Expert Systems With Applications 167 (2021) 114403

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

American sign language recognition and training method with recurrent neural network

C.K.M. Lee [a], Kam K.H. Ng [b], Chun-Hsien Chen [c,*], H.C.W. Lau [d], S.Y. Chung [a], Tiffany Tsoi [a]

[a] *Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China*
[b] *Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong SAR, China*
[c] *School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore*
[d] *School of Business, The University of Western Sydney, Australia*

ARTICLE INFO

Keywords:
American sign language
Leap motion controller
Learning application
Sign recognition system

ABSTRACT

Though American sign language (ASL) has gained recognition from the American society, few ASL applications have been developed with educational purposes. Those designed with real-time sign recognition systems are also lacking. Leap motion controller facilitates the real-time and accurate recognition of ASL signs. It allows an opportunity for designing a learning application with a real-time sign recognition system that seeks to improve the effectiveness of ASL learning. The project proposes an ASL learning application prototype. The application would be a whack-a-mole game with a real-time sign recognition system embedded. Since both static and dynamic signs (J, Z) exist in ASL alphabets, Long-Short Term Memory Recurrent Neural Network with k-Nearest-Neighbour method is adopted as the classification method is based on handling of sequences of input. Characteristics such as sphere radius, angles between fingers and distance between finger positions are extracted as input for the classification model. The model is trained with 2600 samples, 100 samples taken for each alphabet. The experimental results revealed that the recognition rate for 26 ASL alphabets yields an average of 99.44% accuracy rate and 91.82% in 5-fold cross-validation with the use of leap motion controller.

1. Introduction

1.1. Problem description

Sign languages are natural languages that have been developed through the evolution of contact between the hearing impaired but not invented by any system (Napier & Leeson, 2016). They differ from spoken languages in primarily two ways. First, sign languages are natural and mature languages are “articulated in visual-spatial modality”, unlike spoken ones, that are presented in “oral-aural modality”. Second, Napier and Leeson (2016) pointed out that sign languages employ two hands, facial muscles, the body and head and sometimes also involve vocalisation. They are neither universal nor mutually intelligible (Beal-Alvarez, 2014). In other words, a sign language that is developed in one region is not applicable in other regions and contains non-relevant varieties that require special methods/techniques of acquisition. Currently, 141 types of sign languages exist worldwide (Liddell & Johnson, 1989).

The American sign language (ASL) is the foremost used language for the deaf in the United States and English-speaking regions of Canada (Napier, Leigh, & Nann, 2007). Though increasing recognition for ASL has boosted confidence among the hearing impaired, the limited resources available has created social and cultural issues among the hearing impaired communities, compared to the amount of linguistics research despite the amount of linguistic research carried out in the field (Marschark & Spencer, 2010). In the United States, hearing impaired and hard-of-hearing students can choose between attending residential (catering to only students who are hearing impaired or hard-of-hearing) or public schools. As the integration of hearing impaired with peers without hearing impairment is emphasised, an increasing number of hearing impaired students are enrolling in public schools. However, they are placed in environments without adequate teaching support in most cases (Marschark & Spencer, 2010).

To create an inclusive environment with hearing students and hearing impaired in public schools, promoting ASL among the hearing public would be effective. With the implementation of ASL in schools,

* Corresponding author.
E-mail addresses: ckm.lee@polyu.edu.hk (C.K.M. Lee), kam.kh.ng@polyu.edu.hk (K.K.H. Ng), MCHchen@ntu.edu.sg (C.-H. Chen), H.Lau@westernsydney.edu.au (H.C.W. Lau), grace.sy.chung@polyu.edu.hk (S.Y. Chung), tiffanytsy.tsoi@connect.polyu.hk (T. Tsoi).

https://doi.org/10.1016/j.eswa.2020.114403
Received 22 October 2019; Received in revised form 3 August 2020; Accepted 27 November 2020
Available online 3 December 2020

hearing teachers and students can communicate through both linguistic and non-linguistics ways that can aid in creating an interactive environment for hearing impaired and hard-of-hearing students and thus enhance the effectiveness of academic learning. Furthermore, the promotion of ASL helps achieve the inclusion of the hearing impaired in society through boosting learning motivation with educational applications. Being a feasible and economical solution, the leap motion controller is commonly used as a device for sign recognition systems (Arsalan, Kim, Owais, & Park, 2020; Elboushaki, Hannane, Afdel, & Koutti, 2020). However, there exists a research gap on the adoption of leap motion controller in sign education purposes. A predominant section of the research only examines the viability of different sign recognition models with the leap motion controller and does not extend the model into an educational application that aids sign language learning and promotes sign languages. Only Parreño, Celi, Quevedo, Rivas, and Andaluz (2017) have proposed a didactic game prototype for Ecuadorian signs. Therefore, there is a paucity of research focusing on the development of educational applications for ASL with the leap motion controller and investigating the effectiveness of such applications in improving sign learning.

1.2. Contributions of the research

This research seeks to design an ASL learning application in game-learning and develop a real-time sign recognition system with leap motion controller for the use of the application. The sign recognition environment starts with identifying and extracting ASL's sign features and by subsequently developing a suitable algorithm for the recognition system. After applying the algorithm and training the network architecture, the system gains the capacity to recognise and classify ASL signs into 26 alphabets. The classification using feature extraction was processed by long-short term memory recurrent neural network (LSTM-RNN) with k-nearest neighbour (kNN) method. Finally, the system will be integrated into the game environment in the ASL learning application. This application is expected to promote ASL among the hearing impaired and the non-hearing impaired, thereby motivating them to learn ASL by entertainment and engagement provided by the game environment and further helping the hearing impaired to better integrate into society. Furthermore, it encourages and promotes the use of ASL as a second language that is worthy of acquiring.

The contributions of the research can be summarised as follows:

- The proposed LSTM-RNN with kNN method could recognise 26 alphabets with a recognition rate of 99.44% accuracy and 91.82% in 5-fold cross-validation using leap motion controller. The proposed method outperforms other well-known algorithms in the literature.
- Leap motion controller is a monochromatic-IR-cameras and three-infrared-LEDs based sensor to track the 3D motion of hand gesture, including Palm centre, fingertip position, sphere radius and finger bone positions for every 200 frames collected. Given that those data are available using a leap motion controller, we could further extract the feature for the classification of ASL, which an application in our study.
- The programming flow of the proposed model was designed as a learning-based program. A game module and recognition module are performed in real-time. We aim at promoting ASL in a learning-based environment as our application.

1.3. The organisation of the paper

The rest of this article is organised as follows. Section 2 describes the literature review and Section 3 illustrates the proposed framework for the ASL learning application, including the game module and real-time recognition system. Section 4 presents the validation results and analyses the performance of the proposed recognition system. Section 5 summarises the research, including the conclusion, research contributions, limitations and future development.

2. Literature review

2.1. Learning application

In terms of educational technology, knowledge acquisition in students can be improved through the fusion of academic activities with interactive, collaborative and immersive technologies (Souza, 2015). Notably, several studies have proposed new approaches that stimulate sign language mastering and knowledge acquisition by promoting motivation and excitement in pedagogical activities. Parreño et al. (2017) suggested that an intelligent sign learning game-based system is more effective in the improvement of sign language skills. Pontes, Duarte, and Pinheiro (2018) have also proposed an education digital game with the provision of a modular software architecture that acts as a motivator in the Brazilian Sign Language learning process. Notably, modular software architectures can allow adjustments to accommodate other sign languages (Rastgoo, Kiani, & Escalera, 2020). Furthermore, it is suggested that engagement is ensured when students concentrate and enjoy sign learning via the game, which eventually improves learning performance among students (Kamnardsiri, Hongsit, Khuwuthyakorn, & Wongta, 2017). In summary, educational games are proven to be effective tools in learning sign languages and are further supported by the engagement, motivation and entertainment they warrant.

2.2. The comparison of sign recognition methods

Past research has suggested several methods for the recognition of ASL, including the usage of motion gloves, Kinect Sensor, image processing with cameras and leap motion controllers. Oz and Leu (2011) developed an artificial neural network model to track the 3D motion for 50 ASL words. Motion gloves for ASL recognition are more expensive, have higher restrictions in terms of hand anatomy and are less comfortable for users compared to vision-based methods. Moreover, it is time-consuming and may result in imprecise calibrations caused by the wear and tear from repeated use of the gloves (Huenerfauth & Lu, 2010; Luzanin & Plancak, 2014; Oz & Leu, 2007). Due to sign complexities, constant finger occlusions, high interclass similarities and significant interclass variations, the recognition of ASL signs is still remains a challenging task for Kinect sensors used in isolation (Sun, Zhang, Bao, Xu, & Mei, 2013; Tao, Leu, & Yin, 2018). Furthermore, the calibration of the sensory data are is also important. Several studies have focused on the measurement of angular positions to predict the motion gestures (Fujiwara, Santos, & Suzuki, 2014). Tubaiz, Shanableh, and Assaleh (2015) and Aly, Aly, and Almotairi (2019) suggested that an ASL recognition system could can be developed in a user-dependent mode and proposed a modified kNN approach. Readers can refer to the review article on sensory gloves for sign language recognition. The sensing board and wearable application for ASL recognition have been also been extensively studied in the literature (Lee & Lee, 2018; Paudyal, Lee, Banerjee, & Gupta, 2019; Wu & Jafari, 2017; Wu, Sun, & Jafari, 2016; Wu, Tian, Sun, Estevez, & Jafari, 2015).

Among all vision-based sign recognition methods, image processing is a low-cost, widely accessible and effective option (Ciaramello & Hemami, 2011; Starner, Weaver, & Pentland, 1998); however, it requires a long calculation to recognise hand and fingers, which results in a long interval before projecting the recognition result (Khelil et al., 2016). Furthermore, skin colour and lightning conditions are critical factors that severely affect and hinder data accuracy (Bheda & Radpour, 2017). However, the leap motion controller in palm-size is a more economical and portable solution than motion gloves or Kinect sensors discussed above (Chuan, Regina, & Guardino, 2014). Fast processing, robustness and requirement of less memory are additional advantages for the leap motion controller (Naglot & Kulkarni, 2016). However, the controller has an inconsistent sampling frequency. It requires postprocessing to reduce its effect on real-time recognition systems (Guna, Jakus, Pogačnik, Tomažič, & Sodnik, 2014). The comparison of glove-based and vision-based methods of gesture recognition application are shown in Table 1.

2.3. Structure and recognition framework of leap motion controller

The controller, comprised of infrared cameras and optical sensors, is used for sensing hand and finger movements in 3D space. According to the sensor's coordinate system, the position and speed of the palm and fingers can be recognised with infrared imaging (Khelil et al., 2016). The controller employs a right-handed Cartesian coordinate system, which has the XYZ axes intersecting in the centre of the sensor as shown in Fig. 1. The controller can be programmed through the leap motion application programming interface (API). Positioning and speed data are mentioned above and can be obtained through API.

General sign recognition system with leap motion controller consists of the following essential steps: data acquisition, feature extraction, classification and validation. Basically, a general recognition would start with a sign recognised by the leap motion controller and then the data is sent for pre-processing. In the stage of data acquisition, hand palm data and finger data can be acquired from the API. For the feature extraction, different studies have defined and extracted features for sign recognition that proposed numerous methods to compute feature vectors for further processing (Chong & Lee, 2018; Chuan et al., 2014; Khelil et al., 2016). Furthermore, the classification and validation techniques used in the literature on sign recognition systems with leap motion Controller are compared and the results are shown in Table 2.

It is observed that the support vector machine (SVM) has been a popular classification method used over the years in sign recognition systems with leap motion, and the use of neural network would be a newer classification method (H. Lee, Li, Rai, & Chattopadhyay, 2020; Valente & Maldonado, 2020). Moreover, different types of cross-validation techniques are used in model validation as well. Neural network, also called deep neural network (DNN), is a type of deep learning and is commonly used for classification or regression with success in different areas (Akyol, 2020; Zhong et al., 2020). The predominant reason for neural networks outperforming SVMs is the former's ability to learn important features from any data structure and to handle multiclass classification with a single neural network structure (Rojas, 1996). Artificial neural network is the most commonly used type of neural network while recurrent neural network (RNN) is one of its categories, whose connections between nodes would form a directed graph along temporal sequences (Asghari, Leung, & Hsu, 2020; Jeong et al., 2019; Liu, Yu, Yu, Chen, & Wu, 2020; Rojas, 1996). It demonstrates a temporal dynamic behaviour that implies the function is time dependent. However, classic RNN is not able to handle a long-time frame. long-short term memory (LSTM) is a special type of RNN that addresses the limitations of classic RNN (Hochreiter & Schmidhuber, 1997). LSTM is effective in learning long-term dependencies. It is suggested that constant error backpropagation within internal states contributes to its ability to bridge long time lags (Hochreiter & Schmidhuber, 1997). Noise, continuous values and distributed representations can be handled effectively by LSTM.

Table 1
Comparison between glove-based and vision-based methods.

Factors	Motion Gloves	Vision-based Methods
User comfort	Less	High
Portability	Lower	Higher
Cost	Higher	Lower
Hand Anatomy	Low	High
Calibration	Critical	Not Critical

3. Methodology

The system conceptual framework is shown in Fig. 2 and consists of two running modules - game module and the real-time sign recognition system. The proposed learning application is fundamentally, a special Whack-A-Mole game. Rather than mouse-clicking, a question pertaining to ASL signs has to be accurately answered in order to strike the mole. Each mole would come up from 7 holes randomly holding a stick, on which 1 out of the 26 English alphabets is randomly printed. In the meantime, the appropriate hand configuration for the corresponding ASL alphabet is shown on the upper left-hand corner as a hint. Users have to make the ASL sign through the leap motion controller. Subsequently, real-time sign recognition also occurs. The real-time sign recognition system is comprised of three phases: data acquisition, feature extraction and classification. First, data acquisition happens with data that is directly extracted from the leap motion API. Next, some data has to be further processed as features. Following this, the structured data can be input into the pre-trained classification model for real-time recognition. Gestures would be classified into 1 of the 26 classes. If the classification result matches with what is on the stick, accuracy is shown on the game interface. The mole would be struck and a point would be added only if the accuracy rate is 80 or above. Otherwise, a miss would be recorded. The time limit for each question would be half a minute and each trial of the game ends after 5 questions, which means that the steps in the conceptual framework are gone through 5 times. Fig. 3 illustrates a scene in the game when a question is answered correctly through the leap motion controller.

The designed programming flow is shown in Fig. 4 and primarily consists of two scripts running synchronously, i.e. Real-time Recorder and Gaming. When the application is initialised, the Real-time Recorder first creates a file in the CSV format and initialises the real-time listener. The real-time listener continuously collects data from the leap motion API. The sign language includes state and dynamic signs. Furthermore, leap motion is sensitive to hand gesture motion and slightly motion change may be captured. Therefore, 30 features extraction for 200 frames are considered to accommodate the hand gesture motion change and the nature of state and dynamic signs of ASL. For every 200 frames collected, they are passed to RNN classifier for classification. The classification results would be sent back to the Real-time Recorder for saving into the CSV file. On the other hand, Game is synchronously running. When a mole comes up, it would continuously take the latest classification result from the CSV file to determine whether the mole would be hit and to show the accuracy score.

3.1. Data acquisition for ASL recognition using leap motion controller

A general recognition would start with a sign recognised by the leap motion controller; subsequently, data is sent for pre-processing. Hand palm data, hand sphere radius and finger data are acquired. This is demonstrated in Fig. 5.

Hand palm data includes unit vector of palm, position of palm centre, velocity of palm and palm normal (Naidu & Ghotkar, 2016). In the meantime, hand palm sphere radius, grab strength and pinch strength can be obtained. Hand palm sphere radius measures a sphere that matches the curvature of the hand. The line connecting the red dots in Fig. 6. illustrates the diameter of the sphere and hence, half of it would be the radius. The grab strength refers to the strength of showing a grab hand pose; for it, the value 0 represents an open hand and the value 1 represents a grab hand pose. Similarly, pinch strength lies between 0 and 1, where 0 means an open hand detected and 1 means pinch hand pose recognised. Pinching can be done with the thumb and any other finger.

The finger data carries the direction and length of each finger, tip velocity and position of joints as stated in Fig. 7. Other than fingertip positions, the positions of joints between the distal bones, intermediate bones, proximal bones and metacarpal bones can be obtained (Khelil et al., 2016).

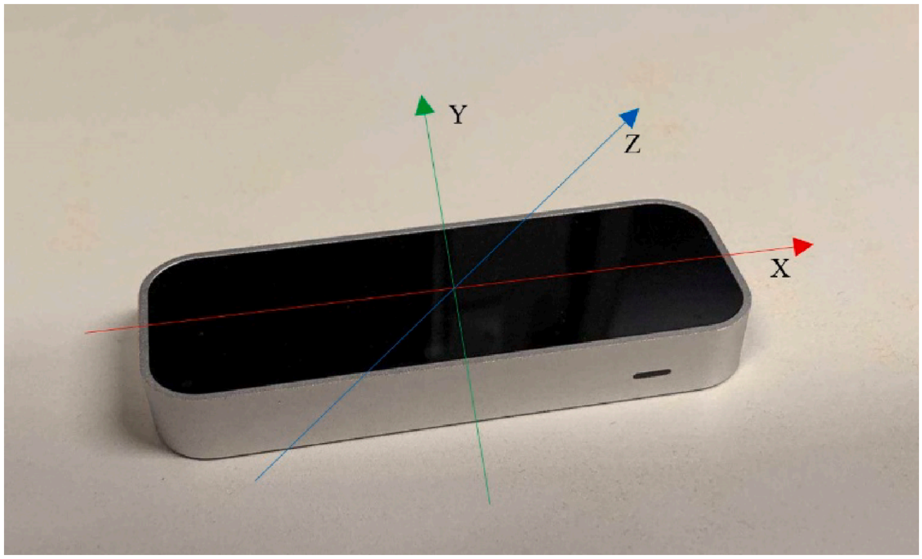

Fig. 1. Orientation of leap motion controller.

Table 2
Comparison of sign recognition systems with leap motion controller on classification and validation techniques.

Ref.	Number of gestures	Classifier	Validation	Accuracy (%)
(Avola et al., 2018)	30 ASL gestures (12 dynamic signs and 18 static signs)	RNN	Not mentioned	96.41
Chong & Lee (2018)	26 ASL gestures (A-Z) 36 ASL gestures (A-Z, 0–9)	DNN	Leave-one-subject-out cross-validation	93.81 88.79
Chuan et al. ((2014)	26 ASL gestures (A-Z)	SVM	4 fold cross-validation	79.83
Du, Liu, Feng, Chen, and Wu (2017)	10 selected gestures	SVM	80% training set and 20% testing set	83.36
Khelil et al. (2016)	10 ASL gestures (0–9)	SVM	Not mentioned	91.30

We referred to the feature extraction methods for leap motion controller proposed by Chong and Lee (2018). The following features extracted are used to describe palm flexion, hand movement, relation of palm and fingertips, as well as the relation between fingertips.

The standard deviation of palm position (S) can be calculated using (1), where P represents the position of the palm centre and N denotes the size of the dataset.

$$S = \sqrt{\frac{1}{n-1}\left(\sum_{i=1}^{n}(P_i - P)^2\right)} \quad (1)$$

Palm sphere radius (R) can be computed as shown in Eq. (2), where F_i represents the positions of the fingertips,$i \in \{1,2,3,4,5\}$ represent thumb, index, middle, ring and little fingers respectively.

$$D_{FP} = \sqrt{(F_{ix} - P_x)^2 + (F_{iy} - P_y)^2 + (F_{iz} - P_z)^2} \quad (2)$$

The angles between 2 adjacent fingers (A) can be calculated with Eq. (3). Note that the angle between the thumb and the little finger is excluded due to the inclusiveness of palm curliness, which is included in R.

$$Aa_{Fi} = \frac{F_i - F_{i+1}}{\pi} \quad (3)$$

Distance between all the fingers (L), with 2 in a group in a total of 10 groups, is computed according to (4). i and j represents all fingertips 1 to 5, while $i \neq j$.

$$L_{FF} = \sqrt{(F_{ix} - F_{jx})^2 + (F_{iy} - F_{jy})^2 + (F_{iz} - F_{jz})^2} \quad (4)$$

3.2. Training of sign recognition model by feature extraction

Real-time sign recognition requires a pre-trained classification model. First, the data samples should be taken as input for the training of the model. Thus, model training would commence by collecting raw data from the leap motion API. Since ASL signs are featured by relative positions and angles between the palm and fingers, both palm and finger data are vital. Thus, data in Table 3 was collected for the proposed work. The front and rear views of ASL on leap motion are presented in Figs. 8 and 9, respectively. Fig. 10

For feature extraction, some raw data was directly used as features such as (a), (c), (d) and (e). The others were further processed into features. Finally, 30 features were generated, as shown in Table 4. A total of 2600 data samples were collected, among which 100 samples for each of the 26 alphabets were collected for training the model. Each sample is constituted of 200 frames of these 30 features. Only right-hand samples were collected. Since the frame rate varies based on the different computing resources and activities performed, 110 frames were collected in this work in a second with the computing resources and the environment by approximation. Subsequently, the 2600 data samples were piled into a file in npy format of sizes of (2600, 200, 30). A set of labels was also created for identifying data samples' classes. This is an npy file of size (2600, 26).

First, the LSTM layer is selected due to its capability for handling data in a long-time frame that is constituted of 28 neurons. For the algorithmic structure of LSTM, the readers can refer to the work by Goyal, Pandey, and Jain (2018). Three parameters are to be determined: batch size, number of epochs and units for LSTM. Batch size refers to number of samples for training each time. Apparently, larger batch size results in a model with lower accuracy while smaller batch size requires much more training time which would not be efficient enough. Number of epochs represents number of passes over the entire dataset. After each epoch, evaluation is made and weights in neural network are updated. With more epochs trained, the model should be more accurate. However, model with too many epochs trained would appear to be overfitting. Overfitting appears when the model predicts data in an unnecessarily complicated way. In other words, it fits known data well

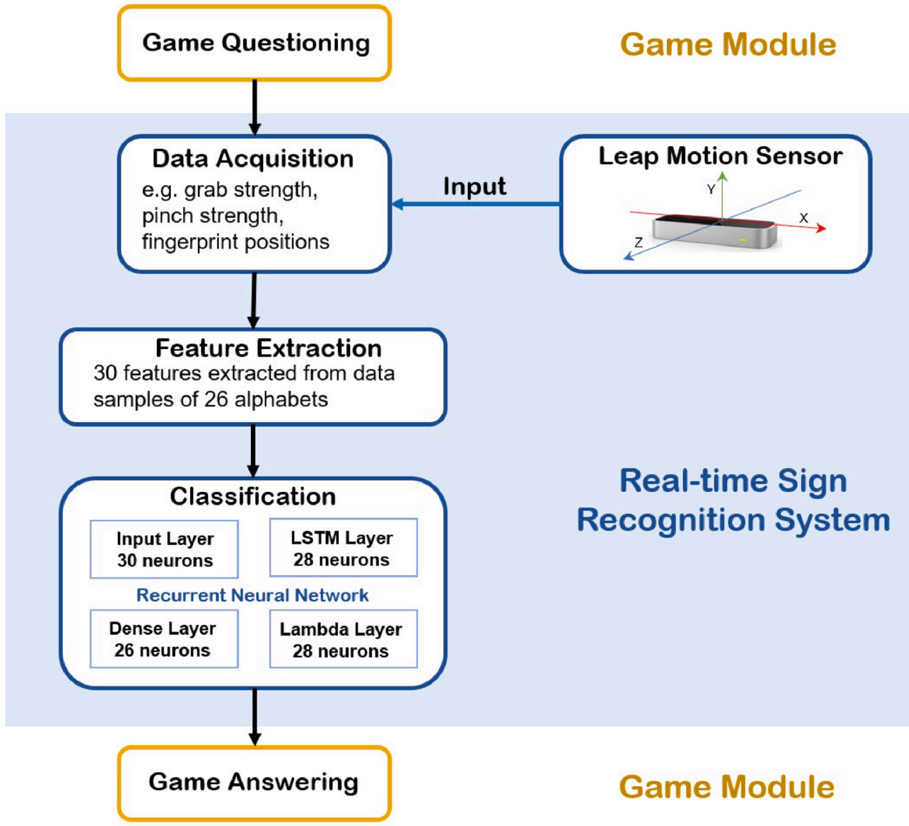

Fig. 2. Conceptual framework of the game modular based ASL recognition.

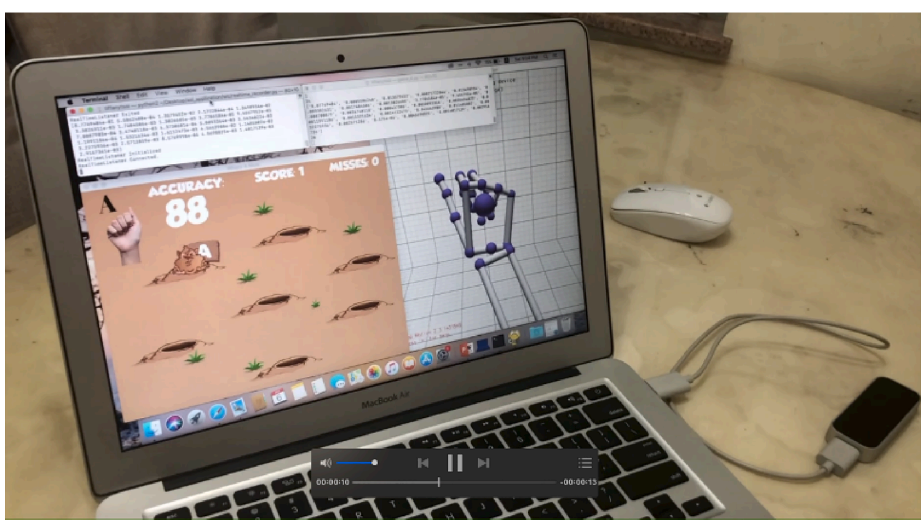

Fig. 3. Questions answered correctly in the application developed.

yet is less successful in fitting subsequent data than a simpler model. For units in LSTM, it refers to the dimensionality of LSTM output space. It can also be seen as number of neurons in the layer. It is hard to determine whether larger or smaller size of units would be better. Every model with different features is optimized by differing number of units.

The final step before model training would be the selection of model parameters. Three parameters are to be determined: batch size, number of epochs and number of units for LSTM. Batch size refers to the number of samples for training each time, whereas the number of epochs represents the number of passes over the entire dataset. For units in LSTM, it refers to the dimensionality of LSTM output space. It can also be considered as the number of neurons in the layer. To determine the most effective parameters, "gridsearchCV" function from "sklearn" library in Python was used. It is observed that the units of LSTM, batch size and number of epochs are selected between 28 and 30, 32 and 64, 30 and 40 respectively. Table 5 shows a model grid created after applying function (5).

$$y_k(x) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{5}$$

It is illustrated that units of 28, batch size of 32 and number of epochs of 40 would be the best parameters optimising model performance.

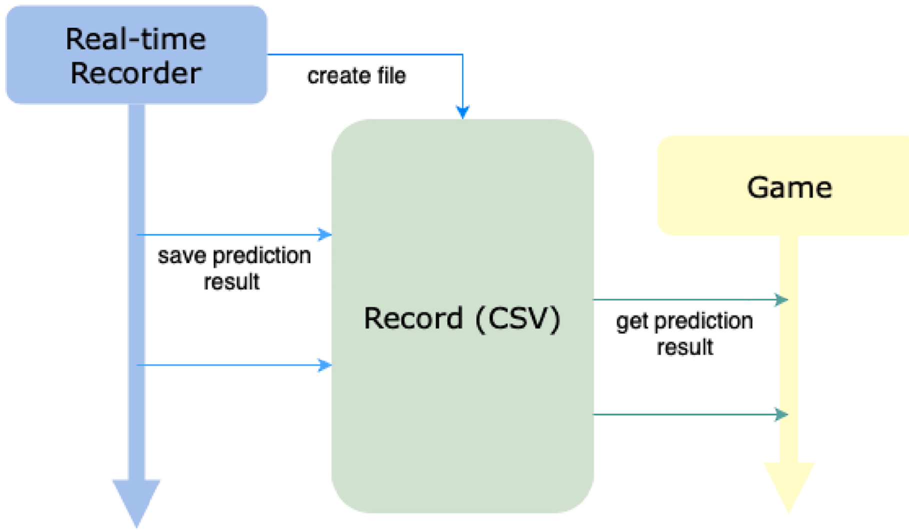

Fig. 4. Designed programming flow.

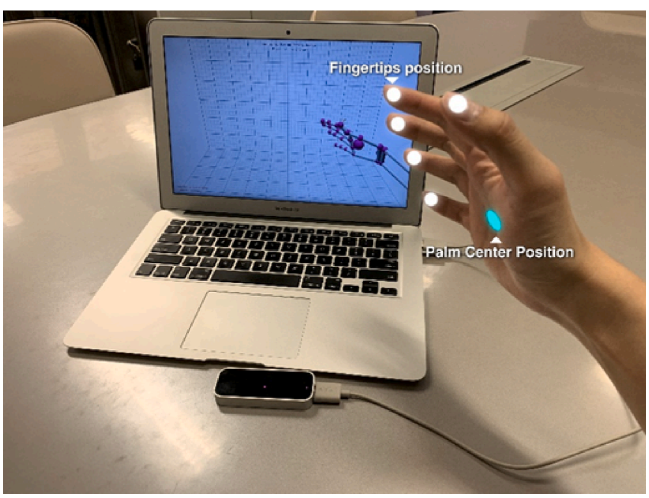

Fig. 5. Palm centre and fingertip position.

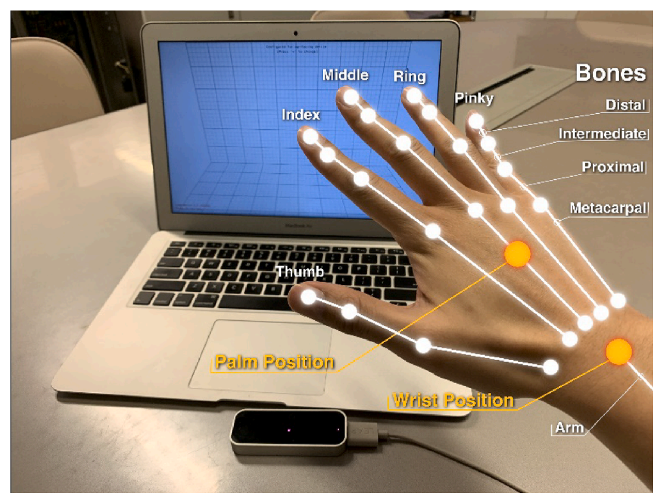

Fig. 7. Finger bone positions.

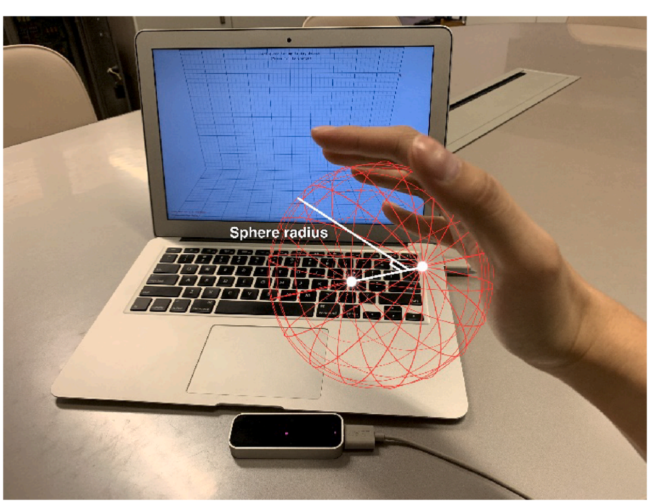

Fig. 6. Sphere radius.

Table 3
Data extracted in proposed work.

Data	Details
(a) Position of palm centre	X, Y and Z coordinates of the palm centre are extracted as 3 separate data.
(b) Unit vector of palm normal	A vector pointing perpendicular to the palm direction
(c) Sphere radius	The radius of the sphere that matches curvature of a hand
(d) Grab strength	Strength of being a grab hand pose [0,1]
(e) Pinch strength	Strength of being a pinch hand pose [0,1]
(f) Fingertip positions	Positions of thumb, index, middle, ring and little fingertips are extracted in radian.
(g) Fingertip directions	Directions of thumb, index, middle, ring and little fingertips are extracted in radian.

Hence, epochs of 80 times are selected for the final model to improve the accuracy. The selected model parameters were also input. Finally, the model is trained and was output in h5 format for use in real-time sign recognition.

After selecting the above parameters, the loss function should be selected for compiling the model to optimise its performance. Categorical cross-entropy, a multi-class logarithmic loss, is selected. For the proposed model, it was created based on the training set. Categorical

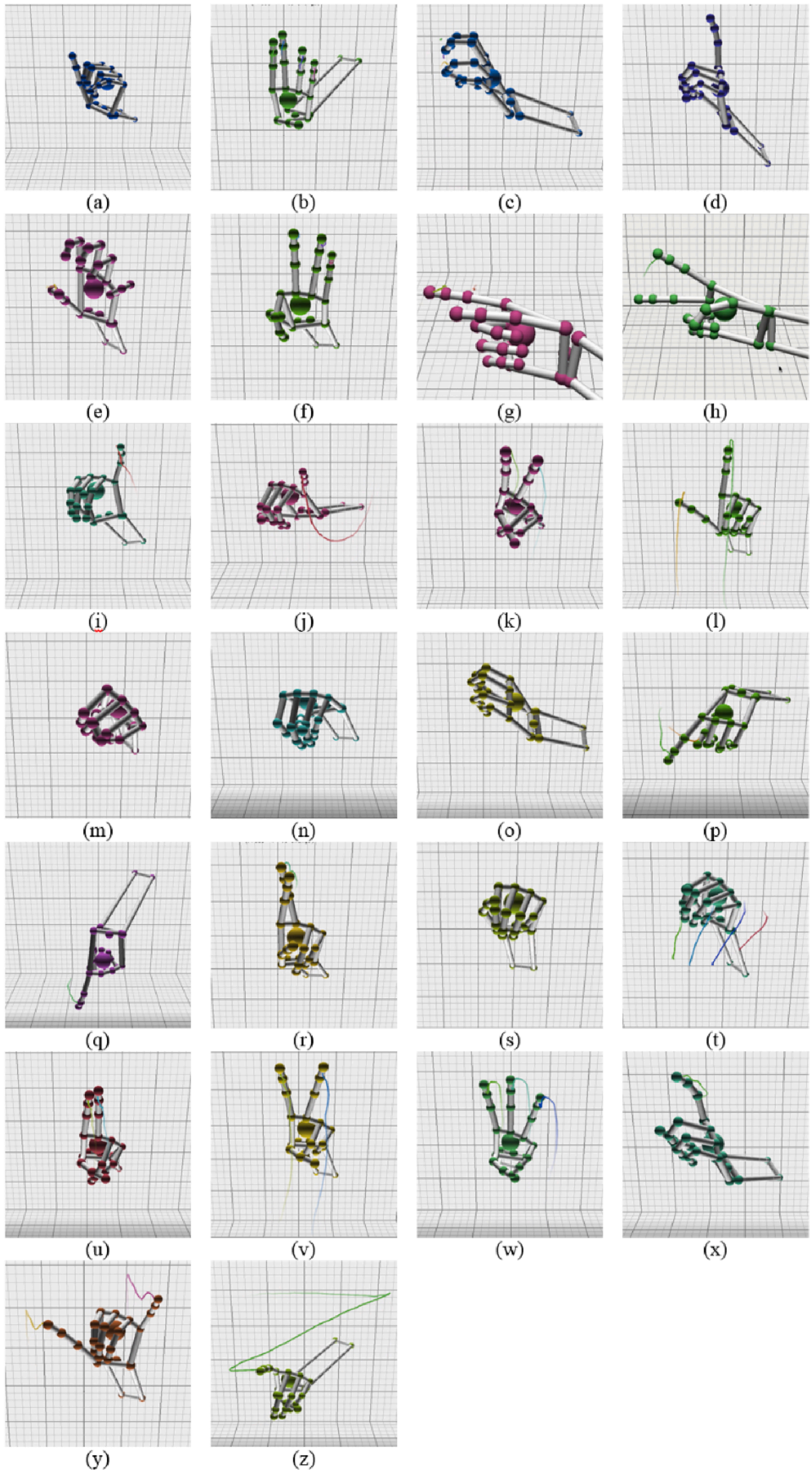

Fig. 8. Front view of American sign language on leap motion.

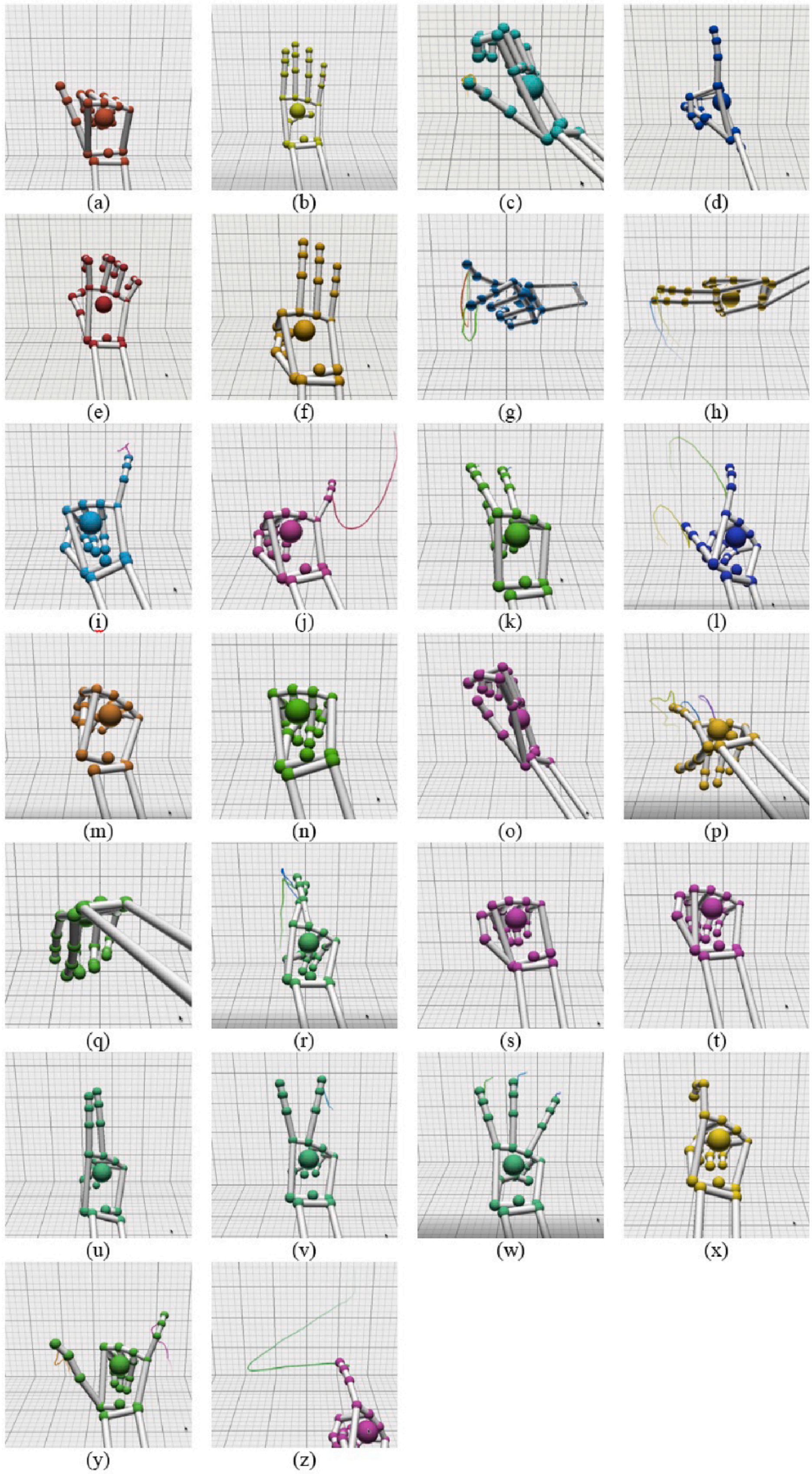

Fig. 9. Rear view of American sign language on leap motion.

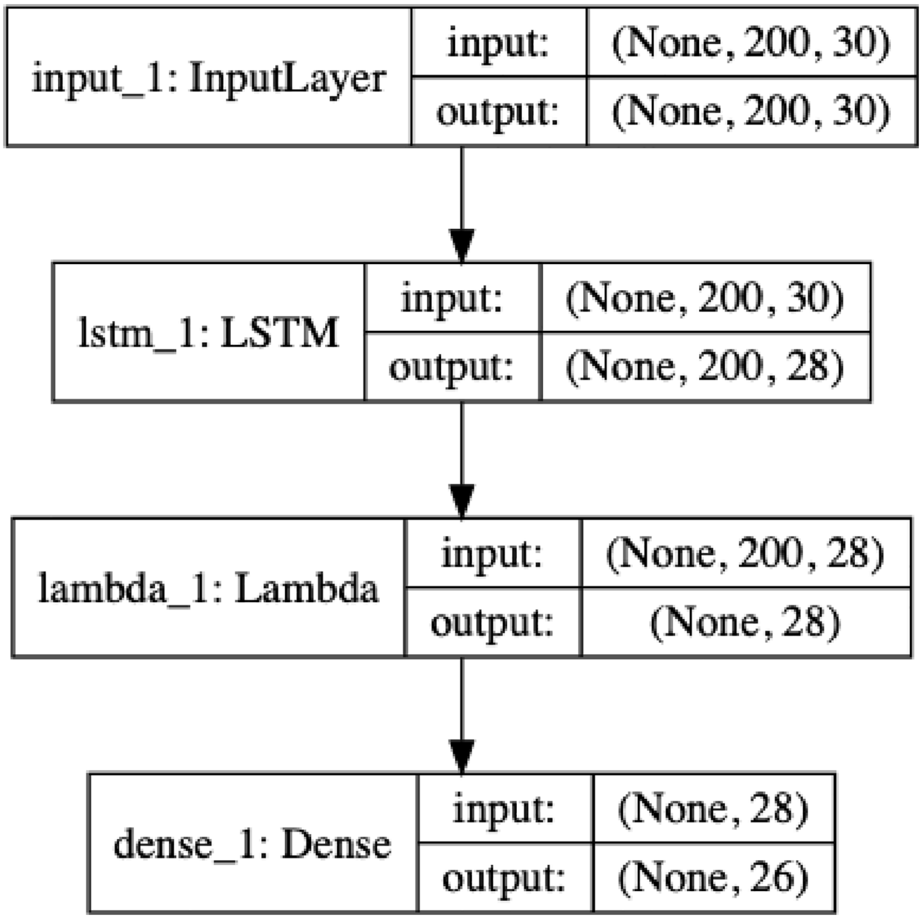

Fig. 10. Proposed classifier model.

Table 4
Features extracted for model training.

1	Palm centre position X	2	Palm centre position Y
3	Palm centre position Z	4	Sphere radius (mm)
5	Grab strength [0,1]	6	Pinch strength [0,1]
7	Distance between palm centre position and thumb tip position	8	Distance between palm centre position and index tip position
9	Distance between palm centre position and middle tip position	10	Distance between palm centre position and ring tip position
11	Distance between palm centre position and little tip position	12	The angle between thumb normal and thumb tip direction (radian)
13	The angle between thumb normal and index tip direction (radian)	14	The angle between thumb normal and middle tip direction (radian)
15	The angle between thumb normal and ring tip direction (radian)	16	Angle between thumb normal and little tip direction (radian)
17	Distance between thumb tip position and index tip position	18	Distance between thumb tip position and middle tip position
19	Distance between thumb tip position and ring tip position	20	Distance between thumb tip position and little tip position
21	Distance between index tip position and middle tip position	22	Distance between index tip position and ring tip position
23	Distance between index tip position and little tip position	24	Distance between middle tip position and ring tip position
25	Distance between middle tip position and little tip position	26	Distance between ring tip position and little tip position
27	The angle between thumb tip direction and index tip direction (radian)	28	The angle between index tip direction and middle tip direction (radian)
29	The angle between middle tip direction and ring tip direction (radian)	30	The angle between ring tip direction and little tip direction (radian)

The proposed model consists of 3 layers after the input layer as shown in Figure 10.

Table 5
Model grid for selection of parameters.

Number of epochs	Units: 28		Units: 30	
	Batch size		Batch size	
	32	64	32	64
30	0.094	0.098	0.091	0.077
40	0.135	0.100	0.120	0.101

cross-entropy was measured on the test set to evaluate the accuracy of the model in the predictions. Cross-entropy, used as an alternative to squared error, is an error measure intended for network with output representing independent hypotheses and node activations representing a probability of each hypothesis being true. In the case, output vector is a probability distribution and cross-entropy is used as an indication of distance between what the network predicts for the result of the distribution and the "actual answer" for the distribution. The equation for categorical cross-entropy in Keras is suggested below (Gulli & Pal, 2017).

$$L_i = -\sum_j T_{ij} \log(P_{ij}) \tag{6}$$

where T_{ij} is the target and P_{ij} refers to the prediction.

Another parameter to be selected in compiling the model is the optimiser. The selected optimiser, Adam, is a gradient-based optimisation of stochastic objective functions. It functions on the basis of lower-order moment estimation. It is different from classical ones by maintaining a single learning rate for all weight adjustments during the entire training process (Kingma & Ba, 2014). However, the method adapts different learning rates for different parameter selections by estimation of first and second moments of gradient. Kingma and Ba (2014) also suggested that Adam combines the advantages of Adaptive Gradient Algorithm and Root Mean Square Propagation. Adaptive Gradient Algorithm is great in handling sparse gradient problems while Root Mean Square Propagation does well on non-stationary problems. Adam possesses both of the advantages. Adam is the most appropriate choice of optimiser for the proposed model due to the following reasons. It is computationally efficient and hence has a low memory requirement. It is well-designed for handling problems with large amounts of data. Finally, it is capable of managing dynamic objectives as well as problems with lots of noise.

Besides, the Lambda layer in the middle would be a K-means clustering layer. The algorithm proposed by Vassilvitskii (2007) would assign N data points into 1 of the K clusters. The pseudo-code of the K-mean clustering algorithm is shown in (Vassilvitskii, 2007). K-mean clustering is opted for the second layer since it is an efficient clustering method for handling multi-class classification. With supervised and unsupervised learning in the same model, the model would optimise advantages from both sides. Furthermore, the k-mean clustering compressed the 200 frames to obtain the centre point of feature extracted for model training mentioned in Table 4. This can accommodate different hand size and motion changes in the 200 frames, especially the relative coordinates between finger, distal, intermediate, proximal and metacarpal.

Third, the final layer before the output of the result would be a Dense layer, which is a classic fully connected layer. A Softmax function, which is logistic regression, is often used as the output function of the network. The log odd ratios calculated would be the probabilities of each class in multiclass classification. The Dense layer is selected as the final layer to transform group predictions into class probabilities for output.

Algorithm 1. *Algorithm of k-mean clustering* (Aly et al., 2019)

1. Randomly chose k initial centres $C = \{c_1, \cdots, c_k\}$
2. repeat
3. For each $i \in \{1, \cdots, k\}$ set C_i to be the set of points in X that are closer to c_i
4. than c_j for any $j \neq i$. {Assignment Step}
5. For each $i \in \{1, \cdots, k\}$ set $c_i = \frac{1}{|C_i|}\sum_{x \in C_i} x$ {Means Step}
6. Until C does not change

3.3. Model validation

Cross-validation, a method that separates the dataset into S folds, is selected. Since data in the proposed model is neither scarce nor expensive in extraction, general 5-fold cross-validation was used. 80% and 20% of the dataset would be used for training and validating respectively in each trial (Refaeilzadeh, Tang, & Liu, 2009). First, the data set was divided into 5 groups (folds), and a total of 5 trials is conducted. For each trial, one of the folds was assigned as the testing set, while the rest were assigned as the training sets. Subsequently, the model was trained with the training sets and validation took place in the testing set. For validation in each trial, the overall accuracy and a confusion matrix for 26 classes were extracted. The 26-class confusion matrix is further produced into another matrix containing true positive (TP), true negative (TN), false positive (FP) and false negative (FN), as explained in Table 7.

TP, TN, FP and FN calculated for each class can be used for generating accuracy (ACC), sensitivity (Se) and specificity (Sp) for each class. Accuracy refers to the ability of the model to correctly identify instances. Sensitivity is the proportion of "real" positives that are accurately identified as positives, while specificity is the proportion of "real" negatives that are correctly identified as negatives by the model. The equations of accuracy, sensitivity and specificity are expressed in terms of TP, TN, FP and FN as follows. TP, TN, FP and FN can also be used for generating the Matthews correlation coefficient (MCC) (Boughorbel, Jarray, & El-Anbari, 2017), Fowlkes-Mallows index (FM) (Campello, 2007) and Bookmaker informedness (BM) (Fluss, Faraggi, & Reiser, 2005) for proving each class statistical significance. MCC is used for measuring the observed and predicted binary classification (Boughorbel et al., 2017). FM is used for measuring the similarity between the observed and predicted binary classification (Campello, 2007). BM is used for estimating the probability of an informed decision (Fluss et al., 2005).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Se = \frac{TP}{TP + FN} \tag{8}$$

$$Sp = \frac{TN}{TN + FP} \tag{9}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{10}$$

$$FM = \sqrt{\frac{TP}{TP + FP} \times \frac{TP}{TP + FN}} \tag{11}$$

$$BM = Se + Sp - 1 \tag{12}$$

3.4. Dataset and experimental environment

Since there are no public datasets available for ASL training under a gaming environment, we recruited 100 participants to train the algorithms. 63 females and 37 males are recruited aged 20–30 years, and all participants declared that they are right-handed people. The dataset composed of 26 alphabet data and 100 sample size for each alphabet from 100 participants. Therefore, 2600 sample size for 26 alphabet data are obtained. As the gaming environment targets for ASL learning, the 100 participants do not have any formal training of ASL before. Before the data collection, an ASL experienced person will present the right ASL hand gesture to the participant several time. If the participants can present the right ASL hand gesture for 26 alphabets after the learning stage, the participants will present their ASL and the leap motion will collect their hand gesture data at the same time.

4. Results and discussion

With cross-validation, the comprehensive performance of the model can be evaluated before the output as the real-time sign recognition module of the game. In this session, 5-fold cross-validation was performed and the overall accuracy of the model is estimated to be 91.8%, averaging the 5 trials. The result is shown in Table 6.

Meanwhile, 26-class confusion matrices for the 5 trials were generated and were further transformed into matrices of TP, TN, FP and FN. Accuracy, sensitivity and specificity were calculated as a result. To accurately analyse the results, an average of over 5 trials were taken for accuracy, sensitivity and specificity for each alphabet as shown in Table 7. Per-class accuracy and specificity for the model were calculated to be over 98%, which implies that the model has a high probability incorrectly identifying negative results in each of the 26 classes; the proportion of accurately identified instances would be high as a result. Sensitivity only attains over 80%, except for the alphabet signs for M, N and S. It shows that the model has relatively poor chances of identifying

Table 6
Model accuracy.

Accuracy (%)	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	AVG	STDEV
	92.88	92.88	91.15	90.96	91.15	91.80	0.99

Table 7
Average accuracy, sensitivity and specificity for 26 classes.

Class	Proposed method			LSTM			SVM			RNN		
	ACC	Se	Sp	ACC	Se	Sp	ACC	Se	Sp	ACC	Se	Sp
A	99.92%	100.00%	99.92%	97.96%	83.00%	98.56%	98.35%	80.00%	99.08%	98.19%	84.00%	98.76%
B	99.96%	100.00%	99.96%	97.96%	64.00%	99.32%	97.42%	78.00%	98.20%	97.12%	66.00%	98.36%
C	99.73%	95.00%	99.92%	97.85%	88.00%	98.24%	97.46%	62.00%	98.88%	97.08%	64.00%	98.40%
D	99.42%	89.00%	99.84%	97.85%	56.00%	99.52%	97.58%	76.00%	98.44%	96.96%	70.00%	98.04%
E	99.85%	99.00%	99.88%	97.73%	85.00%	98.24%	95.73%	63.00%	97.04%	95.19%	69.00%	96.24%
F	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	96.38%	58.00%	97.92%	96.00%	40.00%	98.24%
G	99.77%	100.00%	99.76%	97.73%	56.00%	99.40%	96.69%	40.00%	98.96%	96.62%	56.00%	98.24%
H	99.81%	98.00%	99.88%	97.85%	83.00%	98.44%	97.54%	76.00%	98.40%	96.19%	45.00%	98.24%
I	99.89%	97.00%	100.00%	97.85%	61.00%	99.32%	96.96%	63.00%	98.32%	96.62%	56.00%	98.24%
J	99.89%	100.00%	99.88%	97.88%	81.00%	98.56%	97.04%	50.00%	98.92%	96.88%	56.00%	98.52%
K	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	96.00%	68.00%	97.12%	96.38%	57.00%	97.96%
L	99.39%	93.00%	99.64%	97.77%	88.00%	98.16%	97.00%	50.00%	98.88%	96.65%	48.00%	98.60%
M	98.35%	71.00%	99.44%	97.77%	54.00%	99.52%	96.77%	48.00%	98.72%	97.00%	49.00%	98.92%
N	98.08%	68.00%	99.28%	97.88%	64.00%	99.24%	97.15%	45.00%	99.24%	96.38%	38.00%	98.72%
O	99.27%	92.00%	99.56%	97.88%	83.00%	98.48%	97.31%	86.00%	97.76%	96.73%	73.00%	97.68%
P	99.85%	98.00%	99.92%	97.88%	62.00%	99.32%	97.35%	73.00%	98.32%	96.73%	57.00%	98.32%
Q	99.35%	88.00%	99.80%	97.85%	82.00%	98.48%	97.15%	48.00%	99.12%	97.19%	62.00%	98.60%
R	98.77%	69.00%	99.96%	97.85%	62.00%	99.28%	97.12%	80.00%	97.80%	96.19%	71.00%	97.20%
S	98.50%	78.00%	99.32%	97.88%	81.00%	98.56%	97.23%	55.00%	98.92%	96.38%	43.00%	98.52%
T	98.08%	87.00%	98.52%	97.88%	64.00%	99.24%	97.85%	65.00%	99.16%	96.58%	51.00%	98.40%
U	98.69%	98.00%	98.72%	98.04%	74.00%	99.00%	98.00%	80.00%	98.72%	96.81%	48.00%	98.76%
V	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	97.19%	67.00%	98.40%	96.27%	54.00%	97.96%
W	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	97.46%	59.00%	99.00%	96.35%	54.00%	98.04%
X	99.54%	99.00%	99.56%	98.04%	75.00%	98.96%	97.31%	62.00%	98.72%	98.00%	69.00%	99.16%
Y	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	97.27%	63.00%	98.64%	98.50%	79.00%	99.28%
Z	99.92%	99.00%	99.96%	100.00%	100.00%	100.00%	98.62%	68.00%	99.84%	98.69%	71.00%	99.80%
Avg	99.44%	93.00%	99.72%	98.36%	78.69%	99.15%	97.23%	63.96%	98.56%	96.83%	58.85%	98.35%
StdEv	0.64%	10.26%	0.39%	0.92%	15.76%	0.63%	0.62%	12.41%	0.64%	0.78%	12.10%	0.67%

positive results. We also compare the results with other well-known methods in ASL classification, including LSTM, SVM and RNN. Readers can refer to the algorithmic structures of LSTM (Avola, Bernardi, Cinque, Foresti, & Massaroni, 2019), SVM (Chong & Lee, 2018) and RNN (Avola et al., 2018). All the algorithms in the numerical experiments achieve better accuracy results in the class F, K, V, W and Y. The proposed method in predicting other classes outperforms LSTM, SVM and RNN. The average accuracy of the proposed method, LSTM, SVM and RNN is 99.44%, 98.36%, 97.23% and 96.83%, respectively. The proposed method obtained a fair good prediction statistically of the two-class classification, a greater similarity between the observed and predicted binary classifications and higher probability of estimating an informed decision comparing to LSTM, SVM and RNN. The statistical significance is introduced in Table 8. Therefore, we can conclude that the proposed method outperforms LSTM, SVM and RNN in the numerical analysis.

Table 8
Statistical Significance for 26 classes.

Class	Proposed method			LSTM			SVM			RNN		
	MCC	FM	BM	MCC	FM	BM	MCC	FM	BM	MCC	FM	BM
A	98.97%	99.00%	96.12%	75.05%	76.09%	81.56%	77.97%	78.83%	79.08%	77.41%	78.33%	82.76%
B	99.48%	99.50%	96.12%	70.09%	71.11%	63.32%	69.03%	70.33%	76.20%	62.31%	63.80%	64.36%
C	96.32%	96.46%	91.08%	75.55%	76.59%	86.24%	64.05%	65.35%	60.88%	61.24%	62.76%	62.40%
D	92.07%	92.37%	85.07%	66.90%	67.91%	55.52%	69.62%	70.87%	74.44%	62.61%	64.17%	68.04%
E	97.70%	97.78%	95.34%	73.72%	74.84%	83.24%	51.68%	53.82%	60.04%	51.76%	54.04%	65.24%
F	100.00%	100.00%	96.15%	100.00%	100.00%	100.00%	53.42%	55.30%	55.92%	41.59%	43.64%	38.24%
G	97.22%	97.33%	95.62%	65.37%	66.46%	55.40%	47.63%	49.24%	38.96%	54.24%	56.00%	54.24%
H	97.34%	97.44%	94.09%	74.06%	75.14%	81.44%	69.30%	70.56%	74.40%	45.73%	47.70%	43.24%
I	98.35%	98.41%	93.19%	68.00%	69.07%	60.32%	59.90%	61.48%	61.32%	54.24%	56.00%	54.24%
J	99.25%	99.32%	91.54%	73.80%	74.88%	79.56%	55.49%	56.98%	48.92%	56.46%	58.07%	54.52%
K	100.00%	100.00%	96.15%	100.00%	100.00%	100.00%	55.48%	57.47%	65.12%	52.97%	54.85%	54.96%
L	91.60%	91.92%	88.74%	74.94%	76.02%	86.16%	55.10%	56.61%	48.88%	50.98%	52.69%	46.60%
M	76.17%	77.01%	66.62%	65.44%	66.47%	53.52%	52.03%	53.67%	46.72%	54.71%	56.21%	47.92%
N	72.35%	73.33%	63.46%	69.18%	70.25%	63.24%	54.91%	56.25%	44.24%	43.63%	45.42%	36.72%
O	90.27%	90.65%	87.73%	74.39%	75.45%	81.48%	70.89%	72.17%	83.76%	62.14%	63.78%	70.68%
P	97.81%	97.89%	94.16%	68.70%	69.76%	61.32%	66.71%	68.07%	71.32%	55.59%	57.29%	55.32%
Q	90.83%	91.16%	83.88%	73.76%	74.86%	80.48%	55.98%	57.37%	47.12%	61.49%	62.95%	60.60%
R	81.93%	82.47%	65.12%	68.24%	69.32%	61.28%	67.43%	68.85%	77.80%	57.91%	59.79%	68.20%
S	79.25%	80.03%	73.50%	73.80%	74.88%	79.56%	59.33%	60.74%	53.92%	46.24%	48.08%	41.52%
T	76.98%	77.93%	81.64%	69.18%	70.25%	63.24%	68.99%	70.09%	64.16%	51.69%	53.46%	49.40%
U	86.21%	86.83%	92.76%	73.35%	74.37%	73.00%	74.56%	75.59%	78.72%	52.39%	54.00%	46.76%
V	100.00%	100.00%	96.15%	100.00%	100.00%	100.00%	63.31%	64.77%	65.40%	50.76%	52.70%	51.96%
W	100.00%	100.00%	96.15%	100.00%	100.00%	100.00%	63.08%	64.37%	58.00%	51.31%	53.21%	52.04%
X	93.77%	94.00%	94.92%	73.61%	74.63%	73.96%	62.55%	63.95%	60.72%	71.70%	72.73%	68.16%
Y	100.00%	100.00%	96.15%	100.00%	100.00%	100.00%	62.55%	63.97%	61.64%	79.43%	80.21%	78.28%
Z	99.03%	99.07%	94.88%	100.00%	100.00%	100.00%	79.51%	80.14%	67.84%	80.83%	81.44%	70.80%
Avg	92.80%	93.07%	88.71%	77.97%	78.78%	77.84%	62.71%	64.11%	62.52%	57.36%	58.97%	57.20%

On the other hand, model accuracy was assumed to be significantly below the expectation for 40 epochs of training and thus 80 times of training was selected. To evaluate the suitability of selection, a graph was plotted on model accuracy over epochs, as shown in Fig. 11. As can be observed, the accuracy of the model increases with the increased number of epochs and the graph for testing set eventually goes flat between the 70th and 80th epochs. Contrarily, model loss decreased significantly in the first 20 epochs and subsequently decreases in loss narrows but continues as shown in Fig. 12. The graph of loss for the testing set eventually goes flat just before 80th epoch. Thus, 80 epochs for training the model is shown to be optimising.

The proposed work using RNN with 26 alphabets is compared with other literature proposing sign recognition systems with leap motion controller. First, it is observed that the proposed work has generally stronger performance than those that previously employed SVM. Compared to other models that proposed employing the neural network, this undertaking has slightly higher accuracy. It specifically outperforms those that employed SVM as their classification method; this can probably be attributed to neural network's higher ability to handle large datasets.

In this research, we considered the leap motion controller for ASL recognition. Compared to image processing approach, the Leap Motion controller offers a quick hand gesture detection and captures the change of hand gestures in real-time with less computational power. Image processing using conventional cameras may require a high-level computer specification. In contrast, Leap Motion Control does not require a high-level computer specification, and most of the hand gestures and motion are detected using Infrared LEDs and cameras and output to the computer units for secondary processing. The primary restriction of leap motion controller is the exposed regions is from frog's eye view, as the leap motion controller must place on a surface. One may consider the integrated approach using leap motion controller and conventional cameras from different angles to achieve better accuracy of classification using agent-based modelling.

5. Concluding remarks

Sign recognitions in real-life applications are challenging due to the requirements of accuracy, robustness and efficiency. This project explored the viability of a real-time sign recognition system embedded in an ASL learning application. The proposed system involves the classification of 26 ASL alphabets and 30 selected features for the training of the model. The RNN model is selected since dynamic signs J and Z require the process of sequences of input. The overall accuracy of the model in the proposed work is 91.8%, which would sufficiently indicate the reliability of the approach for American Sign Language recognition. On the other hand, the Leap Motion Controller is a feasible and accurate method for ASL sign recognition. A significant amount of previous research has proposed sign recognition systems that utilise Leap Motion Controller; however, very few of them have further developed these systems into educational applications. This work fills this research gap and can subsequently open up more opportunities in the form of teaching other sign languages as well. Furthermore, the learning application can help promote ASL with its attractiveness in interactions and entertainment. In particular, the use of the application in sign instructions in schools is expected to enhance the learning motivation of hearing students in ASL and stimulate communication between hearing and hearing impaired/ hard-of-hearing students. Several suggestions are made regarding potential areas of research. A more mature application model can be produced by collecting samples from ASL users and developing more features for training the model in order to accurately classify the signs M, N and S, thereby addressing the low sensitivity of the 3 alphabets caused by thumb features. Replace if applicable to ensure clarity

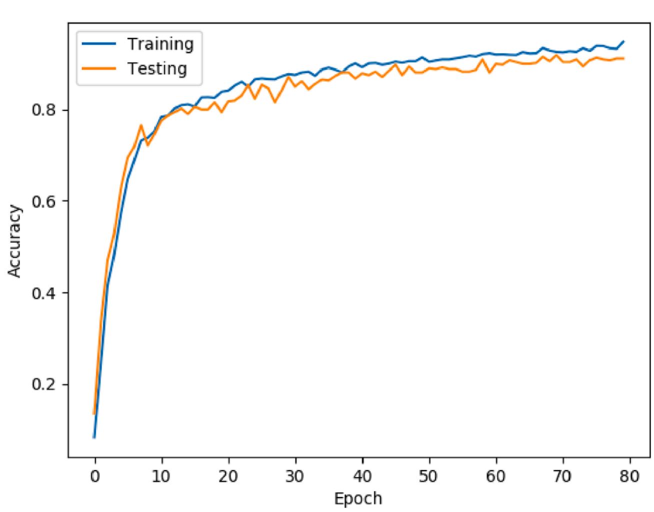

Fig. 11. Model accuracy over Epochs.

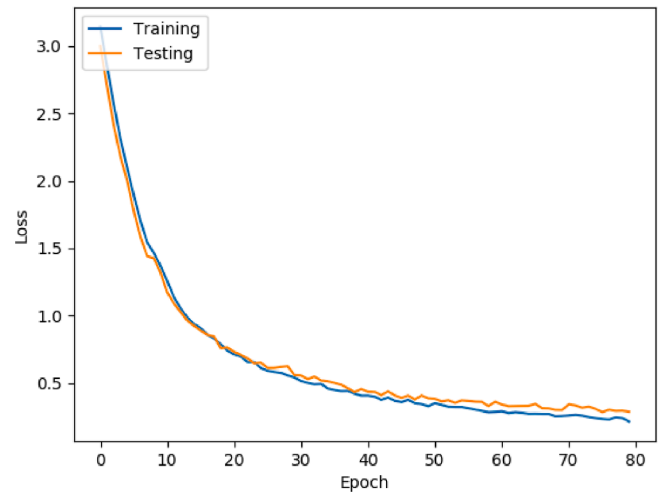

Fig. 12. Model loss over Epochs.

The study has several limitations. First, the position, angle, and number of users of leap motion will affect the accuracy of the model. The leap motion controller can detect several hand gestures, but the proposed method is restricted to recognise only one hand gesture. The leap motion controller must keep flat in order to recognise the ASL. Second, the present prototype only considers and is trained with samples from the right hand. The samples are expected to be extended to include the left hand, so that the application can also be utilised by the left-handed.

Several future works are presented to foster the relevant studies in ASL recognition. First, readers may consider the modification of the algorithmic structure, such as different types of SoftMax function, different classifiers in ASL recognition. Second, the current method is limited to leap motion controller. Readers may realise other ASL recognition methods, including image processing, video processing and deep learning approaches. The integrated approach with leap motion controller could achieve better computational accuracy using agent-based modelling. Third, the non-contactless approaches using hand gesture and motion detection can also be extended to other expert system and engineering applications in interaction design.

CRediT authorship contribution statement

C.K.M. Lee: Conceptualization, Validation, Resources, Supervision, Funding acquisition. **Kam K.H. Ng:** Conceptualization, Methodology, Resources, Writing - original draft. **Chun-Hsien Chen:** Conceptualization, Validation, Supervision. **H.C.W. Lau:** Conceptualization, Validation, Resources. **S.Y. Chung:** Data curation, Formal analysis, Writing - original draft. **Tiffany Tsoi:** Data curation, Formal analysis, Writing - original draft.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to express their gratitude and appreciation to the anonymous reviewers, the editor-in-chief and editors of the journal for providing valuable comments for the continuing improvement of this article. The research was supported by *the* Hong Kong Polytechnic University, Hong Kong and *Nanyang Technological University*, Singapore and *The University of Western Sydney*, Australia.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2020.114403.

References

Akyol, Kemal (2020). Comparing of deep neural networks and extreme learning machines based on growing and pruning approach. *Expert Systems with Applications, 140*, 112875. https://doi.org/10.1016/j.eswa.2019.112875

Aly, W., Aly, S., & Almotairi, S. (2019). User-independent american sign language alphabet recognition based on depth image and PCANet Features. *IEEE Access, 7*, 123138–123150. https://doi.org/10.1109/ACCESS.2019.2938829

Arsalan, M., Kim, D. S., Owais, M., & Park, K. R. (2020). OR-Skip-Net: Outer residual skip network for skin segmentation in non-ideal situations. *Expert Systems with Applications, 141*, 112922. https://doi.org/10.1016/j.eswa.2019.112922

Asghari, V., Leung, Y. F., & Hsu, S.-C. (2020). Deep neural network based framework for complex correlations in engineering metrics. *Advanced Engineering Informatics, 44*, 101058. https://doi.org/10.1016/j.aei.2020.101058

Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., & Massaroni, C. (2019). Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. *IEEE Transactions on Multimedia, 21*(1), 234–245. https://doi.org/10.1109/TMM.2018.2856094

Beal-Alvarez, J. S. (2014). Deaf students' receptive and expressive American Sign Language skills: Comparisons and relations. Journal of deaf studies and deaf education, 19, 508-529.

Bheda, V., & Radpour, D. (2017). Using deep convolutional networks for gesture recognition in American sign language. arXiv preprint arXiv:1710.06836.

Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PloS one, 12, e0177678.

Campello, R. J. G. B. (2007). A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters, 28*(7), 833–841. https://doi.org/10.1016/j.patrec.2006.11.010

Chong, T.-W., & Lee, B.-G. (2018). American sign language recognition using leap motion controller with machine learning approach. Sensors, 18, 3554.

Chuan, C.-H., Regina, E., & Guardino, C. (2014). In *American sign language recognition using leap motion sensor* (pp. 541–544). IEEE.

Ciaramello, F. M., & Hemami, S. S. (2011). A Computational Intelligibility Model for Assessment and Compression of American Sign Language Video. IEEE Transactions on Image Processing, 20, 3014-3027.

Du, Y., Liu, S., Feng, L., Chen, M., & Wu, J. (2017). Hand Gesture Recognition with Leap Motion. arXiv preprint arXiv:1711.04293.

Elboushaki, A., Hannane, R., Afdel, K., & Koutti, L. (2020). MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences. *Expert Systems with Applications, 139*, 112829. https://doi.org/10.1016/j.eswa.2019.112829

Fluss, R., Faraggi, D., & Reiser, B. (2005). Estimation of the Youden index and its associated cutoff point. *Biom. J., 47*(4), 458–472. https://doi.org/10.1002/bimj.200410135

Fujiwara, E., Ferreira Marques Dos Santos, M., & Suzuki, C. K. (2014). Flexible optical fiber bending transducer for application in glove-based sensors. *IEEE Sensors J., 14*(10), 3631–3636. https://doi.org/10.1109/JSEN.2014.2330998

Goyal, P., Pandey, S., & Jain, K. (Eds.). (2018). *Deep Learning for Natural Language Processing.* Berkeley, CA: Apress.

Gulli, A., & Pal, S. (2017). *Deep learning with Keras: Implement neural networks with Keras on Theano and TensorFlow*. Packt Publishing.

Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., & Sodnik, J. (2014). An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. Sensors, 14, 3702-3720.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huenerfauth, M., & Lu, P. (2010). Accurate and accessible motion-capture glove calibration for sign language data collection. *ACM Transactions on Accessible Computing (TACCESS), 3*(1), 1–32. https://doi.org/10.1145/1838562.1838564

Jeong, S., Ferguson, M., Hou, R., Lynch, J. P., Sohn, H., & Law, K. H. (2019). Sensor data reconstruction using bidirectional recurrent neural network with application to bridge monitoring. *Advanced Engineering Informatics, 42*, 100991. https://doi.org/10.1016/j.aei.2019.100991

Kamnardsiri, T., Hongsit, L.-O., Khuwuthyakorn, P., & Wongta, N. (2017). The effectiveness of the game-based learning system for the improvement of american sign language using kinect. *Electronic Journal of e-Learning, 15*, 283–296.

Khelil, B., Amiri, H., Chen, T., Kammüller, F., Nemli, I., & Probst, C. (2016). *Hand gesture recognition using leap motion controller for recognition of arabic sign language*. Sci: Lect. Notes Comput.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Lee, B. G., & Lee, S. M. (2018). Smart wearable hand device for sign language interpretation system with sensors fusion. *IEEE Sensors J., 18*(3), 1224–1232. https://doi.org/10.1109/JSEN.2017.2779466

Lee, H., Li, G., Rai, A., & Chattopadhyay, A. (2020). Real-time anomaly detection framework using a support vector regression for the safety monitoring of commercial aircraft. *Advanced Engineering Informatics, 44*, 101071. https://doi.org/10.1016/j.aei.2020.101071

Liddell, S. K., & Johnson, R. E. (1989). American sign language: The phonological base. *Sign Language Studies, 1064*(1), 195–277. https://doi.org/10.1353/sls.1989.0027

Liu, H., Yu, C., Yu, C., Chen, C., & Wu, H. (2020). A novel axle temperature forecasting method based on decomposition, reinforcement learning optimization and neural network. *Advanced Engineering Informatics, 44*, 101089. https://doi.org/10.1016/j.aei.2020.101089

Luzanin, O., & Plancak, M. (2014). Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network. *Assembly Automation, 34*(1), 94–105. https://doi.org/10.1108/AA-03-2013-020

Marschark, M., & Spencer, P. E. (2010). *The Oxford handbook of deaf studies, language, and education, (Vol. 2):*. Oxford University Press.

Naglot, D., & Kulkarni, M. (2016). *Real time sign language recognition using the leap motion controller* (Vol. 3,, 1–5.

Naidu, C., & Ghotkar, A. (2016). Hand Gesture Recognition Using Leap Motion Controller. *International Journal of Science and Research, 5*.

Napier, J., & Leeson, L. (2016). Sign language in action. In *Sign Language in Action* (pp. 50–84). Springer.

Napier, J., Leigh, G., & Nann, S. (2007). Teaching sign language to hearing parents of deaf children: An action research process. *Deafness & Education International, 9*(2), 83–100. https://doi.org/10.1179/146431507790560020

Oz, C., & Leu, M. C. (2007). Linguistic properties based on American Sign Language isolated word recognition with artificial neural networks using a sensory glove and motion tracker. *Neurocomputing, 70*(16-18), 2891–2901. https://doi.org/10.1016/j.neucom.2006.04.016

Oz, C., & Leu, M. C. (2011). American Sign Language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence, 24*(7), 1204–1213. https://doi.org/10.1016/j.engappai.2011.06.015

Parreño, M. A., Celi, C. J., Quevedo, W. X., Rivas, D., & Andaluz, V. H. (2017). In *Teaching-learning of basic language of signs through didactic games* (pp. 46–51). ACM.

Paudyal, P., Lee, J., Banerjee, A., & Gupta, S. K. (2019). A Comparison of Techniques for Sign Language Alphabet Recognition Using Armband Wearables. ACM Transactions on Interactive Intelligent Systems (TiiS), 9, 14.

Rastgoo, R., Kiani, K., & Escalera, S. (2020). Hand sign language recognition using multi-view hand skeleton. *Expert Systems with Applications, 150*, 113336. https://doi.org/10.1016/j.eswa.2020.113336

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-Validation. In L. Liu & M. T. ÖZsu (Eds.), Encyclopedia of Database Systems (pp. 532-538). Boston, MA: Springer US.

Rojas, R. (1996). Neural networks-a systematic introduction springer-verlag. New York.

Souza, A. M. (2015). As Tecnologias da Informação e da Comunicação (TIC) na educação para todos. *Educação em Foco*, 349–366.

Starner, T., Weaver, J., & Pentland, A. (1998). Real-time American sign language recognition using desk and wearable computer based video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 1371-1375.

Sun, C., Zhang, T., Bao, B.-K., Xu, C., & Mei, T. (2013). discriminative exemplar coding for sign language recognition with kinect. *IEEE Transactions on Cybernetics, 43*(5), 1418–1428. https://doi.org/10.1109/TCYB.2013.2265337

Tao, W., Leu, M. C., & Yin, Z. (2018). American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. *Engineering Applications of Artificial Intelligence, 76*, 202–213. https://doi.org/10.1016/j.engappai.2018.09.006

Tubaiz, N., Shanableh, T., & Assaleh, K. (2015). Glove-based continuous arabic sign language recognition in user-dependent mode. *IEEE Trans. Human-Mach. Syst., 45*(4), 526–533. https://doi.org/10.1109/THMS.2015.2406692

Valente, J. M., & Maldonado, S. (2020). SVR-FFS: A novel forward feature selection approach for high-frequency time series forecasting using support vector regression. *Expert Systems with Applications, 160*, 113729. https://doi.org/10.1016/j.eswa.2020.113729

Vassilvitskii, S. (2007). *K-Means: Algorithms, analyses, experiments*. Stanford University.

Wu, J., & Jafari, R. (2017). Wearable Computers for Sign Language Recognition. In S. U. Khan, A. Y. Zomaya, & A. Abbas (Eds.), *Handbook of Large-Scale Distributed Computing in Smart Healthcare* (pp. 379–401). Cham: Springer International Publishing.

Wu, J., Sun, L., & Jafari, R. (2016). A wearable system for recognizing american sign language in real-time using IMU and surface EMG sensors. *IEEE Journal of Biomedical and Health Informatics, 20*(5), 1281–1290. https://doi.org/10.1109/JBHI.2016.2598302

Wu, J., Tian, Z., Sun, L., Estevez, L., & Jafari, R. (2015). Real-time American Sign Language Recognition using wrist-worn motion and surface EMG sensors. In *In 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)* (pp. 1–6).

Zhong, B., Xing, X., Luo, H., Zhou, Q., Li, H., Rose, T., & Fang, W. (2020). Deep learning-based extraction of construction procedural constraints from construction regulations. *Advanced Engineering Informatics, 43*, 101003. https://doi.org/10.1016/j.aei.2019.101003