SEMARAK ILMU PUBLISHING

Semarak International Journal of Current Research in Language and Human Studies

Journal homepage:
https://semarakilmu.my/index.php/sijcrlhs/index
ISSN: 3083-9572

An Education-Based System for Two-Way Translations of Sign Language

Muhammad Faiz Aiman Che Umar[1], Wong Hock Tze @ Farrah Wong[1,*], Sariah Abang[2], Jamal Ahmad Dargham[1], Chung Seng Kheau[1], Mazlina Mamat[1]

[1] Faculty of Engineering, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu, Sabah, Malaysia
[2] Faculty of Engineering, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia

ARTICLE INFO

Article history:
Received 25 April 2025
Received in revised form 12 May 2025
Accepted 6 June 2025
Available online 30 June 2025

Keywords:
Sign language translations; 2D CNN; educational system

ABSTRACT

There are approximately 70 million people with hearing disability and at least over 200 sign languages exist. People with hearing disability seldom feel excluded since only few people able to communicate with them. The two-way sign language translator system is a project aimed to aid deaf and hard hearing person to communicate and helping people learning sign language. The research begins with a comparison of various algorithms used in previous studies, aiming to identify the most suitable one. The 2D CNN algorithm is selected and implemented in six steps, including data acquisition from a dataset of 28 different static sign language signs obtained from Kaggle. The data is then pre-processed, and features are extracted and selected, leveraging a pretrained weight of EfficientNetB0 for better generalization. A 2D CNN architecture is utilized for model training, comprising three layers which are 2D global average pooling layer, dropout layer, and fully connected layer with one node. Mean absolute error and the Adam optimizer are employed as the loss function and optimizer, respectively. Fine-tuning the model involves experimenting with different batch sizes and iterations, with a batch size of 32 and 100 iterations yielding the best results. The model's performance is assessed, achieving 97.3% accuracy within 12 trials through Android mobile applications. The model exhibits occasional misclassifications, primarily for certain hand orientations, such as 'I', 'N', 'E', and 'S'. The model is further tested in various scenarios, demonstrating its robustness with 96.4% accuracy in complex backgrounds and a lower accuracy of 60.7% in poor lighting conditions. Moreover, the results indicate that different body mass index (BMI) can influence the model's performance since higher BMI tend to have less joint flexibility. To enhance the system's functionality, future work could focus on continuous signing, dataset expansion, and adaptive learning. These improvements aim to broaden the system's capabilities and ensure more inclusive communication for individuals with hearing disabilities.

1. Introduction

Thanks to the development of the formal sign language from B.C 1500 until the present, people with hearing impairment can now access spoken language. Many deaf have shown that despite their disability, they managed to be successful, for instance Laura Bridgman, Helen Keller and William

[*] *Corresponding author.*
E-mail address: farrah@ums.edu.my

https://doi.org/10.37934/sijcrlhs.3.1.2538

"Dummy" Hoy to name a few. However, the access is now still limited as not everyone in this world knows sign language and modern signing systems have a myriad of different rules as well as types of sign language.

In 2023, based on research done by Adrián Núñez-Marcos *et al.,* [1], it is noted that, there are about 70 million people with hearing disability around the world. Moreover, according to them, also at least over 200 sign languages exist in the world. Every year, we frequently heard a story about how people with hearing disability feel that they are not feel included whenever they are out in a public place. Most of it is because almost all people do not know how to communicate with them in sign language. Moreover, there are also issues where a physically and mentally healthy parent does not know how to interact and teach their kid with hearing disability. The reason why this project was created as medium to help and hearing disability community and hard-heard people interact with ease with other people and a platform to learn and teach sign language conveniently.

With the advancement of AI and how it is flexible in deployment, many researchers had tried to implement AI in countering the language barrier problem for people with hearing disability and non-heard community. Most of the past researcher are reluctant in implementing CNN as the main method of algorithm. The main reason why researchers choose this method because the convolutional layer built-in is able to decrease a high and complex of images while preserving information. According to previous study, CNN also has significantly increased the accuracy of many machine learning tasks and is effective at handling picture classification and identification issues [2]. Moreover, Convolutional neural network (CNN) is a multilayer neural network and the most traditional and widely used deep learning framework and hence many researchers used it to create a sign language translation system.

A virtual sign language translator on smartphone was proposed by Ku *et al.,* [3] based on CNN. In this paper, the researcher used CNN algorithm and a real time multi-person human detection from OpenPose library deployed on a smartphone. In this method it is reported that the model needs a high computational power to run the model. The researcher also needs to sacrifice the performance by reducing the input frames from 100 to 10 to gain a satisfactory result in real time. Moreover, the model also only has 3 gestures for sign language translation.

According to model proposed by Agapito *et al.,* [4] for a sign language recognition using CNN, it achieved the accuracy of test set for almost 96% with 4.16% false positive rate caused by noise environment. In this model, it had 2 different CNN model to extract hand and upper body features respectively, and ANN to concatenate both CNN result. Moreover, the model proposed also used Local Contrast Normalization (LCN) with a Rectified Linear activation function, dropout and data augmentation to achieve a high accuracy. Although it achieved a high accuracy percentage but the model only tested on an unedited video sample of a user signing 10 to 20 movements which indicate that the model proposed is not a real time prediction and the test set is only based on a single user for a limited gesture only.

A deep learning in vision-based static hand gesture recognition based on CNN and Stacked Denoising Auto Encoder (SDAE) was proposed by previous study [5]. The database of the gesture was collected from Thomas Moeslund's gesture recognition database. The proposed method by them were able to predict from public database with accuracy of 91.33% and 92.83%. According to them, they believed that adding rectified linear activations in the network's hidden layers can help overcome the difficulties in learning seen in CNNs with higher depth because they lessen the effect of neurons being saturated and disappearing gradients. Although their model achieved a high accuracy but their model only able to detect a hand at a time. The next method was proposed by Alnaim *et al.,* [6] which integrated with a smartphone camera. This proposed method able to achieve a 100% accuracy for training while 99% accuracy for testing but this model method lack hand gestures which are only seven gestures used.

In one study, a hand gesture recognition using an adapted CNN with data augmentation was proposed. In this method, the researcher retrieved hand gesture dataset from Leeds Sports Pose (LSP) [7]. The researcher had done two different methods by comparing the baseline CNN without tuning and the proposed method using ADCNN. The baseline CNN achieved 95.73% accuracy while the ADCNN achieved 99.73% accuracy. However, this proposed method only use 6 gestures which deemed as limited. Besides that, Harini *et al.,* [8] had also proposed a sign language translation using CNN method. The differences were only that the dataset is from local meaning that it captures through the webcam before training. A total of 99.91% accuracy was achieved from the proposed method. However, the proposed method does not able to predict accurately in a poor lighting condition.

Next, deep learning-based sign language recognition system for static signs was proposed in previous study [9]. In this method, the researcher was using CNN with 3 different optimizers which were Adam, RMSProp and SGD. The method also was computed on a strong computational hardware which used Tesla K80 Graphical Processing Unit (GPU) with a 12 GB memory and 64 GB Random Access Memory (RAM) using a 100 GB Solid State Drive (SSD). The researcher reported that the SGD had outperformed ADAM and RMSProp optimizers with validation accuracy of 99.90% and 98.70% on a grayscale image dataset.

Another researcher had proposed another method using CNN in recognizing America Sign Language [10]. It incorporated inception model which based on CNN with RNN and LSTM. The dataset was curate from Neidle *et al.*, [11] and it was trained using GeForce GTX 920 GPU. The model deployed using iPhone 6 with 60fps camera and 720-pixel resolution was able to achieved 91% accuracy using SoftMax and 55% for pool layer with 150 signs. However, this model performs poorly when there was a variation in clothing and it is video based translation which not very efficient. Next, in one study, a deep CNN sign language recognition system using CNN and Google Net transfer learning was implemented [12]. The dataset for this proposed method was retrieved from random internet website and the training was done using MATLAB and a Windows 10, intel core i7 with 16 GB RAM and GTX 1060 GPU. This proposed method was able to get 91.02% of accuracy for the sign language recognition. Despite the impressive accuracy, this methodology lacks credibility since the dataset was retrieved from random internet sites the researcher did not specify how many datasets was taken.

Lastly, for CNN method, a researcher proposed Indian sign language numeral recognition using region of interest CNN [13]. In this method, the researcher used CNN with Contrast Limited Adaptive Histogram Equalization (CLACHE). The system was realize using RGB camera and the model was train on NVIDIA GeForce 920MX having a 2GB of graphic memory and i5 CPU processor with 8GB RAM. This proposed system attained an accuracy of 99.56% in the same subject testing while 97.26% in a low light condition. Although the proposed model was able to counter the low light condition problem but this model only using 9 different classes of gesture from 0 to 9 which is limited and does not represent the real sign language dataset.

A proposed model by Venugopalan *et al.,* [14] using a hybrid deep neural network model with a pretrained 2-D CNN from Google Net as the front end for feature extraction and the bidirectional LSTM (BiLSTM) sequence network for feature classification. The model was trained on a 4Gz i5-4210 CPU with 8GB of memory via MATLAB software. The proposed method obtained a satisfactory percentage of accuracy with 76.21% which the researcher suggest that this model is scalable and able to be implemented to huge hand gestures dataset with GPU-based implementation. However, this shown that the proposed method contains a limited number of datasets.

Another researcher proposed a new method for hand sign language recognition using multi view hand skeleton [15]. A 2D CNN, 3D CNN and LSTM were used by the researcher with Intel® Xeon® CPU E5-2699 and 90GB of RAM and 10 NVIDIA GPU. The model was trained using phyton language. From the result, the researcher claimed that their proposed method had provides an incremental performance and justify all their design choice. However, the researcher did not report on the accuracy obtained for the prediction which make and unreliable method.

Moreover, another research for real time recognition of Bangla sign language character proposed a method using a 2D-CNN architecture [16]. In this system, the researcher obtained dataset from public images using a high-definition webcam. From the research, it was reported that it achieved a 99.72% validation accuracy. Despite that high accuracy, the prediction accuracy would decrease as the angular position of hand increases. Lastly, a real time system for recognition of American sign language using 2D-CNN has been proposed in previous study [17]. In this proposed system, it used OpenCV library to predict it in real time and a GPU to trained the model. The researcher claim that the accuracy attained is 98.05% in real-time environment. However, the system proposed was tested only in a constant environment meaning that it does not account the variation of background and lighting problem.

Research about a pattern recognition model for static gestures in Malaysia Sign Language based on machine learning was done in study by Alrubayi *et al.,* [18]. In this research, the authors proposed a method of using vision-based gesture sensor integrated with a machine learning model. A data gloves wand flex sensor to capture the hand attributes and movement. The proposed method able to achieved a 99% using ANN and 98.5% using K-Nearest Neighbour. The data acquisition and data processing were the main problem for this method according to them since they had to create the dataset by themselves using their own hardware setup. Next, research by Núñez-Marcos *et al.,* [19], the same method has also been used which is using vision-based recognition and glove. From this proposed methodology, the researcher stated that when translating from SLs to spoken language text or vice versa, it needed a learning guidance which make it inconvenient. Moreover, the method also had a limited dataset and does not take into account a multi signer.

Furthermore, a Haar-like features based on AdaBoost classifier in previous research [20]. The model was trained in a natural fluorescent lighting condition using Intel Core i7 processor. The system able to obtained a 98.7% accuracy but it was only applicable to a hand in a time. Shibata *et al.,* [21] have proposed a method by using a coloured gloves with a decision tree method. However, the prediction is not sufficiently good since it only managed to obtained 82% although using a small type of dataset. Next, a method in study by Ahmed *et al.,* [22] has proposed a model using random forest machine learning algorithm. In this system, a total accuracy of 84% for sign language to speech and 87% accuracy for speech to sign language conversion obtained. As the system is a software based, it does not require disable person to wear any gadget. However, since the software was done on Kinect v2 SDK, the system only able to be used in device that compatible with Kinect v2 and this method also operate on a limited set of words.

Next, previous research has proposed a method of sign language system through mobile apps and database [23]. However, this proposed method does not disclose the effectiveness of the system. Johnny and Nirmala [24] had proposed a method of using gloves and sensor with 3 different algorithm which were KNN, decision tree and MLP. Out of three, KNN achieved the highest accuracy of 97% while the other two algorithms only achieved 76%. Based on the research, decision trees perform poorly because the branches are overfitted with various discrepancies. The algorithm keeps reducing the training set error at the expense of increasing the test set error, which leads to the overfitting issue. As for neural network, it was due to a lack of diverse and extensive training datasets. Other than the training set, it cannot classify any test set. The system does not generate a generic answer because it was taught to work for a particular dataset. As a result, accuracy with smaller datasets is poorer.

Based on the comparison from all the previous works, almost all proposed methods were able to achieved a high percentage of accuracy which is within the 90% range. However, most sign language translation system proposed were using a limited class of sign language gestures which is around eight to ten. This is concerning since it will highly likely affect the outcome when more classes of sign language gesture were used because when the number of classes increase the more complex the AI model would be. Furthermore, some of the previous researcher also facing a challenge to get a correct real time prediction when it is being predicted in a poor lightning condition, different background and different hand colours. Moreover, some of the previous works were only applicable to certain type of powerful hardware application only that are expensive like Kinect because the AI model was power-hungry. Among all the proposed methodology, CNN method is proven to be the most suitable and efficient due to high percentage of accuracy despite being implemented alone without combining with other method. Within the CNN method, the 2D CNN seems like a reliable option for sign language detection since it is able to analyse and process any dataset into 2 dimensions, high and width which is good enough to create a real time sign language translation system.

2. Methodology

To simply put, in this project, a dataset compromising of sign language letters and static gestures were obtained, saved and train through a deep learning model which is a 2D CNN. The trained model is then saved. The saved file will be loaded into a hardware or a mobile application to make a prediction. From the hardware or the mobile application, the AI model is able to predict sign language input from user and display the result in real time. Moreover, the mobile application also has other function such as learning and practicing sign language and sign language game. This system was proposed to help break the language barrier between the people with hearing disability with normal people and as a medium to learn sign language.

Data pre-processing is crucial to every model. This is part of the reflection of the model capabilities. Having a dirty data would be not good for any model. Data pre-processing mainly involved three things, which are data cleansing, data dimension reduction and data normalization. Data cleansing is often concern with fill in any missing values or eliminate faulty data. Moreover, the operation of data cleansing includes data filtering in which will filter any dirty data, processing the lost data, possible exceptions, errors, abnormal values or combination data from multiple resources and data consolidation. Data dimension reduction is closely related to simplifying the data attributes to reduce the computational power as well as to avoid dimension explosion. Data normalization usually will normalize the data to reduce the noise and improve the accuracy. The data pre-processing steps are in the framework of artificial intelligence because, generally the real data that we obtained will have some quality problems in which we called it as dirty data. These problems will affect the outcome or prediction of our model when deployed. Noisy data consist of incompleteness, which is concerned with containing missing values or lack of attributes, noise, which has a faulty records and inconsistency.

Feature extraction and selection in AI is concerned with the data representation. From the raw data, it will be represented in a suitable form and easy to deploy for the training of the model build. Moreover, it is done to select the most relevant features to feed it to the model, which will reduce chances of overfitting as well as making it less complex.

The next process is model training. In this process, the model build will be trained on the basis of the method proposed which are 2D CNN and parameters that have been decided. The method chosen is 2D-CNN because 2D-CNN is able to filter through 2 dimension which is suitable for image detection such as hand gesture. Moreover, CNN method generally able to filter high dimensionality of images with minimal losing of information. Next, the parameters include the training rules, activation function, normalizer, and optimizers. The model is then evaluated based on a few things, such as loss function and validation rate.

The model will undergo fine tuning and upon achieving a satisfied model, the model will get deployed and integrated. A model will usually deploy and integrate in a software of a hardware that have Central Processing Unit (CPU) for instance smartphone from mobile application and laptop or computer from the desktop application. Figure 1 refers to the overall AI model system operation.

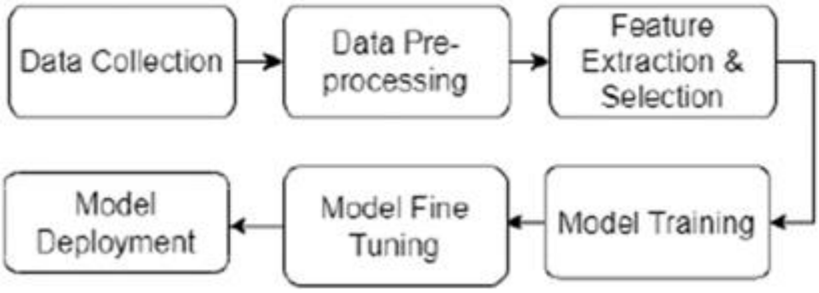

Fig. 1. The block diagram for the AI model system operation

For this project the AI model will get deployed on android mobile application using java. In this android application, it will consist of four different modules. The modules are practice module, gesture to text module, text to gesture module and game module. Figure 2 refers to the block diagram for the android mobile application system operation.

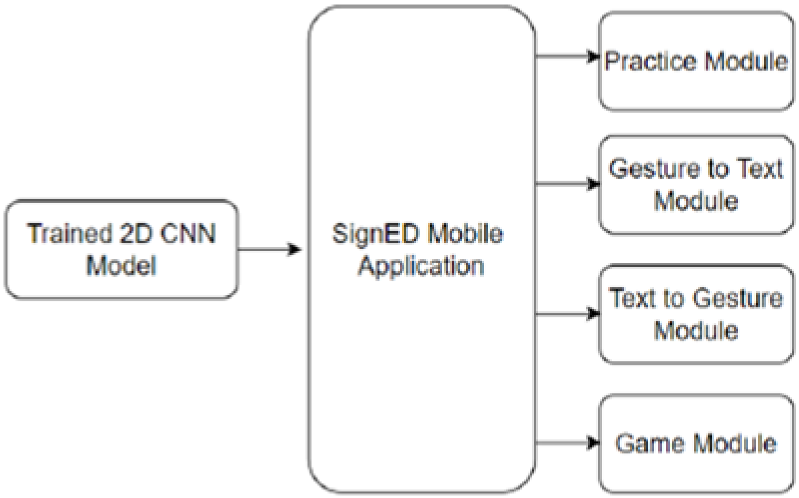

Fig. 2. The block diagram for the android mobile application system operation

In the practice module, all the 26-sign language letter will be displayed as a reference for the user to mimic and exercise their sign language letter ability. At the same time, camera module also will be initiated to capture the real time input from user. The user input then will be resized so that it will be the same as the AI model input size. This is to make sure that an accurate prediction able to be made by the AI model. The hand will be tracked via a pretrained hand detection model and a rounding box will be drawn around the tracked hand. This is to eliminate any unnecessary false detection where it only guesses if hand appears. The output from that will be fed to the CNN model and prediction will be made. Figure 3 refers to the simplified flow for the practice module.

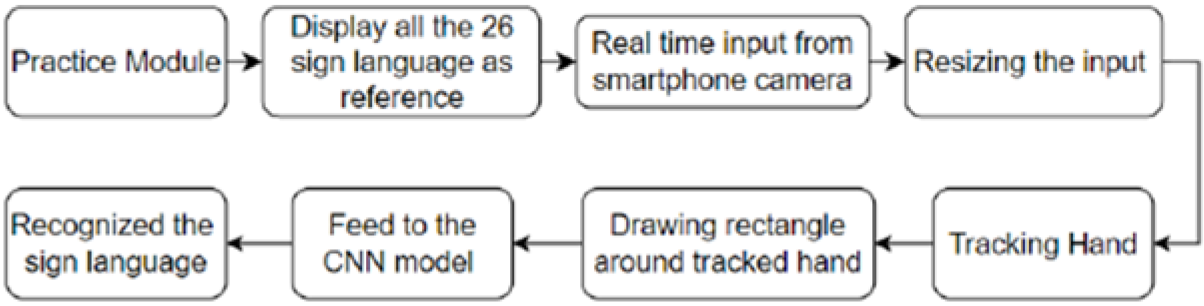

Fig. 3. The simplified flow for the practice module

In the gesture to text module, it is almost the same as practice module. Firstly, a camera module will be used to capture a real time input from user. The input from the camera then will be resize the same as the AI model input for a more robust prediction. The hand will be tracked using pretrained hand detection AI model and rounding box will be drawn. The output will be given to the proposed 2D CNN model, and it will predict the outcome. Figure 4 refers to the simplified flow for the gesture to text.

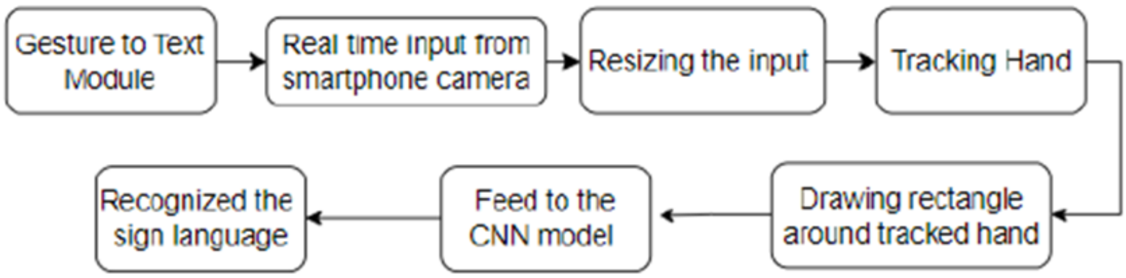

Fig. 4. The simplified flow for the gesture to text module

Next, for the text to gesture module, it will require user to input any text for the translation. Then, the sign language images corresponding to each letter the user input will appear at the background. This is not only will help for a fast-spelling checker, but it also can be used as letter-by-letter translation when needed.

As for the game module, a one-minute timer will be started and shown. At the same time, an image will be displayed. From the image, the user will have to guess the possible answer within the time limit. The aim of the game module is to access the ability of the user about sign language. Next, camera module will be initiated to capture the input from user and will be resized to be the same as AI model input size for an accurate prediction. A pretrained hand detection AI model I used to track the hand and a rounding box will be drawn. Then, the output will be fed to the CNN model and prediction will be made. Lastly, user will have to append the prediction as a text and submit the answer if the user thinks that is the correct answer. Figure 5 refers to the simplified flow for the game module.

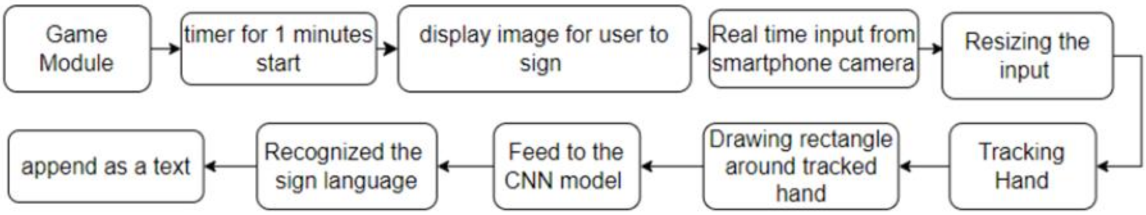

Fig. 5. The simplified flow for the game module

3. Results

Table 1 shows the result of loss function for different batch size. This suggests that the model with a batch size of 32 is performing the best. The difference in performance between the models with different batch sizes is likely due to the fact that the model with a batch size of 32 is able to see more data at once.

Table 1
The result of loss function for different batch size

Batch size	32	64	128
Loss function value	0.2142	0.2199	0.2329
Loss validation	0.1651	0.1883	0.0888
Learning rate	2.5419×10^{-4}	2.2877×10^{-4}	3.1381×10^{-4}

Next is the fine tuning the number of iterations. The results in the graph show that the loss function value decreases as the number of iterations increases for both the 100-iteration and 25-iteration models. This suggests that the models are learning and improving over time. The graph below clearly depicts that the 25-iteration used in training is still not enough. This is because, the 25-iteration graph still has a lot of fluctuation value and do not reach a stable value or plateau. This indicates that the graph is still underfit and should be trained for more iteration. For the 100 iterations, the graph clearly shows that the accuracy of training and validation data is almost stable. Table 2 shows the result for different iteration. From Table 1 and Table 2, the best hyperparameter for batch size is 32 with 100 iterations. The model is able to make accurate predictions on both the training data and the validation data.

Table 2
The result for different iteration

Iteration	100	25
Batch size	32	32
Loss function value	0.2142	0.4446
Loss validation	0.1651	0.3735
Learning rate	2.5419e-4	9.0000e-4

Figure 6 refers to the UI design of the SignEd mobile application for main page. The practice module is designed for the user to practise sign language for every letter excluding Z. In this module, a picture of each sign language was put at the bottom of the screen as a guidance for the user to follow. There is also a box for camera for sign language detection that display the result in real time. Figure 7 refers to the UI design of the SignEd mobile practice module.

Fig. 6. The UI design of the SignEd mobile application for main page

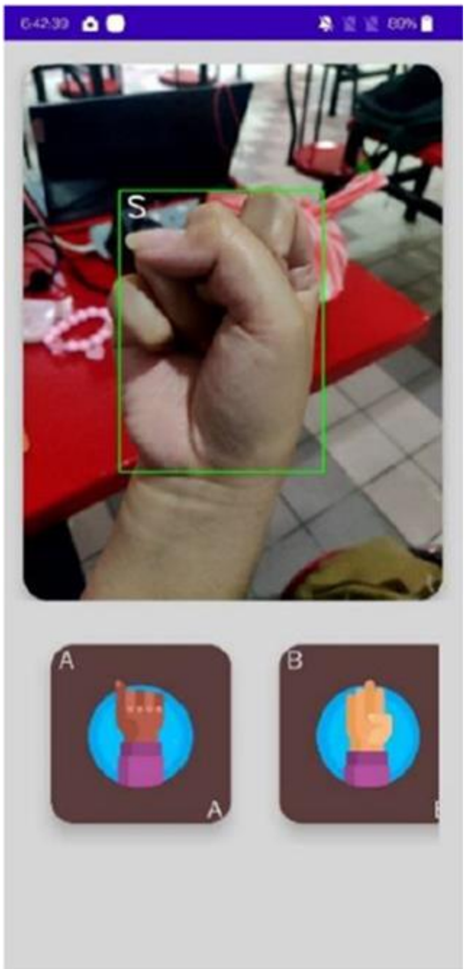

Fig. 7. The UI design of the SignEd mobile application for practice module

The next module is gesture to text module. In this module, it is used to translate any static sign language alphabets gestures to text. There is a camera box that will display the hand making it easier for user to see what sign that doing. There is also a text box that stored the predicted letter. Moreover, at the bottom of the application, there are 3 button which are ‘Add’, ‘Read’ and ‘Clear’. Figure 8 refers to the UI design of the SignEd mobile application for gesture to text module.

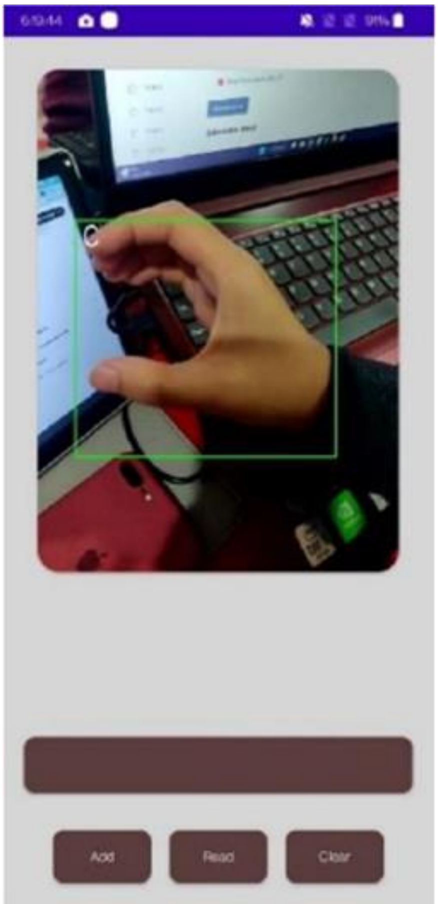

Fig. 8. The UI design of the SignEd mobile application for gesture to text module

The add button is used to append the letter. The read button was used to voice out the constructed word. By having this function, a short sentence can be form and can be translated in real time via voice. Figure 9 refers to the appending of alphabets process via add button for gesture to text module.

In the game module, the user can challenge user skills of sign language with a fun time attack finger spelling game. The goal is to achieved highest round possible within one minute. This module has a camera box for user to see their fingerspelling, an image in each round for the user to guess, add button to append letter forming word, submit button to submit the answer and clear button to clear the output if it is wrong. This module was designed to test the user sign language skills. Figure 10 refers to the UI design and game flow of the SignEd mobile game module.

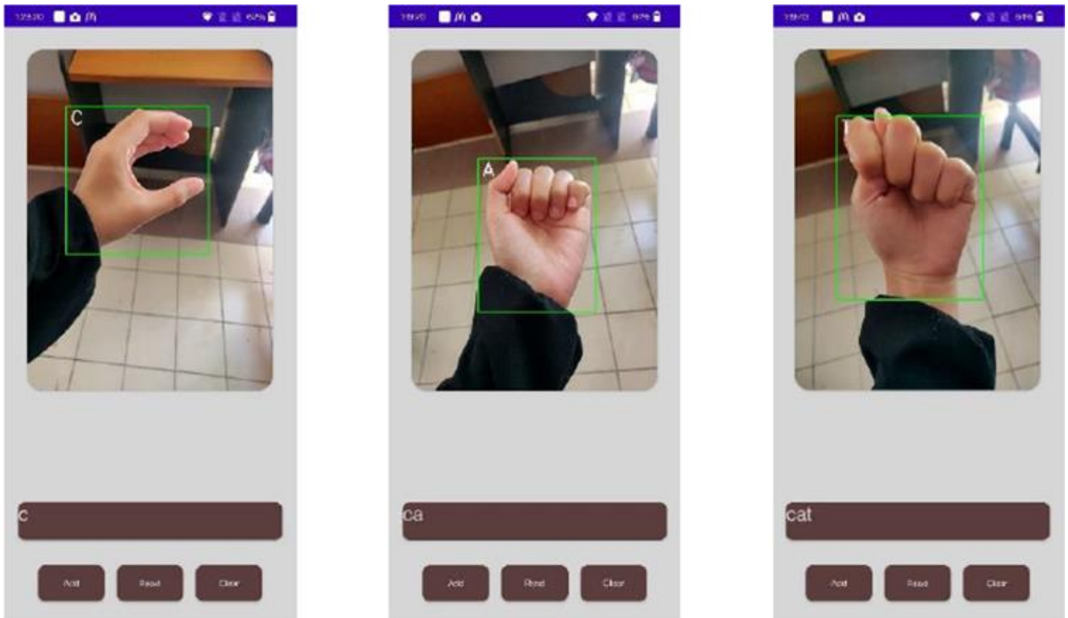

Fig. 9. The appending process of alphabets via add button for gesture to text module

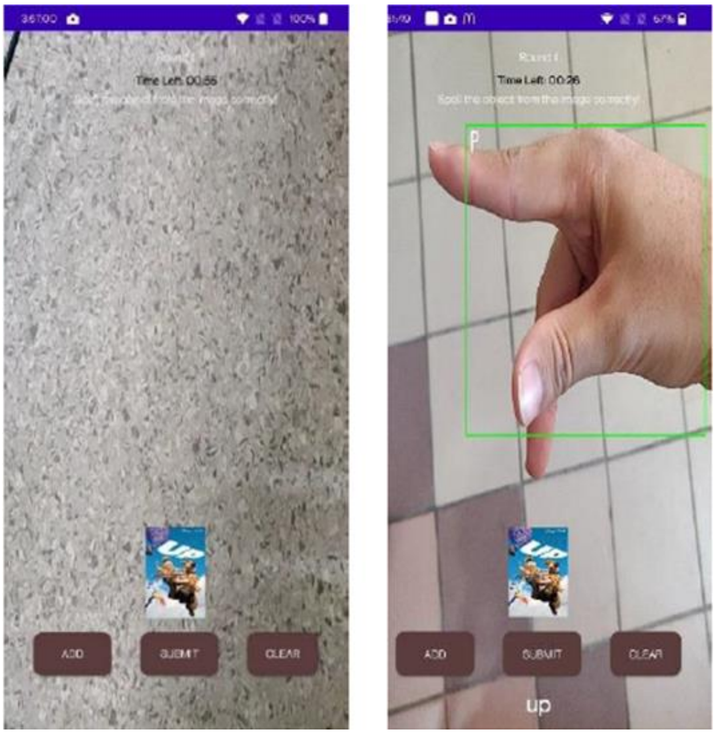

Fig. 10. The UI design and game flow of the SignEd mobile application for the game module

Table 3 refers to the accuracy for each trial and the average accuracy for the repeatability test. From the table, within the 12 trials, the AI model was able to achieved 100% accuracy for five trials, 96.4% accuracy for another five trials and 92.8% accuracy for two trials. The 100% accuracy meaning that the AI model is able to correctly identified all sign language, 96.4% accuracy shows that there is only two misprediction and for 92.8%, it depicts that the AI model make two misclassifications. The measure average of the 12 trials is 97.3%. From the result discussed, it depicts that the performance of the AI model is good. This is because, the AI model is able to perform very well with a maximum of only two error throughout the 12 trials.

Table 3
The accuracy for each trial and the average accuracy of repeatability test

	Trials											
Alphabets | Trials	1	2	3	4	5	6	7	8	9	10	11	12
Correct prediction	28	28	27	27	26	27	27	28	27	28	26	28
Incorrect prediction	0	0	1	1	2	1	1	0	1	0	2	0
Accuracy of detection %	100	100	96.4	96.4	92.8	96.4	96.4	100	96.4	100	92.8	100
Average accuracy or repeatability test	97.3											

Alphabets ‘A’, ‘B’, ‘C’ , ‘D’, ‘F’, ‘G’, ‘H’, ‘J’, ‘K’, ‘L’, ‘M’, ‘O’, ‘P’, ‘Q’, ‘R’, ‘T’, ‘U’, ‘V’, ‘W’, ‘X’, ‘Y’, and static hand gesture ‘I LOVE YOU’, and ‘GOOD JOB’ classes are able to be distinguished all correctly within the 12 trials. This is because, from the dataset these classes images are very well defined which having a lot of quality images. Next ‘E’, ‘S’ and ‘SPACE’ is the classes with second highest accuracy with 91.67% with only 1 incorrect prediction and ‘N’ achieved 2 misclassifications with recorded accuracy of 91.67%.

For example, letter ‘E’ and ‘S’ is almost similar in sign language gesture. As for ‘SPACE’ and ‘N’ it might be because of the user hand orientation because the hand orientation is slightly dissimilar for each person. The least accurate is alphabets ‘I’ with 4 wrongly identified within the 12 trial which obtained 66.67% accuracy. One plausible explanation is that it might be because of the low quality of dataset obtained for ‘I’ classes. Moreover, since the sign language of ‘I’ is almost similar with ‘J’ which the same hand orientation but only slanted to the side thus making the AI model not able to classify.

In the poor lighting condition, the lux recorded was a value lower than 50 lux and the good lighting condition is more than 650 lux. From the result recorded, it clearly shows that the AI model perform better in good lighting which obtained 85.7% of accuracy compared to 60.7% in poor lighting. The good lighting condition was able to predict 24 out of 28 correctly while the poor lighting was only able to guess 17 out of 28 correctly. The plausible reason for this is that there is fewer data that is recorded in poor lighting compared to the good lighting. Images with poor lighting often have less contrast than images with good lighting. Therefore, this makes it more difficult for a model to identify the features that it needs to make a prediction. As a result, the AI models predicts better when user in good lighting condition compared to poor condition.

In the skin colour testing, there are two types of conditions chosen which are dark and fair skin. The dark colour skin correctly predicts 23 out of 28 alphabets achieved 82.1% accuracy while the fair colour skin able to identify 25 out of 28 alphabets correctly. The plausible reason is that due to different brightness and contrast. Fair skin is typically brighter than dark skin which mean that it reflects more light. The reflecting of light makes it easier for AI model to identify the features or key point that are used to classify the sign language. As a result, models that are trained on fair skin may be slightly more accurate than models that are trained on dark skin.

Different background has been tested as well for the apps. The complex background achieved the highest accuracy of 96.4% followed by light monotone 85.7% accuracy and dark monotone with 82.1% accuracy. The plausible reason for this is because complex background has more contrast than light and dark monotone background. This will make it easier for AI model to identify the signer hand and the shapes the signer made. On top of that, complex background also has more variety than the others. This will help models to learn to focus on important features and ignore irrelevant features. This is because the model must learn to ignore the distracting background to make a correction prediction. As a result, complex backgrounds tend to produce better results in sign language prediction than light monotone or dark monotone backgrounds.

4. Conclusions

As conclusion, a 2D CNN model architecture was proposed to make a real time sign language translation system for education. From the model architecture, the result obtained is satisfactory with 0.2142 loss on training set, 0.1651 loss on validation set. Moreover, an android mobile application was also developed as a platform for the AI model deployment. In this mobile application there are 4 module which are practice module, gesture to text module, text to gesture and game module. From the mobile application two evaluation was implemented. The first evaluation is the repeatability test of the proposed architecture within satisfactory environment and condition for daily case uses. In this test, the 2D CNN model able to achieve an impressive of accuracy which are 97.3%. This shows that the AI model is generally repeatable, but there is some room for improvement for daily case uses. Last but not least, the accuracy of the developed software was tested on different condition. In this test, the accuracy for different type of background and skin colour achieved a good accuracy of 80%. For the different lighting condition, the proposed algorithm was only able to perform well in good lighting with 85.7% but bad in poor lighting with only 60.7%. This is because the dataset contained a diverse set of images that varied in background and skin colour but not diverse in different lighting condition.

Acknowledgement
This research was not funded by any grant

References
[1] Núñez-Marcos, Adrián, Olatz Perez-de-Viñaspre, and Gorka Labaka. "A survey on Sign Language machine translation." *Expert Systems with Applications* 213 (2023): 118993. https://doi.org/10.1016/j.eswa.2022.118993

[2] Xin, Mingyuan, and Yong Wang. "Research on image classification model based on deep convolution neural network." *EURASIP Journal on Image and Video Processing* 2019, no. 1 (2019): 1-11. https://doi.org/10.1186/s13640-019-0417-8

[3] Ku, Yun-Jung, Min-Jen Chen, and Chung-Ta King. "A virtual sign language translator on smartphones." In *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*, pp. 445-449. IEEE, 2019. https://doi.org/10.1109/CANDARW.2019.00084

[4] Agapito, Lourdes, Michael M. Bronstein, and Carsten Rother, eds. *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III*. Vol. 8927. Springer, 2015. https://doi.org/10.1007/978-3-319-16199-0

[5] Oyedotun, Oyebade K., and Adnan Khashman. "Deep learning in vision-based static hand gesture recognition." *Neural Computing and Applications* 28, no. 12 (2017): 3941-3951. https://doi.org/10.1007/s00521-016-2294-8

[6] Alnaim, Norah, Maysam Abbod, and Abdulrahman Albar. "Hand gesture recognition using convolutional neural network for people who have experienced a stroke." In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1-6. IEEE, 2019. https://doi.org/10.1109/ISMSIT.2019.8932739

[7] Alani, Ali A., Georgina Cosma, Aboozar Taherkhani, and T. M. McGinnity. "Hand gesture recognition using an adapted convolutional neural network with data augmentation." In *2018 4th International conference on information management (ICIM)*, pp. 5-12. IEEE, 2018. https://doi.org/10.1109/INFOMAN.2018.8392660

[8] Harini, R., R. Janani, S. Keerthana, S. Madhubala, and S. Venkatasubramanian. "Sign language translation." In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 883-886. IEEE, 2020. https://doi.org/10.1109/ICACCS48705.2020.9074370

[9] Wadhawan, Ankita, and Parteek Kumar. "Deep learning-based sign language recognition system for static signs." *Neural computing and applications* 32, no. 12 (2020): 7957-7968. https://doi.org/10.1007/s00521-019-04691-y

[10] Bantupalli, Kshitij, and Ying Xie. "American sign language recognition using deep learning and computer vision." In *2018 IEEE international conference on big data (big data)*, pp. 4896-4899. IEEE, 2018. https://doi.org/10.1109/BigData.2018.8622141

[11] Neidle, Carol, Ashwin Thangali, and Stan Sclaroff. "Challenges in development of the american sign language lexicon video dataset (asllvd) corpus." In *Proc. 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, Language Resources and Evaluation Conference (LREC) 2012*. 2012.

[12] Yemenoglu, Ismail Hakki, AFM Shahen Shah, and Haci Ilhan. "Deep convolutional neural networks-based sign language recognition system." In *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0573-0576. IEEE, 2021. https://doi.org/10.1109/IEMCON53756.2021.9623068

[13] Sajanraj, T. D., and M. V. Beena. "Indian sign language numeral recognition using region of interest convolutional neural network." In *2018 second international conference on inventive communication and computational technologies (ICICCT)*, pp. 636-640. IEEE, 2018. https://doi.org/10.1109/ICICCT.2018.8473141

[14] Venugopalan, Adithya, and Rajesh Reghunadhan. "Applying deep neural networks for the automatic recognition of sign language words: A communication aid to deaf agriculturists." *Expert Systems with Applications* 185 (2021): 115601. https://doi.org/10.1016/j.eswa.2021.115601

[15] Rastgoo, Razieh, Kourosh Kiani, and Sergio Escalera. "Hand sign language recognition using multi-view hand skeleton." *Expert Systems with Applications* 150 (2020): 113336. https://doi.org/10.1016/j.eswa.2020.113336

[16] Tanvir, Mahib, Md Shahinur Alam, Dip Kumar Saha, Shahid A. Hasib, and S. Islam. "Real-Time Recognition of Bangla Sign Language Characters: A Computer Vision Based Approach Using Convolutional Neural Network." In *2021 3rd International Conference on Electrical & Electronic Engineering (ICEEE)*, pp. 177-180. IEEE, 2021. https://doi.org/10.1109/ICEEE54059.2021.9718800

[17] Taskiran, Murat, Mehmet Killioglu, and Nihan Kahraman. "A real-time system for recognition of American sign language by using deep learning." In *2018 41st international conference on telecommunications and signal processing (TSP)*, pp. 1-5. IEEE, 2018. https://doi.org/10.1109/TSP.2018.8441304

[18] Alrubayi, Ali H., Mohamed Aktham Ahmed, A. A. Zaidan, Ahmed Shihab Albahri, B. B. Zaidan, Osamah Shihab Albahri, Abdullah Hussein Alamoodi, and Mamoun Alazab. "A pattern recognition model for static gestures in malaysian sign language based on machine learning techniques." *Computers and Electrical Engineering* 95 (2021): 107383. https://doi.org/10.1016/j.compeleceng.2021.107383

[19] Núñez-Marcos, Adrián, Olatz Perez-de-Viñaspre, and Gorka Labaka. "A survey on Sign Language machine translation." *Expert Systems with Applications* 213 (2023): 118993. https://doi.org/10.1016/j.eswa.2022.118993

[20] Truong, Vi NT, Chuan-Kai Yang, and Quoc-Viet Tran. "A translator for American sign language to text and speech." In *2016 IEEE 5th Global Conference on Consumer Electronics*, pp. 1-2. IEEE, 2016. https://doi.org/10.1109/GCCE.2016.7800427

[21] Shibata, Hirotoshi, Hiromitsu Nishimura, and Hiroshi Tanaka. "Basic investigation for improvement of sign language recognition using classification scheme." In *Human Interface and the Management of Information: Information, Design and Interaction: 18th International Conference, HCI International 2016 Toronto, Canada, July 17-22, 2016, Proceedings, Part I 18*, pp. 563-574. Springer International Publishing, 2016. https://doi.org/10.1007/978-3-319-40349-6_55

[22] Ahmed, Mateen, Mujtaba Idrees, Zain ul Abideen, Rafia Mumtaz, and Sana Khalique. "Deaf talk using 3D animated sign language: A sign language interpreter using Microsoft's kinect v2." In *2016 SAI Computing Conference (SAI)*, pp. 330-335. IEEE, 2016. https://doi.org/10.1109/SAI.2016.7556002

[23] Priya, L., A. Sathya, and S. Kanaga Suba Raja. "Indian and English language to sign language translator-an automated portable two way communicator for bridging normal and deprived ones." In *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, pp. 1-6. IEEE, 2020. https://doi.org/10.1109/ICPECTS49113.2020.9336983

[24] Johnny, Swapna, and S. Jaya Nirmala. "Sign language translator using Machine Learning." *SN Computer Science* 3, no. 1 (2022): 36. https://doi.org/10.1007/s42979-021-00896-y