Solution Review

Introduction

SignPal is a python application that uses a webcam to accept American Sign Language (ASL) fingerspelling gestures and returns its translation in real-time via displaying it on the screen. This was achieved by using various python libraries to create a base interface for the application. Then a neural network model was created for fingerspelling to be used as a detector. The model was trained on an ASL dataset using Google's Teachable Machine website. With this, a functional ASL fingerspelling detector was achieved.

The rest is organized as follows: Literature Comparison provides an overview of literature related to the Sign Language Translation and compares it to SignPal. Conclusion summarizes the findings and provides future guidance for development in the field.

Literature Comparison

Comparison 1:

A Sign Language Translator proposed by Zhang et al. (2020) uses an existing model developed by Camgoz et al. (2018) and improves on the it to translate longer sentences. The model has a multimodal architecture with a convolutional neural network (CNN) and Neural Machine Translation (NMT) connected sequentially. The CNN extracts image features while the NMT encodes and decodes generated target sentences (Zhang et al. 2020).

The model used in SignPal is a simple neural network that is trained on images and its respective labels (supervised training) to recognize basic fingerspelling gestures. It is not capable of forming sentences since the input is in video format, meaning that there are multiple frames (that are closely related to each other) being passed into the detector, from which each frame could have redundant information, making it extremely challenging to form sentences (Zhang et al. 2020). It would require long-term dependencies and greater computational costs, both of which are unattainable. Moreover, the idea is beyond the scope of the assessment and is a very high-level task.

Features	SignPal	Improved SLT
Translate Alphabets	✓	✓
Translate Numbers	✓	✓
Form Sentences		✓
Multimodal Architecture		✓

Table 1. SignPal vs Improved SLT.

The reason why these two detectors are compared is because both of them are sign language detectors. Although the Improved SLT (Zhang et al. 2020) has a lot more advantages over SignPal, it is computationally heavy on the system and might be unsuitable for low performance environments. Hence, it would be recommended to reduce these costs to not only make it compatible with low level systems, but also to improve performance on high level systems.

Comparison 2:

A paper by Van der Merwe et al. (2021) proposes a South African Alphabet Sign Language (SASL) detector that uses a trained artificial neural network for translation. This is achieved using a glove equipped with flex and touch sensors for fingerspelling purposes. The data is then passed through to the application that pre-processes it and runs it through the Neural Network (see Fig. 1). The NN outputs the result, which is then displayed on the computer application. The trained model achieved an accuracy of 96% on unseen data (see Fig. 2).

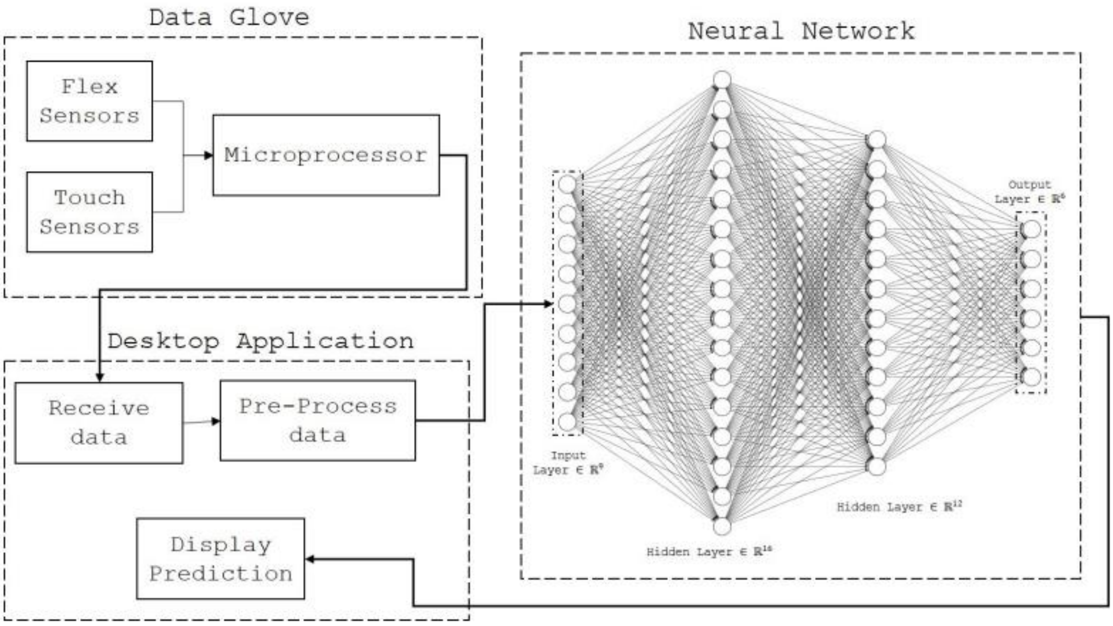


Figure 1. African Sign Language Detector System Design (Van der Merwe et al. 2021).

SignPal, although trained on American Sign Language (ASL) alphabets, use a simpler network to translate sign language. Unlike the SASL Detector, it uses a simple neural network trained on labelled images to recognize ASL fingerspelling. It is not only computationally light on the system but is also a better way of sign language translation. The accuracy of SignPal detector is shown in Figure 3.

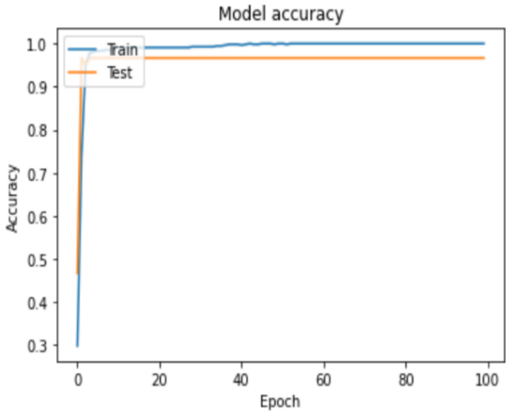


Figure 2. Model Accuracy of SASL (Van der Merwe et al. 2021).

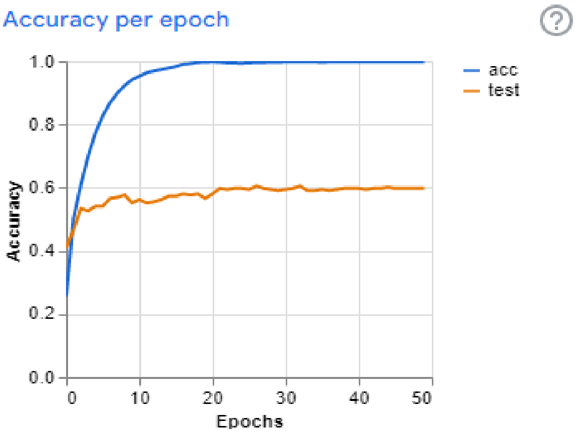


Figure 3. Model Accuracy of SignPal ASL Detector.

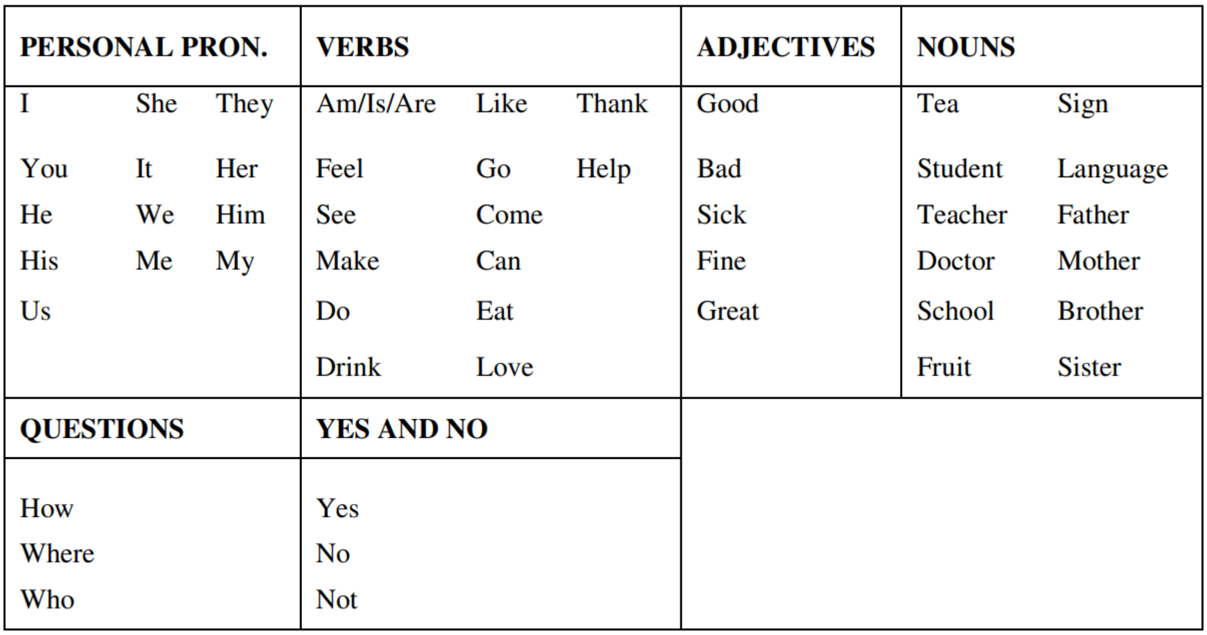
Although the test results of SignPal fall behind in comparison, it is to be noted that all of it was done using object detection to detect user's hand gestures through a webcam. However, in the case of the SASL detector, expensive equipment is required to make translation possible. Hence, it is recommended to utilize object detection techniques and technologies as the application base before proceeding into advanced realms of sign language detection.

Comparison 3:

An experiment conducted by Arsan & Ulgen (2015) uses an Xbox 360 Kinect camera device to make sign language detections. It has features such as 3D depth sensors, RGB camera, motorized base, and a multi-array mic (Arsan & Ulgen, 2015). It is calibrated to recognize all features points of the human user and process bodily gestures effectively using the depth sensors, which allows it to understand complex movements accurately. Moreover, the multi-array mic is capable of picking up voice lines that can be translated into sign language for the deaf user to understand. Meanwhile, SignPal uses the webcam attached to the computer to process the frames and has no sense of depth.

The application proposed by Arsan & Ulgen (2015) provides speech recognition conversion to sign language and vice versa, making it extremely flexible. On the other hand, SignPal only offers basic one way translation.

The literature application (Arsan & Ulgen, 2015) uses a wide variety of words for translation purposes (see Fig. 4) making it extremely convenient for public stores and services to communicate with deaf people. For example, it can be used to order a drink at Starbucks or attend a doctor's appointment. Meanwhile, SignPal only translates alphabets and numbers (see Fig. 5) and can be used in educational scenarios to teach young kids how to fingerspell using an interactive interface and engaging on-screen overlay translation.



PERSONAL PRON.			VERBS			ADJECTIVES	NOUNS	
I	She	They	Am/Is/Are	Like	Thank	Good	Tea	Sign
You	It	Her	Feel	Go	Help	Bad	Student	Language
He	We	Him	See	Come		Sick	Teacher	Father
His	Me	My	Make	Can		Fine	Doctor	Mother
Us			Do	Eat		Great	School	Brother
			Drink	Love			Fruit	Sister

QUESTIONS	YES AND NO
How	Yes
Where	No
Who	Not

Figure 4. All words for speech and sign language recognition (Arsan & Ulgen, 2015).

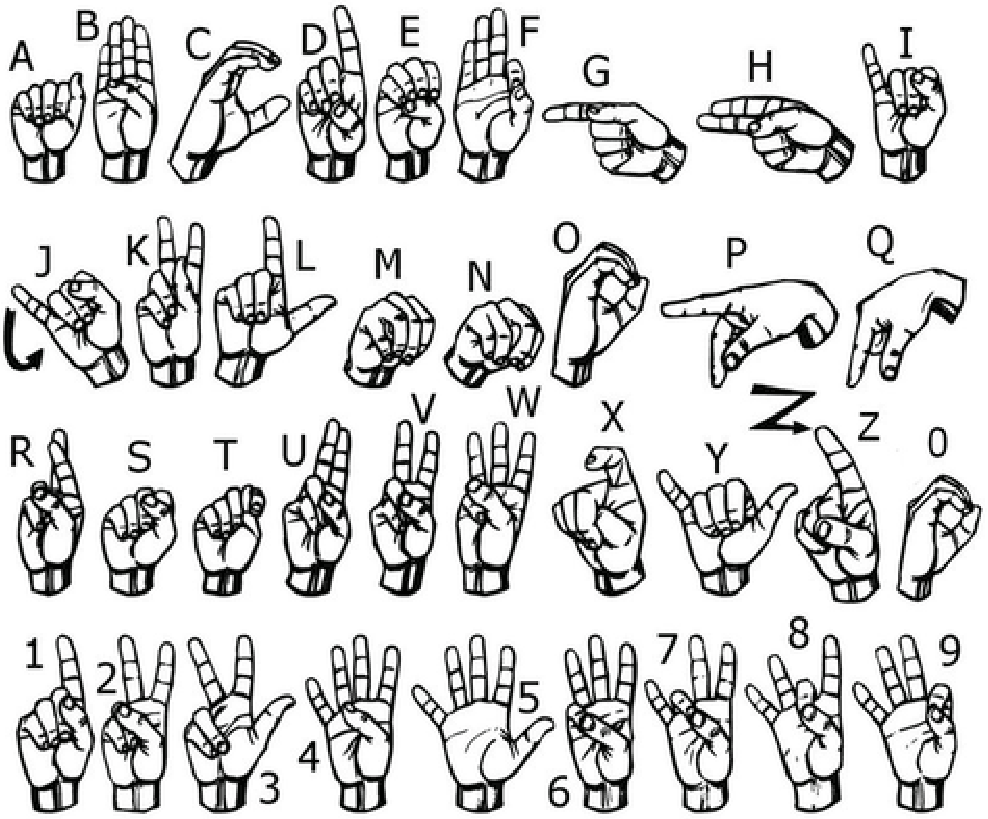


Figure 5. All alphabets and numbers that SignPal can translate (Lee et al., 2020).

Below is the comparison between the Literature application and SignPal flowchart:

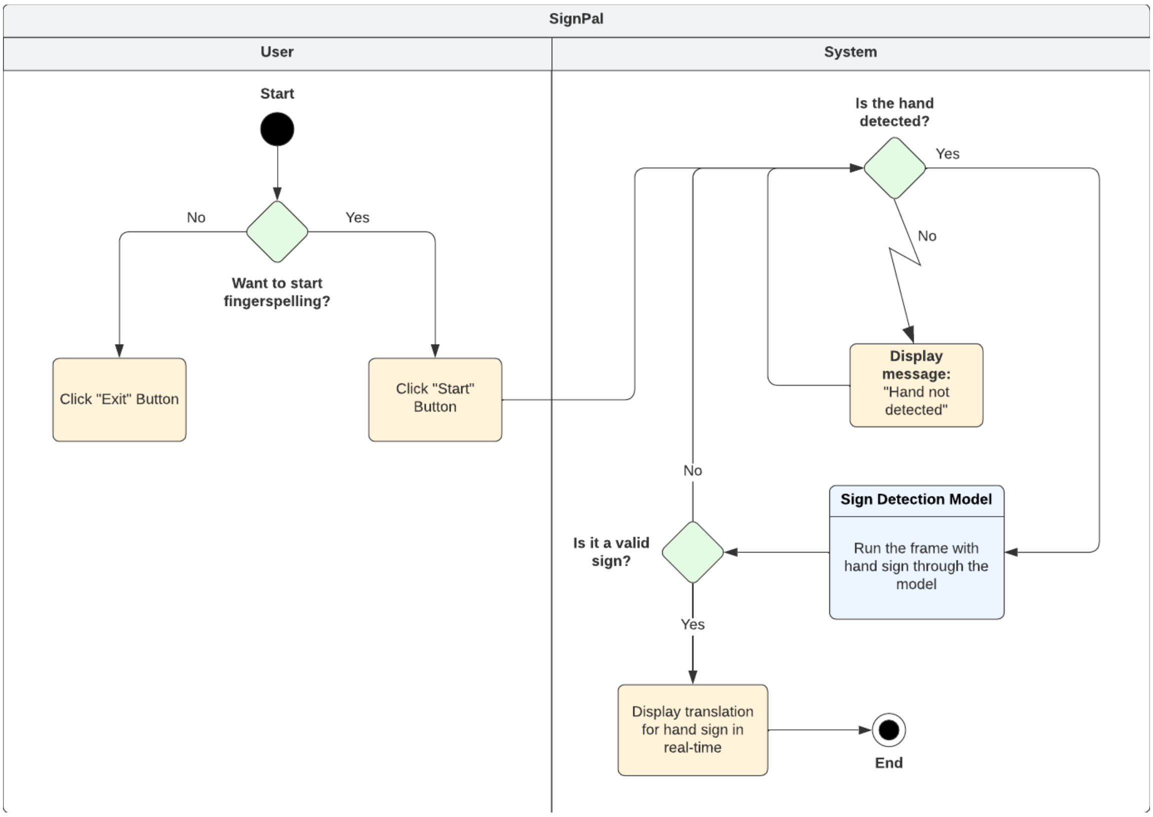

SignPal
User
System
Start
Want to start fingerspelling?
No
Yes
Click "Exit" Button
Click "Start" Button
Is the hand detected?
Yes
No
Display message: "Hand not detected"
Sign Detection Model
Run the frame with hand sign through the model
Is it a valid sign?
No
Yes
Display translation for hand sign in real-time
End

Figure 6. SignPal Flowchart

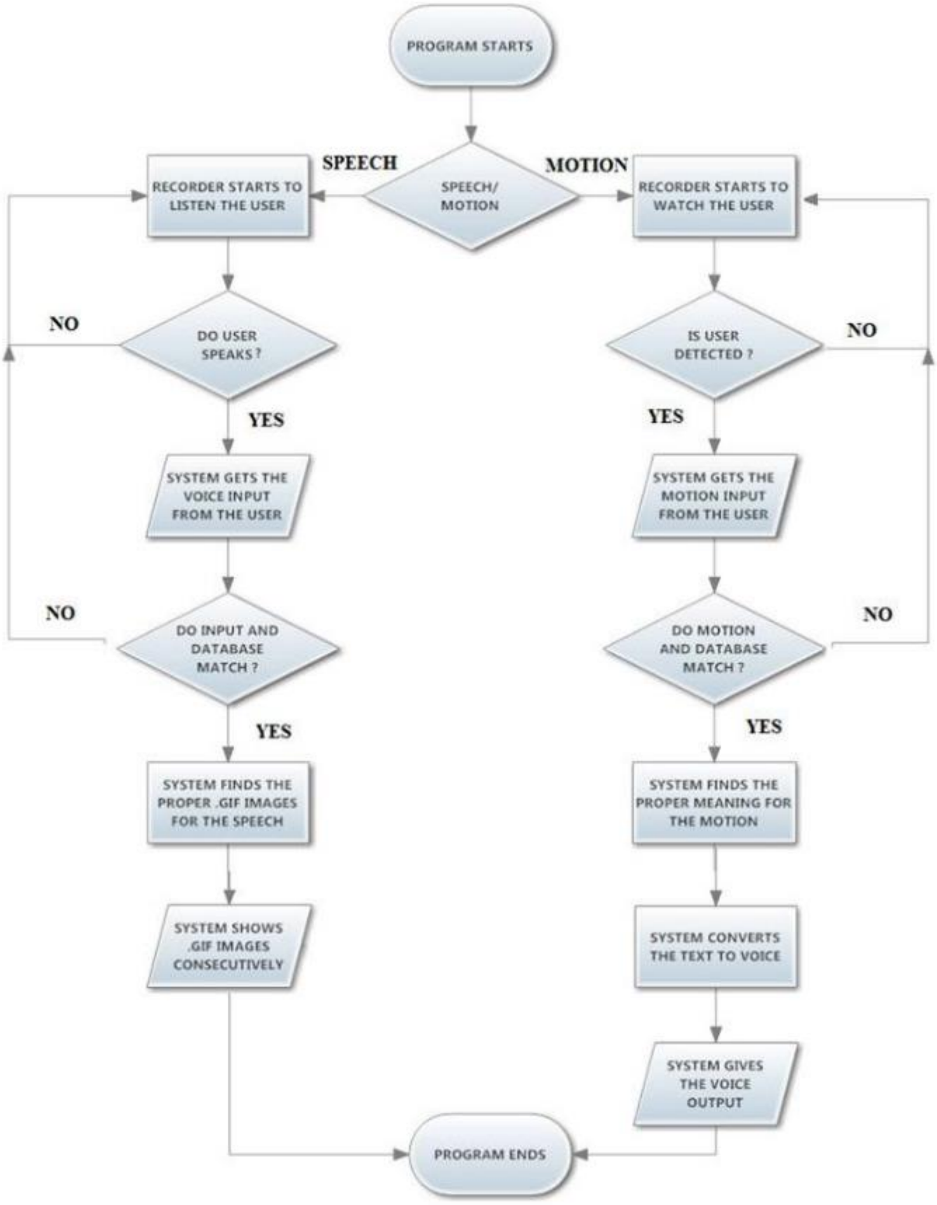

PROGRAM STARTS
SPEECH
MOTION
SPEECH/ MOTION
RECORDER STARTS TO LISTEN THE USER
RECORDER STARTS TO WATCH THE USER
NO
DO USER SPEAKS ?
IS USER DETECTED ?
NO
YES
YES
SYSTEM GETS THE VOICE INPUT FROM THE USER
SYSTEM GETS THE MOTION INPUT FROM THE USER
NO
DO INPUT AND DATABASE MATCH ?
DO MOTION AND DATABASE MATCH ?
NO
YES
YES
SYSTEM FINDS THE PROPER .GIF IMAGES FOR THE SPEECH
SYSTEM FINDS THE PROPER MEANING FOR THE MOTION
SYSTEM SHOWS .GIF IMAGES CONSECUTIVELY
SYSTEM CONVERTS THE TEXT TO VOICE
SYSTEM GIVES THE VOICE OUTPUT
PROGRAM ENDS

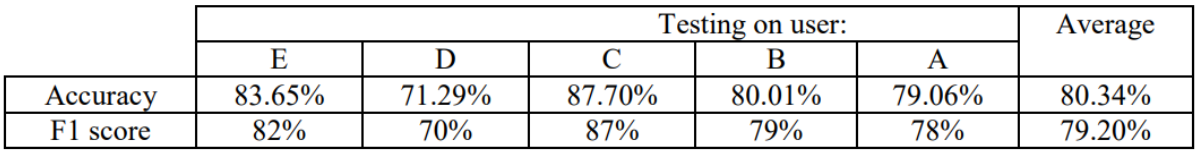
Figure 7. Literature Application Flowchart (Arsan & Ulgen, 2015).

While the literature program (Arsan & Ulgen, 2015) has a flowchart similar to SignPal, it has multiple differences. Firstly, the literature program (Arsan & Ulgen, 2015) has speech translation as well as sign translation while SignPal only has sign translation. The literature program (Arsan & Ulgen, 2015) searches for the user to make translations while SignPal looks for the user's hand. The literature application (Arsan & Ulgen, 2015) is more suitable for social interactions in businesses while SignPal is built for educating deaf children. Although the literature program (Arsan & Ulgen, 2015) uses a Kinect sensor to record the user gestures, it makes it less portable as reusing the application in different scenarios would require relocating the computer and the Kinect device. Not to mention, the device would need to calibrate to each new environment. Although SignPal is also a computer application, it has the potential to be functional in mobile devices and tablets since it is computationally lighter and does not require an external camera device for detection purposes. Although this is an unfair comparison, it highlights how each application has its own place and time of use.

Comparison 4:

A paper published by Ameen and Vadera (2017) proposes as application very similar to SignPal that detects fingerspelling gestures to recognize American Sign Language (ASL). However, the literature model (Ameen and Vadera, 2017) is a customized CNN that only recognizes ASL signs from images. The paper (Ameen and Vadera, 2017) also discusses the probable errors and confusion that can be caused due to the similarities in the signs as well as the different angles that users might use to fingerspell. This is very well the problem that SignPal is experiencing. However, this is partly solved in the literature (Ameen and Vadera, 2017) by using another input channel for the CNN network to input the depth of the user’s hand. However, it still finds it difficult to recognize similar signs if the hand is close to the camera.

As shown in Figure 8, The first 5 alphabetical letters were tested in the literature (Ameen and Vadera, 2017), the average accuracy being 80.34% (see Figure 8). SignPal is tested on the first 5 alphabets as shown in Table 2 with an average accuracy of 70.6%. Although the testing results for SignPal are not as good as the literature model (Ameen and Vadera, 2017), SignPal is achieving these results with a standard neural network as opposed to a custom multi-layered CNN that uses the depth values of the user’s hand from still images to make detections. SignPal is doing it in real-time with only 9.74% lesser accuracy.



	Testing on user:					Average
	E	D	C	B	A	
Accuracy	83.65%	71.29%	87.70%	80.01%	79.06%	80.34%
F1 score	82%	70%	87%	79%	78%	79.20%

Figure 8. Average Accuracy and F1 score of Literature Model on the first 5 alphabets (Ameen and Vadera, 2017).

Class	Accuracy
A	85%
B	54%
C	75%
D	82%
E	57%
Average	70.6%

Table 2. SignPal Accuracy of the first 5 signs.

Overall, the literature model (Ameen and Vadera, 2017) is impressive but does not provide detail about real-time functionality. Hence, it is undetermined whether it can function in real-time situations. Although SignPal has a lower accuracy of 61.11% as opposed to 80.34% of the literature model (Ameen and Vadera, 2017), it is fully capable of functioning in real-time.

It is recommended to implement the literature model (Ameen and Vadera, 2017) in real-time applications by passing data to the two inputs of the CNN model using a depth camera for greater accuracy and real-time translations.

Conclusion

In conclusion, the literature has multiple features that are superior to SignPal whereas there are a few features in SignPal that are superior to others. SignPal could take inspiration from Arsan & Ulgen (2015) to implement voice-to-sign language translation and from Ameen and Vadera (2017) to use a CNN to train the model to successfully identify asymmetric signs for higher model accuracy. Overall, it would be appropriate to utilize external cameras like Arsan & Ulgen (2015) did for business usage where relocation is not frequent. However, it would be recommended to implement such translation features using portable devices such as tablets and smartphones for greater flexibility and cost reduction. With increasing development in artificial intelligence, specifically object detection, it would be the best choice to train a CNN model to translate sign language for further development in the field.

References

Zhang, N., Zheng, J., Zhao, Z., Chen, M., Chen, J., Wu, C., Chen, Y., Shi, X., & Tong, Y. (2020). An Improved Sign Language Translation Model with Explainable Adaptations for Processing Long Sign Sentences. Computational Intelligence and Neuroscience, 2020, 8816125. doi: 10.1155/2020/8816125

Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). Neural sign language translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7784-7793).

Van der Merwe, A., Ngomseu Mambou, E., & Swart, T. G. (2021). Automated Sign Language Alphabet Detection. In 2021 IEEE AFRICON (pp. 1-6). Arusha, Tanzania, United Republic of. doi: 10.1109/AFRICON51333.2021.9570931.

Arsan, T., Ulgen, O. (2015). Sign Language Converter. International Journal of Computer Science & Engineering Survey. 6. 39-51. 10.5121/ijcses.2015.6403.

Ameen, S. and Vadera, S., 2017. A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images. Expert Systems, 34(3), p.e12197.

Lee, Boon Giin & Chong, Teak-Wei & Chung, Wan-Young. (2020). Sensor Fusion of Motion-Based Sign Language Interpretation with Deep Learning. Sensors. 20. 10.3390/s20216256.