%
I	1	1	1	100.00%
J	1	1	0	66.67%
K	0	0	0	0.00%
L	1	0	0	33.33%
M	1	0	0	33.33%
N	0	0	0	0.00%
O	1	1	1	100.00%
P	0	0	0	0.00%
Q	0	1	0	33.33%
R	0	0	0	0.00%
S	0	0	0	0.00%
T	0	0	0	0.00%
U	1	0	0	33.33%
V	1	1	0	66.67%
W	1	0	0	33.33%
X	1	1	0	66.67%
Y	1	1	0	66.67%
Z	1	0	0	33.33%
Cumulative Average				35.90%

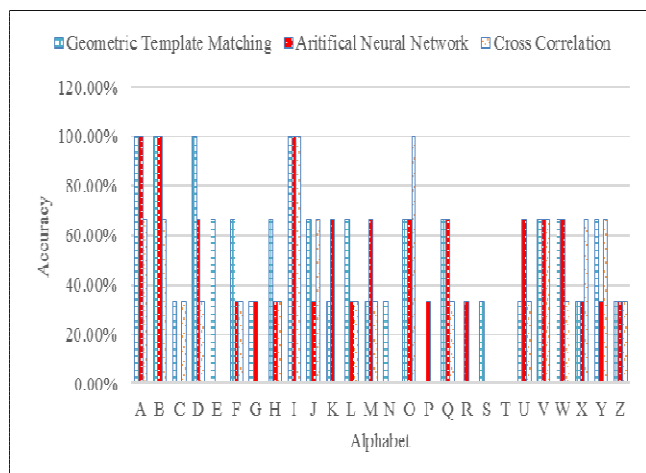


Fig. 11 Comparison of recognition results for Geometric Template Matching, Artificial Neural Network and Cross Correlation

IV. CONCLUSION

This study proposed a prototype that can convert sign language into text by using a sensor device called Leap Motion. The proposed SLTC has a huge potential in enabling better communication between deaf and hearing people as Leap Motion is capable of reading hand and finger motions as inputs. With tracking technology, it has become possible for SLTC to recognise complex hand gestures with substantial accuracy. One of the challenges in using Leap Motion is in creating custom gesture with Leap Motion API. Leap Motion should allow better ways to add custom gesture rather than by writing custom code for each gesture.

Besides that, it is found that the power consumption of Leap Motion is fairly high. In order to make sure Leap Motion can respond instantly and quickly without any lag, it is important to include an algorithm that will eliminate the extra load on both Leap Motion and application, which will provide users with a better and seamless experience. For instance, an algorithm that can detect when Leap Motion is idle and active. Moreover, in order to improve the accuracy, better gesture recognition techniques are required not only to recognise static gestures but also for dynamic gestures. Therefore, for future improvement, better ways of capturing tracking data, creating custom gestures and recognising gestures are needed for Leap Motion in building an effective SLTC.

REFERENCES

- [1] S. D. Emmett and K.P. West, Gestational vitamin A deficiency: A novel cause of sensorineural hearing loss in the developing world?, *Medical Hypotheses*. 82(1) (2014) 6–10.
- [2] L. E. Potter, J. Araullo and L. Carter. The Leap Motion controller: a view on sign language. in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, ACM (Adelaide, Australia, 2013), pp. 175–178.
- [3] S. Mitra and T. Acharya, Gesture Recognition: A Survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 37(3) (2007) 311–324.
- [4] M. W. Kadous. Machine Recognition of Auslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language. in *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, (Wilmington, DE, USA, 1996), pp. 165–174.
- [5] P. Vamplew and A. Adams, Recognition of sign language gestures using neural networks, *Australian Journal of Intelligent Information Processing Systems*. 5 (1998) 94–102.
- [6] E. J. Holden and R. Owens, Visual Sign Language Recognition, in *Multi-Image Analysis: 10th International Workshop on Theoretical Foundations of Computer Vision Dagstuhl Castle, Germany, March 12–17, 2000 Revised Papers* (Springer Berlin Heidelberg 2001), pp. 270–287.
- [7] T. E. Starner and A. Pentland, Visual Recognition of American Sign Language Using Hidden Markov Models, in *Dept. of Architecture. Program in Media Arts and Sciences* (Massachusetts Institute of Technology, 1995).
- [8] N. Tanibata, N. Shimada and Y. Shirai. Extraction of Hand Features for Recognition of Sign Language Words. in *International Conference on Vision Interface*, 2002), pp. 391–398.
- [9] Signtel Inc, *Products*. 2013 [cited 2015 10 June]; Available from: <http://www.signtelinc.com/products.html>.
- [10] iCommunicator, *Product Information*. [cited 2015 15 June]; Available from: <http://www.icommunicator.com/productinfo/>
- [11] F. Weichert, et al., Analysis of the Accuracy and Robustness of the Leap Motion Controller, *Sensors*. 13(5) (2013) 6380–6393.
- [12] atchtmind, *Leap Motion*. 2015; Available from: <http://techtmeindia.com/modern-technology/leap-motion/>

- [13] Leap Motion Developer, *System Architecture*. 2015 [cited 2014 16 June]; Available from: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Architecture.html
- [14] O. L. Robert, *LeapTrainer.js v0.31*. 2013 [cited 2015 12 May]; Available from: <https://github.com/roboleary/LeapTrainer.js/tree/master>
- [15] P. Bourke, *Cross Correlation*. [cited 1996 August 20]; Available from: <http://paulbourke.net/miscellaneous/correlate/>
- [16] OpenCV, *Template Matching*. [cited 2016 April 15]; Available from: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html.
- [17] R. D. Vatavu, L. Anthony and J.O. Wobbrock. Gestures as point clouds: a \mathcal{P} recognizer for user interface prototypes. in *Proceedings of the 14th ACM international conference on Multimodal interaction*, ACM (Santa Monica, California, USA, 2012), pp. 273-280.
- [18] A. Stepanov, *Enable Talk*. 2012 [cited 2015 February 5]; Available from: <http://enabletalk.com/>