Emerging Advances in Integrated Technology Vol.4 No. 1 (2023) 1-6

EmAIT

http://publisher.uthm.edu.my/ojs/index.php/emait
e-ISSN : 2773-5540

Emerging Advances in Integrated Technology

Application of Machine Learning in Malaysia Sign Language Translation

Muhammad Nazmi Ramli[1]*, S. K. Yee[2]

[1]Faculty of Electrical and Electronic Engineering,
University Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

[2]Research Center for Applied Electromagnetic, Institute of Integrated Engineering,
University Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/emait.2023.04.01.001
Received 15 June 2023; Accepted 20 July 2023; Available online 31 July 2023

Abstract: Sign language serves as a vital form of communication for individuals with hearing impairments, enabling seamless interaction among those who cannot hear. It is a widely used means of communication worldwide, facilitating communication within the deaf community. In Malaysia, Malaysian Sign Language (MySL) prevails as the primary sign language employed by the deaf community. However, sign languages possess unique grammatical rules and structures, making them unfamiliar to hearing individuals, leading to potential misunderstandings and communication barriers. This project aimed at developing a sign language translator capable of translating 24 alphabets based on hand gestures. The system employs a dataset of sign language alphabet images, gathered and trained using the Teachable Machine. To evaluate the translator's performance, response time is thoroughly analyzed. The results indicate that 18 out of the 24 alphabets can be recognized within 5 seconds, displaying promising accuracy. By bridging the communication gap between deaf and hearing individuals, the findings of this study hold substantial potential to enhance interactions and foster better understanding between these two communities. The sign language translator represents a significant step towards inclusive communication and improved accessibility for individuals with hearing impairments.

Keywords: OpenCV, Malaysian sign language, pycharm, machine language

1. Introduction

Deaf people can be defined as people with listening problems, those who are considered normal people who cannot hear as a whole, or those who can normally hear below the hearing threshold of 20 dB or better than both. Most people who have deafness usually face difficulties in communicating, especially between them and normal people [1]. Receiver-in-canal (RIC) and receiver-in-the-ear (RITE) are devices that help people with hearing loss to hear what people are saying [2]. However, just listening does not make them with hearing problems easy to communicate. Therefore, a language using hand signals is used for those who understand to communicate with each other.

Sign language is a commonly used language for those with hearing impairments and this language is used worldwide to help those who cannot hear anything to communicate with each other. Many different sign patterns are used for sign language according to each country. As an example, in the United Kingdom, there is a format for sign language that practice for deaf people there, and it is called British Sign Language (BSL) [3]. BSL combines hand gestures, facial expressions, and body language and, like English, has its grammar, syntax, and lexicons.

In Malaysia, MySL (stand for Malaysian Sign Language) is a language format used in speech for deaf people in Malaysia [4]. MySL has many different dialects from one region to another. MySL symbolizes the identity of deaf people in Malaysia who are multi-ethnic, religious and cultural. This is a local sign used by the deaf community then and now.

*Corresponding author: de190204@student.uthm.edu.my

Not only that, utilizing MySL can be a tool that can help deaf people master the Malay language. Unlike British Sign Language (BSL), it only uses one hand at a time to show the alphabet. The hand gesture for 26 Malaysian sign language alphabets is shown in Fig. 1 - Malaysian sign language alphabetFig. 1.

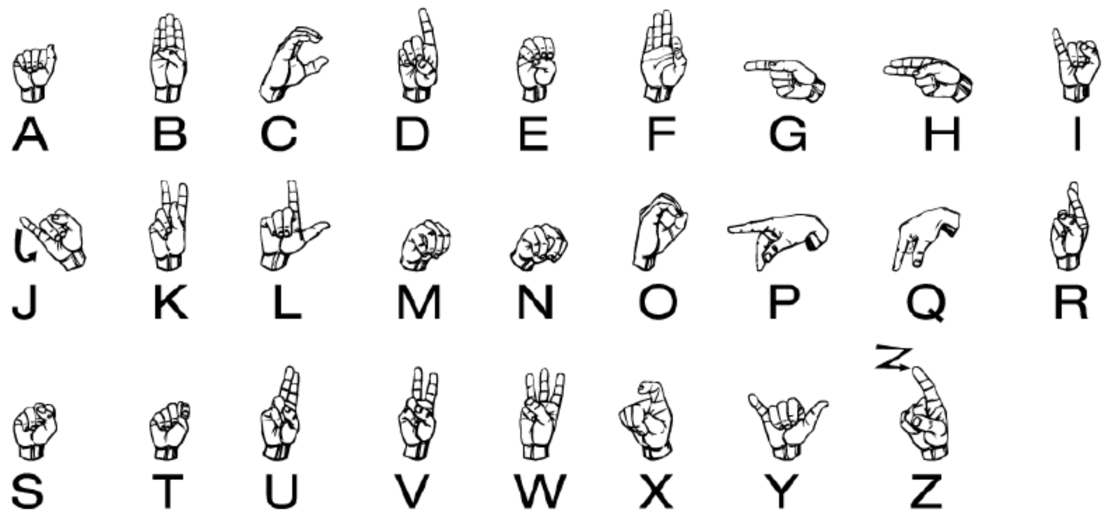

Fig. 1 - Malaysian sign language alphabet

Huge problems occur when normal people are unaware of sign language, making it difficult for them to understand what the deaf individuals are communicating. Not only that, just because of the disadvantages of miscommunication between deaf people and normal people, it can cause much confusion among them and somehow, the discrimination situation might appear. For this project, a system called Sign Language Detector and Translator uses to ease the user in translating hand gesture that shows Malaysia Sign Language (MySL) into the alphabet without the need to learn the sign language through proper education or referring it through books or Google when they are communicating with deaf people.

This project focuses on how the translator detects 24 alphabets; from the beginning, the camera detects the hand gesture until the alphabet is shown on display, including collecting images as dataset and data augmentation techniques. Two alphabets, 'J' and 'Z' are excluded since both alphabets require both hand gestures and motion. They will be considered in future work that requires video collection.

2. Methodology

The methodology involves developing a machine learning system to collect a dataset of 24 alphabet images from sign language gestures captured by a camera. It includes processes for translating hand gestures into alphabets, capturing images for dataset collection, and data augmentation to improve accuracy in reading and translating hand gestures.

The main process started with the hand gesture shown in front of the camera. As an input device, the camera captures all the hand movements and registers the data in the memory. The computer then converts it from the hand gesture image into the alphabet that is already assigned in the dataset. After the conversion, the interface will show the alphabet assigned by the desired hand gesture on the screen monitor as an output. Then, the process continues so that even if the user shows a different hand gesture, it will come out with a different alphabet, referring to the Malaysia Sign Language (MySL).

The data must be stored and initialized before the system reads the hand gesture the user shows to translate it into the alphabet. The data collected from image capture is stored as a dataset in the memory for all 24 alphabets, from A to Y. One of the coding lines is an input that needs to determine the alphabet, which means all the alphabet will be determined into an address. When the code is running, the user must demonstrate the sign language in front of the camera to capture the image of the hand gesture.

For recognizing the image, the camera starts recognizing the hand that shows on the camera once the program is started. Then, when a sign has been shown, the machine will automatically recognize and show the desired output as the sign language has successfully translated.

2.1 Capturing Image and Hand Gesture Recognition

Based on the observation, it is found that some of the signs have similarities, such as sign for alphabet 'D' and alphabet 'G' has the same sign but different rotation, alphabet 'M' and alphabet 'N' which nearly have the same image but different thumb finger position, alphabet 'A', 'E', 'S' and 'T' also have the slightly similar image where the position of the thumb finger that differentiate them, and the sign for alphabet ‘K’ and ‘V’ also have a peace looking sign that can make the machine confusing to recognize both signs.

PyCharm, which uses Python languages, captures and stores all 24 images of alphabets. The code for capturing the image starts by installing both library package files: *cvzone* and *mediapipe*. Installing *cvzone* is for applying computer vision as it reads and captures images [5], while *mediapipe* generates a real-time perception pipeline, which is very useful for reading finger gestures and hand tracking [6].

Certain considerations are made for capturing the images. The resolution is set to 300x300 pixels to standardize the images and aid machine recognition. Each sign has different patterns, sizes, lengths, and heights. For example, the sign 'B' has more height than length, while 'Y' has more length than height. A code is created to resize the images to 300x300 pixels while maintaining the 1:1 aspect ratio to ensure consistent sizing. A folder is created for storing the dataset images, with 24 folders corresponding to each alphabet.

Once running, the camera turns on, displaying a real-time video in a new window. The appearance of the pipeline image indicates successful code execution, allowing for capturing images for all 24 alphabet signs. A button, specifically the 'S' button, triggers capturing images. The image is captured and automatically stored in the designated folder when pressed. The file structure is defined to ensure images are stored correctly. For example, when capturing the sign for the alphabet 'T', the code addresses the 'T' folder. Each captured image is given a unique name or numbering, considering a maximum of 300 images per alphabet to avoid overlapping.

2.2 Machine Teaching and Learning

The trial-and-error method was used for machine training to understand image recognition and achieve desired results. After extensive experimentation and research, a website called Teachable Machine was discovered. Teachable Machine, a Google web service, allows users to train machine learning models without coding knowledge. Its user-friendly interface enables users to teach the computer to recognize and classify items using webcams or uploaded photographs. This accessibility aims to make machine learning available to a wider range of users, even those without programming experience. To teach the machine the 24 hand gestures, the image datasets are uploaded into Teachable Machine and organized into folders based on the assigned alphabets. Once the dataset images are uploaded, the learning process begins. The machine may take a considerable amount of time to learn, analyze, and recognize the uploaded dataset. After completing the learning process, a preview tab displays the results. However, since the project uses PyCharm, the preview can be skipped, and the code is exported as Keras code through Tensorflow. The downloaded code is stored in a new folder named 'Model,' which is used for reading images later.

For hand gesture recognition, a set of coding has been written, consisting of recalling all the dataset by putting labels, and customizing the box interface with font, typeface color, box outcome interface color, etc. When the code is running, the new popout windows called image will come out, same as the data collection coding, but this time, it is used to recognize the hand gesture and produce some outcome. The hand recognition interface is shown in Fig. 2.

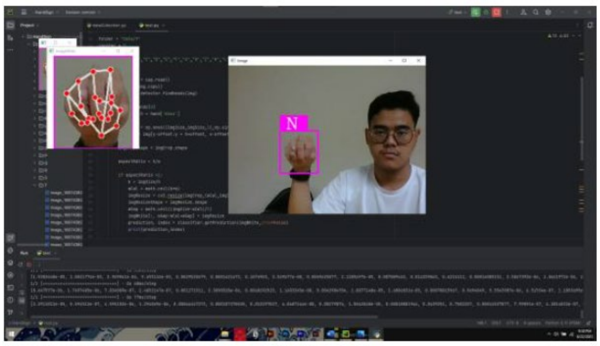

Fig. 2 - Hand recognition interface

3. Results and Discussion

The dataset must first be determined as a reference for the machine to recognize the hand gesture better. All the images need to be captured, around 300 images with a resolution of 300x300 pixels for each alphabet, excluding alphabet ‘J’ and alphabet ‘Z’. Fig. 3 shows all the images that have been captured throughout the process of collecting data.

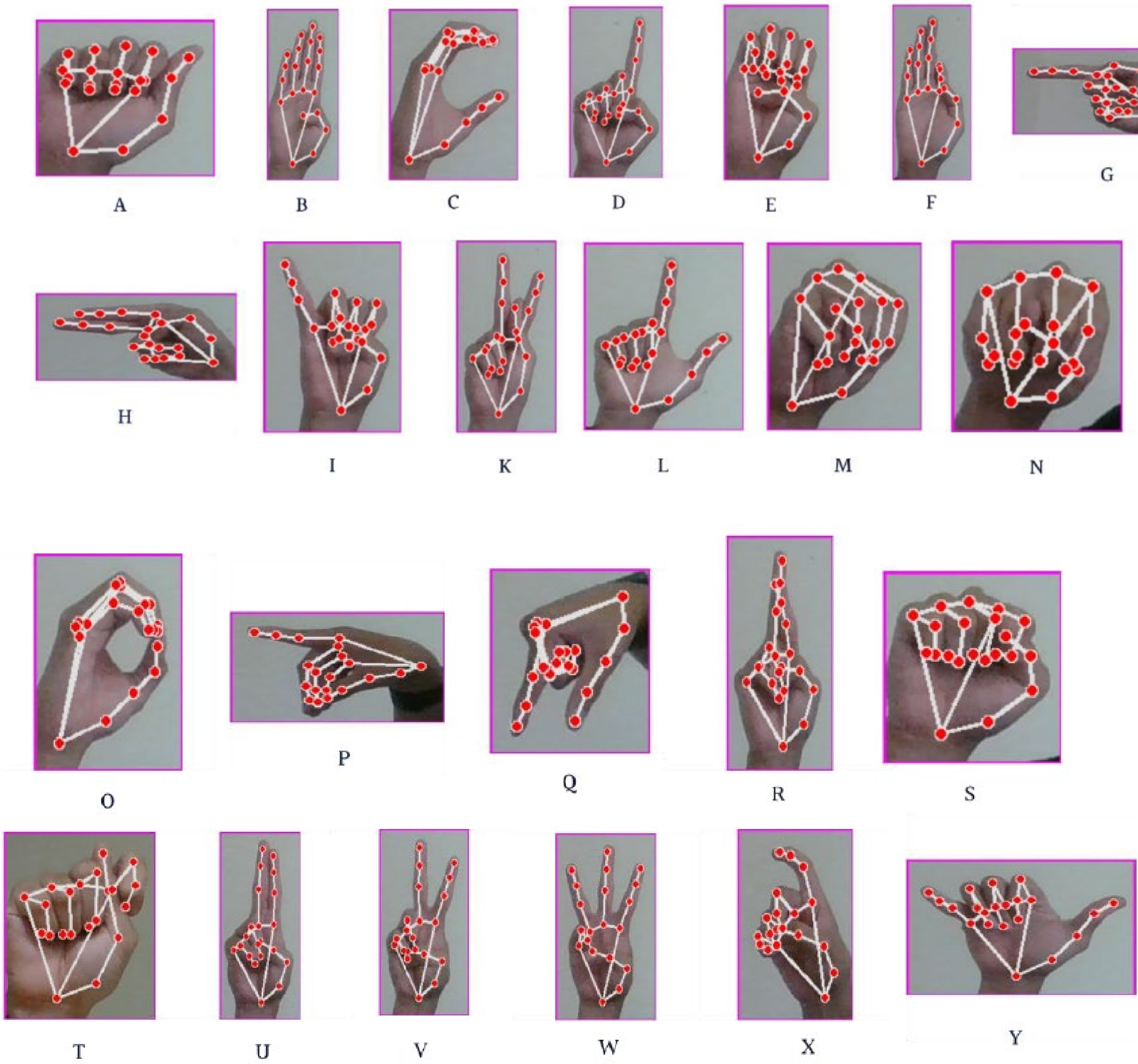

Fig. 3 - Hand gesture image captured including mediapipe for machine learning

3.1 Response Time and Performance Observation

The machine can read all the alphabet correctly since if there is an error or misrecognizing the right alphabet, the dataset will be recaptured and restored, and the machine needs to be trained again. Even though the machine has successfully recognized all 24 alphabets, the response time to show the right alphabet has been observed. From the observation results, it can be concluded that most of the signs with unique hand gestures can easily be recognized and have a better response time. Five hand gestures show the immediate right alphabet. Meanwhile, seven hand gestures have difficulties showing the right alphabet, and it takes time to make sure the machine can read the right alphabet as a result. Table 1 shows the response time results as an overall observation; the immediate response is labeled as good, and the late response (in which up to 5 seconds and above) is categorized as poor.

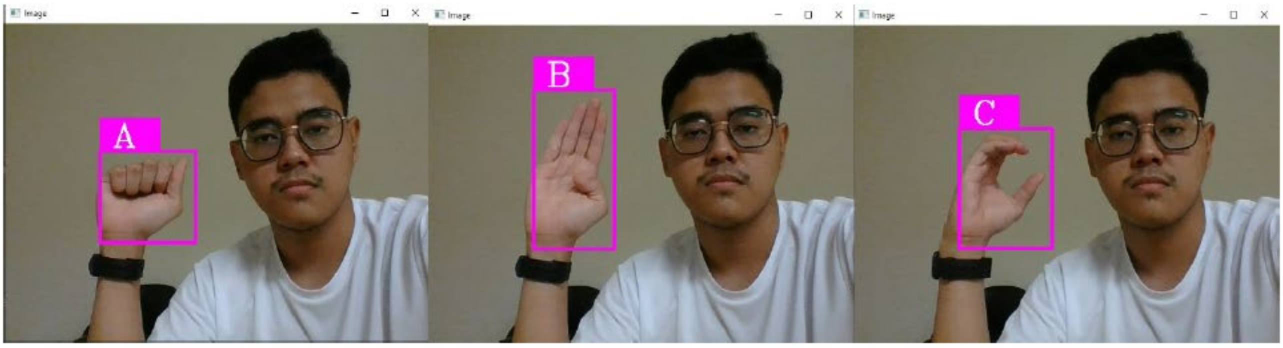

Fig. 4 - All alphabets have been recognized successfully

Table 1 - Response time of each alphabet

Alphabet	Response time (s)	Performance
A	2.4	Good
B	0.5	Good
C	51.1	Good
D	3.8	Good
E	2.8	Good
F	4.8	Good
G	3.9	Good
H	6.1	Poor
I	4.6	Good
K	2.6	Good
L	2.3	Good
M	18.5	Poor
N	11.9	Poor
O	1.1	Good
P	0.8	Good
Q	2.1	Good
R	1.2	Good
S	13.4	Poor
T	8.1	Poor
U	5.3	Poor
V	2.4	Good
W	2.5	Good
X	4.2	Good
Y	3.9	Good

4. Conclusion

This project successfully showcases the feasibility of machine learning in recognizing 24 Malaysia Sign Language gestures, achieved through comprehensive dataset collection, machine training, and accurate output generation. The system's average response time is approximately 6.7 seconds, with the shortest response time being 0.5 seconds and the longest response time reaching 18.5 seconds. It is noteworthy that the longer response time is observed specifically for alphabets 'M' and 'N,' which exhibit significant similarities. The developed sign language recognition system offers several valuable benefits. Firstly, it facilitates improved communication and understanding between deaf and hearing individuals, bridging the gap and fostering inclusivity. Secondly, by automating the recognition process, the system enhances accessibility for the deaf community, empowering them in various aspects of daily life. Furthermore, this technology opens doors to further advancements in assistive devices, interactive learning tools, and communication aids, ultimately contributing to a more inclusive and supportive society for individuals with hearing impairments. In addition to the successful implementation of the sign language recognition system, future work can focus on addressing the challenges associated with recognizing alphabet J and Z, which involve movement. These dynamic signs require an extended scope of analysis and tracking to interpret the gestures and achieve improved recognition accuracy accurately.

Acknowledgement

The authors fully acknowledged the Faculty of Electrical and Electronic Engineering, and the Research Center for Applied Electromagnetic, Institute of Integrated Engineering, University Tun Hussein Onn Malaysiafor supporting this work.

References

[1] R. Kushalnagar, “Deafness and hearing loss,” *Web Access. A Found. Res.*, pp. 35-47, 2019.

[2] J. A. Cook and D. B. Hawkins, “Outcome measurement for patients receiving hearing aid services,” *Laryngoscope*, vol. 117, no. 4, pp. 610-613, 2007.

[3] H. Grote, F. Izagaren, and V. O’Brien, “How to communicate with patients who are D/deaf or have hearing loss,” *bmj*, vol. 373, 2021.

[4] H. C. Omar, *Penterjemahan dan bahasa isyarat*. ITBM, 2009.

[5] I. N. T. A. Putra, K. S. Kartini, Y. K. Suyitno, I. M. Sugiarta, and N. K. E. Puspita, “Penerapan Library Tensorflow, Cvzone, dan Numpy pada Sistem Deteksi Bahasa Isyarat Secara Real Time,” *J. Krisnadana*, vol. 2, no. 3, pp. 412-423, 2023.

[6] S. Salim, M. M. A. Jamil, R. Ambar, R. Roslan, and M. G. Kamardan, “Sign Language Digit Detection with MediaPipe and Machine Learning Algorithm,” in *2022 IEEE 12th International Conference on Control System, Computing and Engineering (ICCSCE)*, 2022, pp. 180-184.