

Development of a PC-based sign language translator

Kamoli Akinwale Amusa, Ayorinde Joseph Olanipekun, Tolulope Christiana Erinosho, Abiodun Akeem Salaam, Sodi Segun Razaq

Department of Electrical and Electronics Engineering, College of Engineering, Federal University of Agriculture, Abeokuta, Nigeria

Article Info

Article history:

Received Apr 24, 2022

Revised Jun 10, 2022

Accepted Aug 15, 2022

Keywords:

Communication aid

Gestures

Hearing-impaired

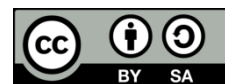
Sign language

Translator

ABSTRACT

While a hearing-impaired individual depends on sign language and gestures, non-hearing-impaired person uses verbal language. Thus, there is need for means of arbitration to forestall situation when a non-hearing-impaired individual who does not understand the sign language wants to communicate with a hearing-impaired person. This paper is concerned with the development of a PC-based sign language translator to facilitate effective communication between hearing-impaired and non-hearing-impaired persons. Database of hand gestures in American sign language (ASL) is created using Python scripts. TensorFlow (TF) is used in the creation of a pipeline configuration model for machine learning of annotated images of gestures in the database with the real time gestures. The implementation is done in Python software environment and it runs on a PC equipped with a web camera to capture real time gestures for comparison and interpretations. The developed sign language translator is able to translate ASL/gestures to written texts along with corresponding audio renderings at an average duration of about one second. In addition, the translator is able to match real time gestures with the equivalent gesture images stored in the database even at 44% similarity.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Kamoli Akinwale Amusa

Department of Electrical and Electronics Engineering, College of Engineering

Federal University of Agriculture

Alabata Road, Off Abeokuta-Ibadan Road, Abeokuta, Nigeria

Email: amusaka@funaab.edu.ng

1. INTRODUCTION

Hearing-impaired is a generally acceptable term used to describe someone who has a hearing loss. It could refer to someone who is born deaf or became deaf later in life. Hearing-impaired individuals are faced with challenges of communication, especially with non-hearing-impaired individuals, as the basic ways of communication for them is through sign language, body language, and facial gestures. A major consideration in communication between hearing-impaired and non-hearing-impaired counterparts bothers on how to aid, facilitate and enhance effective communications between them. Over 5% of the world's population is hearing impaired and it is estimated that by 2050, over 700 million people, or one in every ten people will have a disability of hearing [1]. About 23.7% of Nigerians are estimated to have hearing impairment [2], which translates to about 24 persons in every 100 of the total population. Communication between hearing-impaired and non-hearing-impaired persons can be considered a more serious problem compared to communication with visually-impaired people. While non-hearing-impaired individuals communicate freely via verbal languages, and listen to audio messages. A non-verbal communication method constitutes the main mode of sharing information amongst hearing-impaired people.

Hearing-impaired persons communicate mainly through sign language (SL), which includes the use of facial gestures, certain body part movements, and lip-reading. SL is country-specific which makes its translation a difficult task [3]. The most important thing is getting the attention of the hearing-impaired, which can be done via several ways such as touching the arm or shoulder, waving, stamping on the floor, and switching the light on and off [4]. The choice and effectiveness of a method in catching the attention of the hearing-impaired person depend on how well the two parties in the communication process are familiar.

Going by the number of hearing-impaired people around us, their inability to convey their thought constitutes a major problem, especially in areas where they exist in minority. Effective communication is required for peaceful co-existence and the general well-being of the society at large. In view of this, SLs were developed to facilitate effective communication between hearing-impaired persons. To a large extent, this measure has addressed the problem of communication among educated hearing-impaired individuals. However, what about communication between hearing-impaired and non-hearing-impaired persons? Hence, there is a need for an arbiter between these two sets of people in order to forestall avoidable crises that may emanate from communication gap. Essentially, two key issues are involved in arriving at such a desired arbiter: sign language recognition (SLR) and sign language translation (SLT).

Research into recognizing and interpreting SL gestures started a few years back. SLR schemes can be grouped into three, depending on the form it takes. They are: SLR schemes that rely on the recognition of finger-spellings, those that are based on the recognition of isolated words, and lastly, continuous sentence construction recognition based SLR [5]. Many of the earlier research efforts into the SLR scheme adopted traditional recognition approaches which include the use of hidden Markov model [6] for words recognition, support vector machine [7], [8] for classification of both isolated words and continuous SL alphabets, and trajectory matching for isolated words grouping. Recently, different varieties of deep-learning methods convolutional neural network (CNN) [9], [10], long short-term memory (LSTM) [11]–[14] are being utilized singly and in hybrid configurations [15], [16] to address the problem of SLR, especially in applications involving the recognition of continuous sentence structure.

Unlike the situation with SLR, reports on SLT are scanty in the literature. However, knowledge gained from the research into SLR provides ample leverage in SLT development to facilitate effective communication between the hearing-impaired and non-hearing-impaired persons in our society. To that end, few proposals on SLT that can be found in the literature include deep learning model-based SLT [11], [15], [17]–[20] and sensor-based SLT [21]–[35]. Majority of the neural SLT models adopt a multimodal structure in their construction such that sequential connections exist between CNN and neural machine translation (NMT). While the NMT module is essentially the kernel for the translation of SL gestures into target sentences, CNN is used for the extraction of image-level features that serve as NMT input. The critical problem, about all deep learning-based SLT models, is the requirement for a large dataset, which is not readily available. This requirement hinders the performance of resulting SLT models. Sensor-based SLTs are usually built around a glove incorporated with microtouch switch or other electronic devices for gesture recognition and translation into text and speech equivalents.

Other researchers developed different SLTs such as software-based platform [36], MobilenetV2-based gestures recognition system [37], and tablet-based hearing aid [38]. An open-source software framework developed in [36] presents a development environment for building of augmentative and alternative communication models that include communication aids for the disabled community. A MobilenetV2-based gesture recognition system was developed in [37]. Although, the system is specifically meant for smart home applications, it can as well be deployed for gestures translation. Cameras of mobile devices are used for detection and capturing of data from objects with the image presented in the frame by a bounding box. The focus of the developed device in [38] was a tablet-based hearing aid where sign language gestures are digitally processed by the tablet before being wirelessly relayed to standard earphones for better output.

This work proposes a PC-based communication aid to facilitate effective communication between hearing-impaired and non-hearing-impaired people. American sign language (ASL) gestures are employed in the development whereby a database of hand gestures in ASL is created using Python scripts while the pipeline configuration model for machine learning of annotated images of gestures in the database with the real-time gestures is realized via the use of TensorFlow (TF). The developed SLT running on a PC equipped with a web camera that captures real-time gestures for comparison and interpretations is implemented in Python software environment. Outputs of the developed SLT are translation of ASL/gestures into written texts and corresponding audio renderings at an average duration of about one second. The novelty of this paper is premised on the non-introduction of new device, which leads to reduced cost in the longrun. Personal computer system is a veritable and common tool nowadays, that is available to different individuals (hearing-impaired and non-hearing impaired inclusive) mingling and interacting together. While sensor-based SLTs SLT [21], [23]–[33], [35] tend to be costlier, the proposed PC-based SLT in this paper is

relatively cheaper as no new separate gadgets are required. In addition, no need for additional training as the device can be operated by whoever can handle a computer system. Specifically, a database of ASL gesture is created using Python script, which enables faster further signal processing, and the development of TF pipeline configuration model to interface with the created database for machine learning.

The rest of the paper is structured as follows. Presented in section 2 is the method while section 3 has results and discussion. The paper is concluded in section 4.

2. METHOD

The developed communication aid between hearing and non-hearing-impaired individuals is achieved by converting ASL gestured by the hearing impaired to a corresponding text and audio signals for the non-hearing impaired to interpret. The stages involved in the development are basically three: i) creation of database or datasets of images for each of the selected ASL gestures using a Python script; ii) creation of a pipeline configuration model using TF software to interface with the created database for machine learning; and iii) deployment of the TF pipeline configuration model in a Python software environment for matching, comparison, and decision making with real-time images of ASL gestures and production of corresponding texts and audio words. In order to realize the above steps, a few materials (hardware and software) are required. They are briefly highlighted in what follows, beginning with hardware materials.

2.1. Hardware materials

Hardware materials used in this study include: i) a 4.00 gigabyte (GB) random access memory (RAM), 2.60 gigahertz (GHz) processor, 64-bit, laptop running Windows 10 operating system-the laptop is used as the workbench for the developed communication aid and it housed the software component of the developed communication aid for deployment and utilization; ii) a 1,080 P video recording 12.0 M pixel high definition (HD) webcam-this is used to capture real-time images of the sign language and gestures before it. The webcam is connected to the laptop through its universal serial bus (USB) port and images captured by the webcam are saved to the laptop; iii) a SanDisk 8 GB Micro secure digital (SD) memory card: the memory card is used to compile the selected images of the gestures or sign languages. It will be slotted into the SD port of the laptop to extract the images; iv) other hardware like speakers is provided as peripheral to the computer to enhance audio output rendering of the gesture images; and v) sign language gesture—specifically, ASL gesture images are employed in the development of the communication aid reported in this paper.

2.2. Software components

Software components used in this study include: i) Python software v3.8, a popular open-source software and programming language. It is used as a model building and deployment environment for the developed communication aid. The choice is informed by its versatility and ease of use for different scripting applications in a wide variety of domains; ii) Labeling, a graphical image annotation tool for labeling object bounding boxes in images. It is written in Python for the creation of bounding box annotations of gesture images. The created annotations are saved in CreateML formats; iii) Pyttsx3, adopted a text-to-speech conversion library for this work. It is a library in Python and is chosen because it works offline unlike available alternative libraries that work mainly online; iv) TF, a software library or framework, designed by the Google team, for easier and faster implementation of machine learning and deep learning concepts. Core functionalities of TF that favour its choice for this work are: augmented tensor operations with seamless interfaces with existing programs, automatic differentiation, which occupies the very core of optimization-based algorithms, and parallel and distributed (multi-machine) computing. TF is used in the creation of a pipeline configuration model (an application interface) to facilitate easier detection of objects. The pipeline configuration file is split into five parts: model configuration, train configuration, evaluation configuration, train input configuration, and evaluation input configuration; v) Mobilenet_v2 SSD, an object detector that can be used on real-time images for location finding. Detected points/locations are described by bounding boxes with each of the bounding boxes assigned a class; vi) Jupyter server, an extension in Python software environment that extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. The Jupyter server combines two components: a web application and notebook documents. For this work, the web application component is used; and vii) deepstack server, an artificial intelligence (AI) server that enables development of faster AI systems both on premise and in the cloud. DeepStack runs on the docker platform but can be used from any programming language.

2.3. Procedure

Having highlighted the needed materials (both hardware and software), the next undertaking is the description of the procedure involved in the development of the developed communication aid. A Python script is written to compile and create the database for ASL gestures. The Python written script uses OpenCV, a library in Python for computer vision and imaging, to facilitate interaction between the laptop and webcam. This interaction allows storage of snapshots of ASL gestures made by the webcam for further processing. The images of some of the ASL gestures used and their corresponding meanings are shown in Figure 1 while Figure 2 depicts the snapshot of the ASL database created. Figure 1 describe the typical ASL gestures used in the creation of the database with corresponding meanings (a) hello, (b) I love you, (c) nice to meet you, (d) no, (e) please, and (f) sorry.

As describe earlier, Python Labellmg is then used for annotation of objects in the created ASL database. Figure 3 illustrates a typical form of Labellmg annotation, specifically for the gesture image “hello”. Building of pipeline configuration model for machine learning was realized via the use of TF. The pipeline is used to convert annotations and the created ASL database into TF record format for machine learning. For object detection, Mobilenet_v2_SSD is used in conjunction with the created TF pipeline. Figure 4 shows extract from the TF pipeline created.

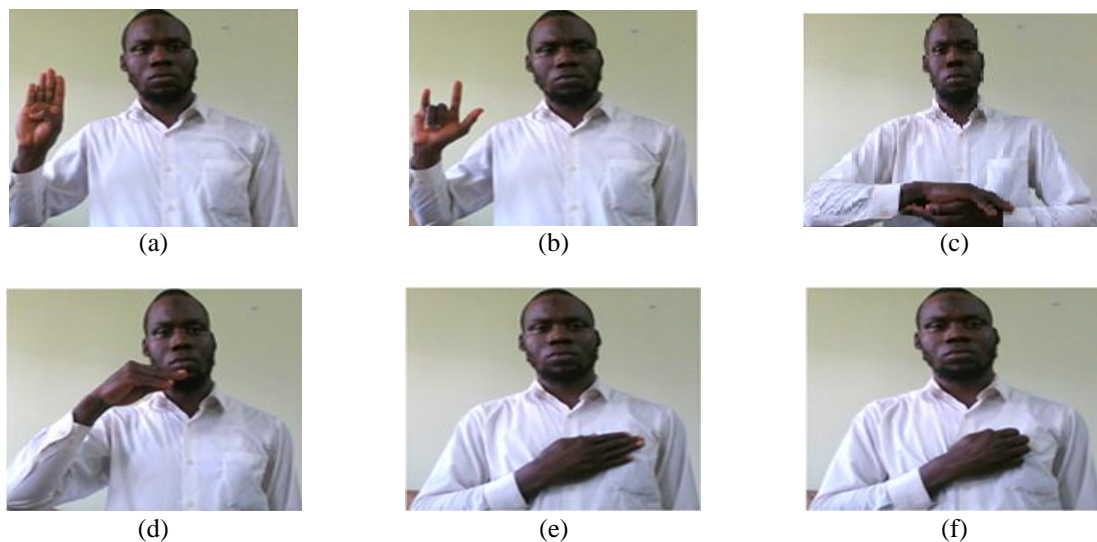


Figure 1. Typical ASL gestures used in the creation of the database with corresponding meanings (a) hello, (b) I love you, (c) nice to meet you, (d) no, (e) please, and (f) sorry

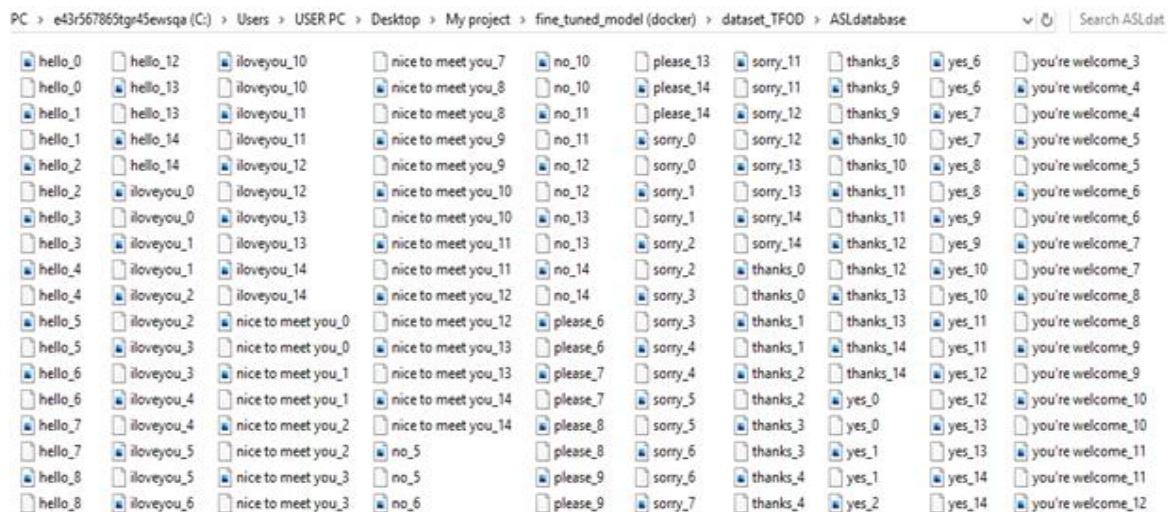


Figure 2. The snapshot of the ASL gesture images database created

```

<annotation>
  <folder>dataset</folder>
  <filename>hello_0.jpg</filename>
  <path>/home/lalah/Downloads/Sign_Language_Dataset/dataset_TFOD/dataset_TFOD/dataset/hello_0.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>640</width>
    <height>480</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>hello</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>110</xmin>
      <ymin>159</ymin>
      <xmax>207</xmax>
      <ymax>340</ymax>
    </bndbox>
  </object>
</annotation>

```

Figure 3. Snapshot of the Python labelling for the annotated gesture image “hello” in the database



```

1 model {
2   ssd {
3     num_classes: 9
4     image_resizer {
5       fixed_shape_resizer {
6         height: 320
7         width: 320
8       }
9     }
10  }
11 }

```

Figure 4. Extract from the created TF pipeline configuration model

The TF pipeline configuration model is trained on the created database and its annotations in Python software environment. The model is first deployed on Jupyter server, then on deepstack server. The deployment on Jupyter and deepstack run on port 80 and simply points to the saved model's directory. The model compares, match and make final decisions on the real time images of ASL displayed by the gesturer. It also displays the boundary box coordinates which indicates the precision range from 0 to 1. An evaluation script that uses OpenCV facilitates interaction with the webcam while Pyttsx3 sees to conversion of the translated sign gestures text equivalents to the corresponding audio in real-time.

3. RESULTS AND DISCUSSION

Findings from deployment of the developed SLT are presented here as well as its response time when real time gestures are positioned before a PC running the developed communication aid. To that end, the discussion is grouped into two, beginning with the process of engaging the developed SLT.

3.1. Engaging the developed SLT

A gesturer is positioned facing the webcam of the PC where the developed virtual communication aid is running. This line of instruction is entered at command prompt on the PC: *deepstack--MODELSTORE-DETECTION"C:\Users\USERPC\Desktop\ Myproject\fine_tuned_model (docker)\ObjectDetection\models"--PORT 80*. To enable the webcam to start the video streaming, the above line of instruction is followed by: *cd "C:\Users\USER PC\Desktop\My project\fine_tuned_model (docker)\ObjectDetection"* and *python livefeed_detection.py*.

Successful operation and entering of correct instructions lead to the display of the properties of the real time image in front of the webcam and matching of the image with the equivalent in the database to give appropriate outputs (text displayed and audio rendering). In addition, a text is displayed on the PC screen to indicate the accuracy of the matching process. Pressing of the letter “Q” on the keyboard halt the live feed. Figure 5 shows a snapshot of the first and second line of instructions when entered at command prompt.


```

Microsoft Windows [Version 10.0.17763.1577]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\USER PC>deepstack --MODELSTORE-DETECTION "C:\Users\USER PC\Desktop\My project\fine_tuned_model (docker)\ObjectDetection\models" --PORT 80
DeepStack: Version 2021.02.1
-----
v1/vision/custom/sign
-----
v1/backup
-----
v1/restore
-----
C:\Users\USER PC>cd "C:\Users\USER PC\Desktop\My project\fine_tuned_model (docker)\ObjectDetection"
C:\Users\USER PC\Desktop\My project\fine_tuned_model (docker)\ObjectDetection>python livefeed_detection.py
[INFO] starting video stream...
_

```

Figure 5. Snapshot of the first and second instructions entered at command prompt to initiate detection and streaming of the gesturer in the front of the PC camera

Once the two instructions have been entered at the command prompt, real time image of gesturer in front of the camera is processed accordingly with the equivalent interpretation of the sign language translated into appropriate outputs (text displayed and audio). Figure 6 portrays obtained results when nine gestures are presented in front of the PC running the developed SLT. As can be observed from Figure 6, in addition to displayed text equivalent rendering of ASL gestures, certain numerical values, which range between 0 and 1 are shown. Those numerical figures indicate precision figure associated with each translated ASL gesture. Figure 6 describe the matching and translation of some ASL gestures using the developed PC-based sign language translator (a) yes, (b) please, (c) nice to meet you, (d) hello, (e) I love you, (f) you are welcome, (g) thanks, (h) sorry, and (i) no.

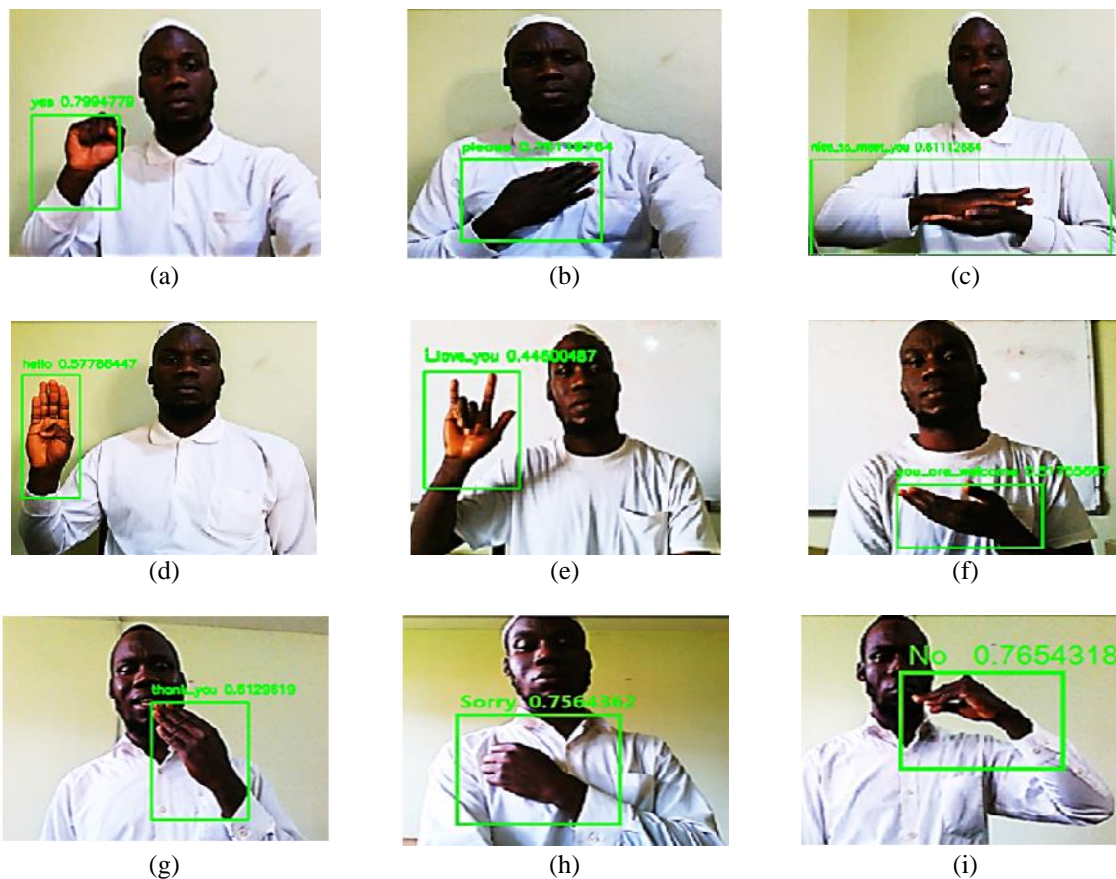


Figure 6. Matching and translation of some ASL gestures using the developed PC-based sign language translator (a) yes, (b) please, (c) nice to meet you, (d) hello, (e) I love you, (f) you are welcome, (g) thanks, (h) sorry, and (i) no

3.2. Time elapse for inference making

Table 1 presents the time taken by the developed PC-based based SLT to render selected ASL gestures presented in front of PC where it is deployed, into corresponding text and audio interpretations. In addition, translation precision values as captured for each of the selected ASL gestures are also presented. It is obvious from the results presented in Table 1 that the developed PC-based sign language translator achieves its aim. Going by the outputs illustrated, each of the nine ASL gestures used for the evaluation is successfully translated into word equivalent within reasonable time. This indicates that the developed sign language translator is suitable as a communication aid between hearing-impaired and non-hearing-impaired individuals.

Table 1. Duration of time for the inference on prediction of gestures and translation precision

Interpreted gestures	Duration of time for inference on prediction (Seconds)			Translation precision (%)
	Minimum	Maximum	Average	
Yes	0.7594	1.1143	0.9369	79.9578
Please	0.7325	1.3072	1.0199	75.1198
Nice to meet you	0.7945	1.2642	1.0294	51.1127
Hello	0.7725	1.2792	1.0259	57.7864
I love you	0.8685	1.3692	1.1189	44.6005
You are welcome	0.9114	1.4731	1.1923	51.7557
Thank you	0.8795	1.2492	1.0644	61.2952
Sorry	0.8885	1.2392	1.0639	75.6436
No	0.8885	1.2392	1.0639	76.5432

The duration for processing of each of the ASL gestures used in the evaluation is generally about one second, which shows that the translation is done in real-time without delay which may introduce frustration in the communication process if a longer time is involved. The precision value attached to each gesture indicates how accurate the real-time images are matched with those in the annotated database. The least in the set of ASL gestures used for the evaluation of the developed sign language translator is about 44% (corresponding to “I love you”) while the highest is approximately 80% (corresponding to “yes”). These results show how robust the developed sign language translator is, for it detect and correctly translate gestures into required outputs (text and audio rendering) when the matching between the real image and real image is about 44%.

It is worth pointing out here that the proposed PC-based SLT in this paper takes lesser time to respond when compare with the response time of most sensor-based SLTs where few seconds are needed for recognition. For instance, it is reported in [28] that the detection of hand motions took a few seconds as the user had to hold a formed sign for two seconds to ensure recognition. This is in addition to the lower cost of implementation, acquisition, and operation of the proposed SLT since it is simply an ‘add-in’ to a typical PC. Table 2 further summarizes the comparison of PC-based SLT proposed in this paper with two others that can be found in the literature.

Table 2. Comparison of different SLTs performances

Parameters	SLTs		
	Sensor and microcontroller-based	Neural-based	Proposed PC-based
Cost	Moderate cost	Moderate	Relatively cheap as no need for other gadgets
Operation	Require training for operation	Required hands-on training	No special training besides basic computer operation
Response time	Few seconds required	Longer time	Average of one-second response time
Database	Limited owing to the size of the microcontroller memory space	Large	Large

4. CONCLUSION

A PC-based sign language translator has been developed in this paper. The developed SLT is shown to successfully translate nine different ASL sign gestures into text and audio equivalents. These written and audio interpretations rendering of sign language gestures will go a long way in facilitating effective communication between hearing-impaired and non-hearing-impaired individuals in our society when deployed. Hence, it provides a method of addressing the problem of communication between these individuals and aid their mutual interactions in daily activities.

REFERENCES




- [1] World Health Organization, “Deafness and hearing loss,” *All Ears International/G. Vaughan.*, 2021. <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss> (accessed Jan. 16, 2022).

- [2] E. I. Asonye, E. Emma-Asonye, and M. Edward, "Deaf in Nigeria: A preliminary survey of isolated deaf communities," *SAGE Open*, vol. 8, no. 2, pp. 1–11, Apr. 2018, doi: 10.1177/2158244018786538.
- [3] R. Brooks, "A guide to the different types of sign language around the World," *K-International*, 2018. <https://www.k-international.com/blog/different-types-of-sign-language-around-the-world/> (accessed Jan. 16, 2022).
- [4] M. J. Napier, J. Fitzgerald, and E. Pacquette, *British sign language for dummies*. Chichester: John Wiley & Sons, 2008.
- [5] J. Zheng *et al.*, "An improved sign language translation model with explainable adaptations for processing long sign sentences," *Computational Intelligence and Neuroscience*, pp. 1–11, Oct. 2020, doi: 10.1155/2020/8816125.
- [6] W. Gao, G. Fang, D. Zhao, and Y. Chen, "A Chinese sign language recognition system based on SOFM/SRN/HMM," *Pattern Recognition*, vol. 37, no. 12, pp. 2389–2402, 2004, doi: 10.1016/S0031-3203(04)00165-7.
- [7] M. C. Hu, H. T. Liu, J. W. Li, C. W. Yeh, T. Y. Pan, and L. Y. Lo, "Sign language recognition in complex background scene based on adaptive skin colour modelling and support vector machine," *International Journal of Big Data Intelligence*, vol. 5, no. 1–2, pp. 21–30, 2018, doi: 10.1504/IJBDI.2018.10008140.
- [8] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen, "Isolated sign language recognition with grassmann covariance matrices," *ACM Transactions on Accessible Computing*, vol. 8, no. 4, pp. 1–21, May 2016, doi: 10.1145/2897735.
- [9] Z. Liang, S. Liao, and B. Hu, "3D convolutional neural networks for dynamic sign language recognition," *The Computer Journal*, vol. 61, no. 11, pp. 1724–1736, Nov. 2018, doi: 10.1093/comjnl/bxy049.
- [10] S. Yang and Q. Zhu, "Video-based Chinese sign language recognition using convolutional neural network," in *2017 IEEE 9th International Conference on Communication Software and Networks*, May 2017, pp. 929–934, doi: 10.1109/ICCSN.2017.8230247.
- [11] D. Guo, W. Zhou, H. Li, and M. Wang, "Hierarchical LSTM for sign language translation," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018, pp. 6845–6852, doi: 10.1609/aaai.v32i1.12235.
- [12] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 2871–2875, doi: 10.1109/ICIP.2016.7532884.
- [13] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 4160–4169, doi: 10.1109/CVPR.2019.00429.
- [14] S. Yang and Q. Zhu, "Continuous Chinese sign language recognition with CNN-LSTM," in *Ninth International Conference on Digital Image Processing (ICDIP 2017)*, Jul. 2017, pp. 83–89, doi: 10.1117/12.2281671.
- [15] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Apr. 2018, pp. 2257–2264, doi: 10.1609/aaai.v32i1.11903.
- [16] C. Mao, S. Huang, X. Li, and Z. Ye, "Chinese sign language recognition with sequence to sequence learning," in *Communications in Computer and Information Science*, vol. 771, Singapore: Springer, 2017, pp. 180–191, doi: 10.1007/978-981-10-7299-4_15.
- [17] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7784–7793, doi: 10.1109/CVPR.2018.00812.
- [18] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho, "Neural sign language translation based on human key point estimation," *Applied Sciences*, vol. 9, no. 13, pp. 1–19, Jul. 2019, doi: 10.3390/app9132683.
- [19] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream CNN-LSTM-HMMS to discover sequential parallelism in sign language videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2306–2320, Sep. 2020, doi: 10.1109/TPAMI.2019.2911077.
- [20] S. Wang, D. Guo, W. Zhou, Z.-J. Zha, and M. Wang, "Connectionist temporal fusion for sign language translation," in *Proceedings of the 26th ACM international conference on Multimedia*, Oct. 2018, pp. 1483–1491, doi: 10.1145/3240508.3240671.
- [21] A. M. Abdullah, J. K. P. Sarwar, and M. A. Fakir, "Flex sensor based hand glove for deaf and mute people," *International Journal of Computer Networks and Communications Security*, vol. 5, no. 2, pp. 38–48, 2017.
- [22] V. V. Shelke, V. V. Khaire, P. E. Kadlag, and K. T. V. Reddy, "Communication aid for deaf and dumb people," *International Research Journal of Engineering and Technology*, vol. 06, no. 11, pp. 1930–1933, 2019.
- [23] J. Wu, L. Sun, and R. Jafari, "A wearable system for recognizing American sign language in real time using IMU and surface EMG sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 5, pp. 1281–1290, 2016, doi: 10.1109/JBHI.2016.2598302.
- [24] K.-W. Kim, M.-S. Lee, B.-R. Soon, M.-H. Ryu, and J.-N. Kim, "Recognition of sign language with an inertial sensor-based data glove," *Technology and Health Care*, vol. 24, no. 1, pp. 223–230, Dec. 2015, doi: 10.3233/THC-151078.
- [25] J. Galka, M. Masior, M. Zaborski, and K. Barczewska, "Inertial motion sensing glove for sign language gesture acquisition and recognition," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6310–6316, Aug. 2016, doi: 10.1109/JSEN.2016.2583542.
- [26] D. Lu, Y. Yu, and H. Liu, "Gesture recognition using data glove: An extreme learning machine method," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2016, pp. 1349–1354, doi: 10.1109/ROBIO.2016.7866514.
- [27] K. Nithyakalyani, S. Ramkumar, and K. Manikandan, "Design and implementation of sign language translator using microtouch sensor," *International Journal of Scientific and Technology Research*, vol. 9, no. 1, pp. 1784–1786, 2020.
- [28] M. Abirami, P. A. Emi, N. V. Devi, and P. Padmapriya, "Sign language communication using micro touch sensor," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 15, pp. 507–514, 2018.
- [29] L. Sousa, J. M. F. Rodrigues, J. Monteiro, P. J. S. Cardoso, and R. Lam, "GyGSLA: A portable glove system for learning sign language alphabet," in *Universal Access in Human-Computer Interaction*, 2016, pp. 159–170, doi: 10.1007/978-3-319-40238-3_16.
- [30] N. Caporusso, L. Biasi, G. Cinquepalmi, G. Trotta, A. Brunetti, and V. Bevilacqua, "A wearable device supporting multiple touch-and gesture-based languages for the deaf-blind," *Advances in Intelligent Systems and Computing*, 2018, pp. 32–41, doi: 10.1007/978-3-319-60639-2_4.
- [31] S. P. Dawane and H. G. A. Sayyed, "Hand gesture recognition for deaf and dumb people using GSM module," *International Journal of Science and Research*, vol. 6, no. 5, pp. 2226–2230, 2017.
- [32] S. P. More and A. Sattar, "Hand gesture recognition system using image processing," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Mar. 2016, pp. 671–675, doi: 10.1109/ICEEOT.2016.7754766.
- [33] S. V. Matiwade and M. R. Dixit, "Electronic support system for deaf and dumb to interpret sign language of communication," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 5, no. 5, pp. 8683–8689, 2016.
- [34] D. Abdulla, S. Abdulla, R. Manaf, and A. H. Jamdal, "Design and implementation of a sign-to-speech/text system for deaf and dumb people," *2016 5th International Conference on Electronic Devices, Systems and Applications*, Dec. 2016, pp. 1–4, doi: 10.1109/ICEDSA.2016.7818467.
- [35] N. M. Kakoty and M. D. Sharma, "Recognition of sign language alphabets and numbers based on hand kinematics using a data glove," *Procedia Computer Science*, vol. 133, pp. 55–62, 2018, doi: 10.1016/j.procs.2018.07.008.
- [36] A. Pino and G. Kouroupetroglou, "ITHACA: An open source framework for building component-based augmentative and alternative communication applications," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 2, no. 4, pp. 1–30, Jun. 2010, doi: 10.1145/1786774.1786775.
- [37] P. N. Huu, H. N. Thi Thu, and Q. T. Minh, "Proposing a recognition system of gestures using MobilenetV2 combining single shot detector network for smart-home applications," *Journal of Electrical and Computer Engineering*, pp. 1–18, Feb. 2021, doi: 10.1155/2021/6610461.




- [38] S. Ricardo, "A practical tablet -based hearing aid configuration as an exemplar project for students of instrumentation," *Audiology Research*, vol. 5, p. 135, 2015, doi: 10.4081/audiores.2015.135.

BIOGRAPHIES OF AUTHORS






Kamoli Akinwale Amusa    received the B.Eng. degree in Electrical Engineering from the University of Ilorin, Ilorin, Nigeria, the M.Sc. degree in Electrical and Electronics Engineering from the University of Lagos, Akoka, Nigeria and the Ph.D. degree in Electrical and Electronics Engineering from the Federal University of Agriculture, Abeokuta, Nigeria. His major field of study is communications engineering and digital signal processing. He has published few papers on antennas and signal processing. His current research interests include electromagnetic fields, signal processing, antennas and radio-wave propagation. He teaches at Federal University of Agriculture, Abeokuta, Nigeria. He works at Federal University of Agriculture, Abeokuta, Nigeria. He can be contacted at email: amusaka@funaab.edu.ng.






Ayorinde Joseph Olanipekun    received the B.Sc. degree in Electrical and Electronic Engineering from the University of Ado-Ekiti, Ado-Ekiti, Nigeria, M. Eng. in Electrical and Electronics Engineering from the Federal University of Agriculture, Abeokuta, Nigeria and currently pursuing his Ph.D. degree in Electrical and Electronics Engineering from the Federal University of Agriculture, Abeokuta, Nigeria. His field of interest is in the area of electronics and control system design, and software programming. Currently, he teaches at Federal University of Agriculture, Abeokuta, Nigeria. He can be contacted at email: ayorindelanny@yahoo.com.






Tolulope Christiana Erinosh    obtained B.Eng. and M.Eng. in Electrical and Electronics Engineering from the Federal University of Agriculture, Abeokuta and University of Lagos, Akoka, Lagos, respectively. She is currently undergoing her doctorate degree in Electrical and Electronics Engineering with bias in Telecommunication and Digital Signal Processing at the Federal University of Agriculture, Abeokuta. Her research areas of interest are: electromagnetic fields, radio frequency energy harvesting and clean energy. Currently, she teaches at Federal University of Agriculture, Abeokuta, Nigeria. She can be contacted at email: toluerinosho@gmail.com.



Abiodun Akeem Salaam    had his B. Eng. degree in Electrical and Electronics Engineering from the Federal University of Agriculture, Abeokuta. He is currently pursuing his M.Sc. degree with bias in software programming and application development. He can be contacted at email: salaamakeem@gmail.com.



Sodiq Segun Razaq    obtained his B.Eng. degree in Electrical and Electronics Engineering from the Federal University of Agriculture, Abeokuta. His interest is in the area of consumer software and applications development. He can be contacted at email: sodiqrazaq@gmail.com.