International Journal of Social Robotics
https://doi.org/10.1007/s12369-021-00819-0

Dynamic Iranian Sign Language Recognition Using an Optimized Deep Neural Network: An Implementation via a Robotic-Based Architecture

Salar Basiri[1] · Alireza Taheri[1] · Ali F. Meghdari[1,2] · Mehrdad Boroushaki[3] · Minoo Alemi[1,4]

Accepted: 24 July 2021

Abstract
Sign language is a non-verbal communication tool used by the deaf. A robust sign language recognition framework is needed to develop Human–Robot Interaction (HRI) platforms that are able to interact with humans via sign language. Iranian sign language (ISL) is composed of both static postures and dynamic gestures of the hand and fingers. In this paper, we present a robust framework using a Deep Neural Network (DNN) to recognize dynamic ISL gestures captured by motion capture gloves in Real-Time. To this end, first, a dataset of fifteen ISL classes was collected in time series; then, this dataset was virtually augmented and pre-processed using the "state-image" method to produce a unique collection of images, each image corresponding to a specific set of sequential data representing a class. Next, by implementing a continuous Genetic algorithm, an optimal deep neural network with the minimum number of weights (trainable parameters) and the maximum overall accuracy was found. Finally, the dataset was fed to the DNN to train the model. The results showed that the optimization process was successful at finding a DNN structure highly suitable for this application, with 99.7% accuracy on the verification (test) data. Then, after implementing the module in a robotic architecture, an HRI experiment was conducted to assess the system's performance in real-time applications. Preliminary statistical analysis on the standard UTAUT model for eight participants showed that the system can recognize ISL signs quickly and accurately during human–robot interaction. The proposed methodology can be used for other sign languages as no specific characteristics of ISL were used in the preprocessing or training stage.

Keywords Deep Neural Network · Pattern recognition · Sign language recognition · Human–Robot interaction · Machine Learning · Children with a hearing problem

1 Introduction

Sign Language (SL) is a visual language and a non-verbal communication tool that is used by individuals with hearing problems to communicate with other people. According to statistics published by the World Health Organization (WHO), 450 million people worldwide are seriously hard-of-hearing, and 30 million of these are children [1].

Studies show that a lack of attention to teaching SL to deaf children affects not only their language development, but also disrupts their mental functioning [2, 3]. These studies considered SL as a structured language necessary for the cognitive and mental development of children with hearing problems.

Despite the important role of sign language in deaf people's lives, this language, especially Iranian Sign Language (ISL), is not well known to the general public, and individuals with hearing impairments face serious problems in communicating with others. Therefore, there is a serious need in society to familiarize both deaf and typically developing people with this necessary communication tool.

In juxtaposition, due to the rapid growth in technology, the use of social robots in the education and clinical treatment of children has been attracting wide attention in recent

✉ Alireza Taheri
artaheri@sharif.edu

[1] Social and Cognitive Robotics Lab, Sharif University of Technology, Tehran, Iran

[2] Chancellor of Fereshtegaan International Branch, Islamic Azad University, Tehran, Iran

[3] Department of Energy Engineering, Sharif University of Technology, Tehran, Iran

[4] Faculty of Humanities, Islamic Azad University, West Tehran Branch, Tehran, Iran

Published online: 29 August 2021

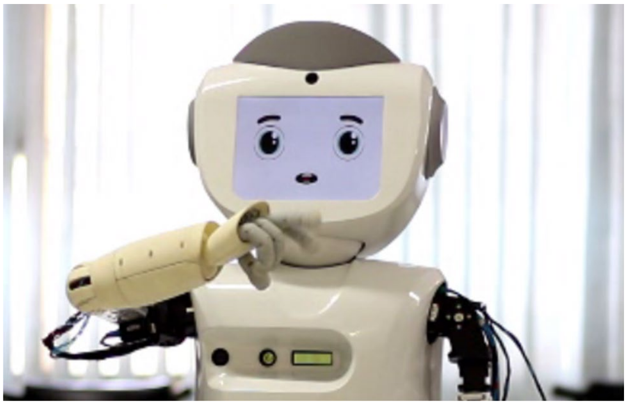


Fig. 1 RASA robot performing an ISL sign

years. Implementing and empowering social robots in sign language education is an example of such studies [4]. Specifically, a social robot named “RASA” (Fig. 1) has been designed and developed in the Social and Cognitive Robotics Lab., Sharif University of Technology, Iran, to facilitate teaching ISL to deaf and hard of hearing children. Having thirteen Degrees of Freedom (DoF) in each arm, including seven DoF in its active fingers, enables RASA to perform different signs and interact with users with hearing problems via ISL [5]. (Some information about the RASA robot: Weight: 36 kg, Height: 119 cm, CPU: mini-PC intel NUC and Arduino, Operating System: Windows 7 and Robot Operating System (ROS), Total DoF: 32). As RASA is able to open and/or close each finger independently, it can perform a large number of ISL classes. In addition, the DoFs in arms and elbows allow RASA to reach an adequate workspace for representation of most of the ISL signs. However, the robot is limited in its ability to display all existing handshapes since some ISL signs require complex finger patterns (for example, laying one finger over another), which are not possible for RASA to perform.

The first requirement for a robot, like RASA, to play an effective role in sign language education is to recognize the different signs of SL performed by users with acceptable accuracy in real-time and in varying conditions. The robotic-based recognition mechanism should be robust enough in various operating conditions, such as different environments and different users (with differing physical properties and/or performances when using SL words), to show an appropriate recognition rate as well as reproduce/perform various signs [6].

Sign language movements generally consist of two parts: 1) dynamic gestures, which include the movement of hands and arms in 3D space, and 2) static postures of hands/fingers during the run of sign language. As the posture of the hand and fingers may alter significantly during the performance, as shown in Fig. 2, simultaneous pattern recognition of both these parts by robotic systems is a challenge. Iranian sign language (ISL) is a complete language that employs signs made with the hands and other gestures, including facial expressions and body postures. There are other factors that play important roles in demonstrating the purport of the gestures as well. These factors include the face, eyes, head, and body. ISL is made up of several main elements including (1) the state of the hand, (2) movement of the hand/arm, (3) the place that hands/arms settle on it, and 4) the direction of the palms of the hands [7, 8]. During the performance of any ISL signs, all the spatial parameters of the arm and the fingers may alter (e.g., a dynamic movement with fingers in a specific arrangement in any of the above steps); hence, the tool used to capture ISL data should be able to simultaneously capture the geometry of the hand and all of the fingers.

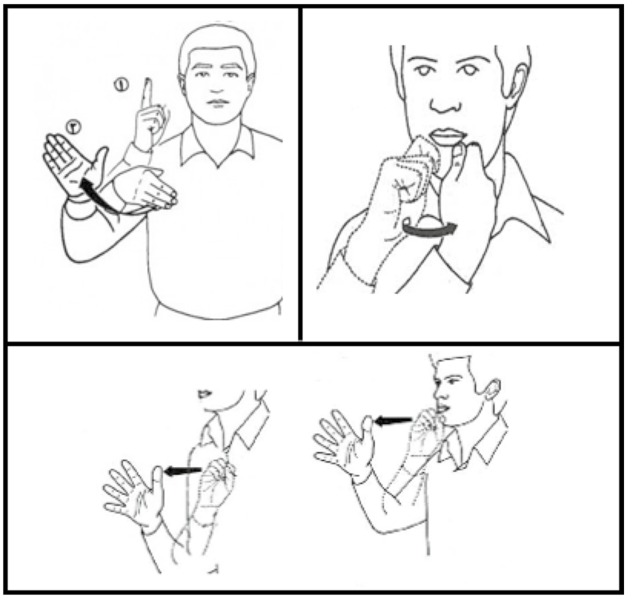


Fig. 2 A Schematic of 3 ISL signs. All spatial parameters of the hand and the fingers vary during the signs

Different tools such as RGB cameras [9, 10], Data Capturing Suite/Gloves [11–13], Microsoft Kinect sensor [14–16], ToF[1] cameras [17, 18], Leap Motion sensor [19–21], and combinations of these tools [22] have been used to capture SL signs data in previous researches. Among these, Data Capturing Gloves have shown the highest accuracy in sign recognition; however, their cost and user discomfort may be considered as disadvantages [23]. In the following, some mechanisms for data capturing and autonomous movement pattern recognition algorithms are presented.

Regarding recognition algorithms, various kinds of Neural Networks [7, 8, 10, 24, 25], classic classification

[1] Time-of-Flight.

methods such as Nearest Neighbor [26, 27], Support Vector Machine (SVM) [28, 29], Hidden Markov Models (HMM) [24, 28], and combinations of these algorithms [30, 31] have been used in related works. One neural network, which has recently become very popular in the field of pattern recognition, is the Deep Neural Network (DNN) [32]. This network consists of more than two layers that are utilized for complex non-linear mapping and mathematical modeling between the input and output [33]. Despite the complexity of the relationship between the input and output, DNN can act as a powerful tool for classification; and with the proper selection of network structures and enough input data, it shows very high accuracy in classification and pattern recognition [34].

In 2019, a framework for continuous sign language recognition using the DNN algorithm was presented by Cui et al. [35]. In their study, the system captured a video of RGB frames and optical flows of SL signs as the input and produced a sequence of detected words as the output. Before this, the HMM algorithm, which has a limited capacity to record temporal data, was often used for autonomous and continuous recognition of SL words/signs in related studies. Conversely, the authors of [34] proposed an algorithm that uses DNN with temporal fusion layers as the sequence learning module. One of the challenges they tried to overcome was how to achieve acceptable accuracy in detecting the SL words/signs with a small dataset (less than typically required) for this methodology.

In 2018, Taskiran et al. [36] developed a system for recognizing American Sign Language (ASL) in real-time that uses (1) a DNN for feature extractions and classification, and (2) a Convex Hull algorithm for detecting arm/hand positions. In their study, the arm/hand positions are first extracted and then sent to the DNN for classification. The hand movements in [36] were not dynamic, and only the initial and final positions of the hands and fingers were used to specify the words'/signs' classes. It should be noted that their results are not applicable to ISL signs because the dynamics of the movement (not necessarily the initial/final hands' postures) very often represent the sign/word in ISL. A similar methodology was presented by Tang et al. [37]. In their research, an algorithm for detecting hands (captured by a Kinect sensor) was implemented, and then a DNN was used to extract the features of the hands' static postures for a number of SL words. In [38], the authors implemented an algorithm similar to that presented in [37] on a larger dataset of ASL signs and observed an accuracy of 92.8% in sign recognition. The authors of [39] deployed an ISL recognition vision-based system that works with 20 classes of dynamic signs. Using the HMM method, they reached a mean accuracy of 97.48% on the dataset gathered using videoframes. In addition to image-based methods, in which the inputs are captured by cameras/Kinect, the authors in [40] also used EMG sensors to first collect the input data and then apply a DNN. To be able to feed the data to the CNN layers, the authors implemented a sliding-window mechanism to segment the pre-processed data. They reported an accuracy of 83% for recognition of sign language postures. Their suggestion that future works find a way to optimize the DNN structure contributed to the formulation of our study. The authors of [41] proposed a method using modified KNN to classify 40 dynamic Arabic sign language words gathered by DG5-VHand sensor gloves, reaching an accuracy of 98.9%. The preprocessing stage in this work strongly emphasizes the temporal dependence of the data. Also, using a leap-motion sensor, the authors of [20] proposed an SVM + KNN methodology, which reached an accuracy of 79.83% on detecting 26 classes of static ASL signs.

In 2018 and 2019, Dong proposed an algorithm to recognize aircraft icing and faults in sensors/actuators of an airplane in real-time using DNN [42, 43]. The novelty of this work was that the airplane data of the time-series was coded as pixels of a picture, which were then used as the inputs for their DNN. The authors called this technique the "state-image" method for the preprocessing of the data. Using this application, they reached higher accuracy in the recognition process compared to other related algorithms in the literature.

In this paper, as a practical application of machine learning algorithms for social Human–Robot Interaction (HRI) to empower our social robotic platform to interact via ISL, a framework which uses Data Gloves for capturing sequential data as time-series to recognize Iranian Sign Language words/signs is proposed. Our goal is to classify the data from different sensors over time; therefore, we are dealing with multi-dimensional sequential data. As we were not able to generate a large database of dynamic ISL signs due to the lack of enough participants, shortage of resources, and the impossibility of using virtual simulation in this application, we aimed to build a platform that does not require a large amount of data to classify ISL signs/words. Thus, without the need to apply Recurrent Neural Networks (RNN) and/or complex feature extraction algorithms from time-series, this study contributes to the existing literature by developing a platform able to robustly recognize the ISL words/signs using (1) the state-image preprocessing technique to convert the sensory data to intensity images that actually transform the dynamic nature of the patterns to a static one (as the first paper's contribution), and more importantly (2) a deep neural network whose optimized structure was determined via a genetic algorithm optimization process (as the second and main contribution of the study). The implemented methodology for data pre-processing significantly transforms the dynamic nature of gesture recognition into a static image processing problem that can easily and accurately be solved by DNNs. This study differs from previous research in the simultaneous usage of the state-image method (as a new approach for analyzing time-series data in this application) and an optimization algorithm to determine the optimized structure of the DNN, which makes the network more accurate, and at the same time, less heavy. The main objective of this study is to build a sign recognition framework that satisfies the need for an acceptable (in terms of robustness, accuracy, and speed) pattern recognition module in order for RASA to interact with human users using sign language and to build a connection between machine learning, social robotics, and human–robot interaction to satisfy the need of an ISL teaching robot.

Due to the non-availability of a standard dataset for ISL words, we first gathered a dataset of fifteen ISL classes, then augmented the data virtually by preprocessing it through the state-image method. Next, the optimized structure of the deep neural network (including the number of Convolution filters and the Convolution kernel size) used to train the generated images was determined by applying a genetic algorithm. After the training process, the performance of the optimized DNN structure was investigated with a test set not seen by the network during the training process. Also, other techniques of this methodology allow the system to act robustly in different user/environmental conditions. Then, we investigated whether the proposed algorithm performs better than similar algorithms from related works in automatic sign language recognition. Finally, we assessed the system's performance in real-time applications through a statistical analysis conducted on data from the standard UTAUT model collected during HRI sessions.

2 Database Collection

2.1 Setup and Conditions

The data collection process was done in conditions that would allow the system to work robustly against the user's anthropometric characteristics, the geometrical direction of the user in 3D space, and as far as possible, the speed and the start/finish position of the movement. Other techniques were also used in the pre-processing stage (details to follow in Sect. 3) to make the system robust against initial sensor bias and establish the global geometric position of the user's body and hands in space. In this way, the system will only be sensitive to the pattern of the hand gesture and relative position of the hand to the previous time frames.

Most of the signs in ISL are performed by the right hand. The movement patterns of the fingers, palm, and forearm (in most cases) determine the sign (i.e., alphabet letters or words) performed by the user. The fingers do not necessarily follow the same pattern during the performance of an ISL sign; thus, we cannot use a single sensor for all the fingers. Thus, the structure of the selected ISL patterns needs to be collected and processed so that the number of parameters describing the gesture is not more than needed (to avoid over-fitting) but is also large enough to be able to describe the gesture (to avoid under-fitting). In this paper, the *XYZ* position of 5 fingertip sensors, palm sensor, forearm sensor, and also the 3 angular directions of the palm sensor were chosen to describe the ISL gestures (Fig. 3). Thus, a total of 24 parameters were chosen to describe the motion patterns, which means each specific ISL class is described by a specific set of 24 time series. Each sensor's output is a time series, and all these time series together correspond to a specific sign. As the readers see in the next sections, these sets of time series will be coded as the pixels of an image and eventually compose images 24 pixels in width.

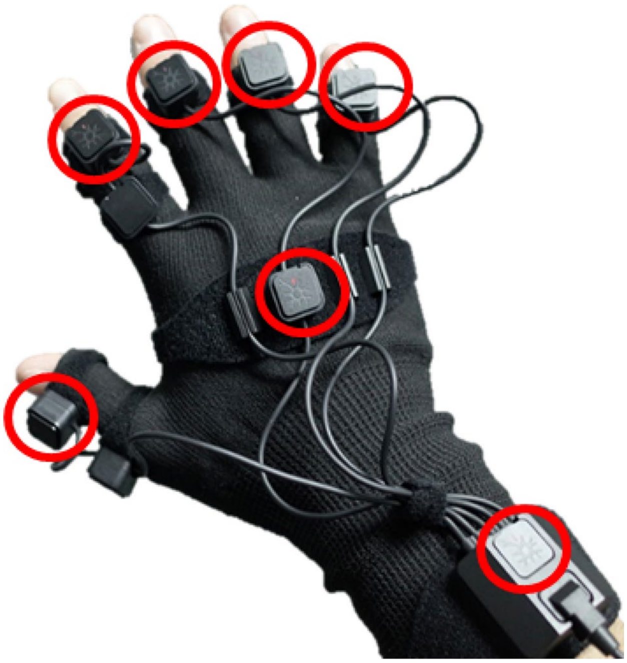


Fig. 3 Glove sensor used in the data collecting process. Five fingertip sensors, a palm sensor, and a forearm sensor are used to describe patterns. IMU sensors are used in this glove

The data transfer rate from the motion capture gloves to the processing unit is almost 60 Hz (with a standard deviation of 2 Hz). The process of data acquisition is done in three different body directions in both the standing and sitting positions, as shown in Fig. 4. Fifteen classes of data were collected at different random speeds and different start/finish hand positions for each class. The signs were performed by five users with significant physical differences and a minimum and basic knowledge of ISL. To be more specific, for each class, we asked each participant to perform twice in each direction (inclined to the right, forward, and inclined to the left), which gave us 6 cases and an additional 2 cases with random direction and speed. They did not know the selected 15 signs prior to

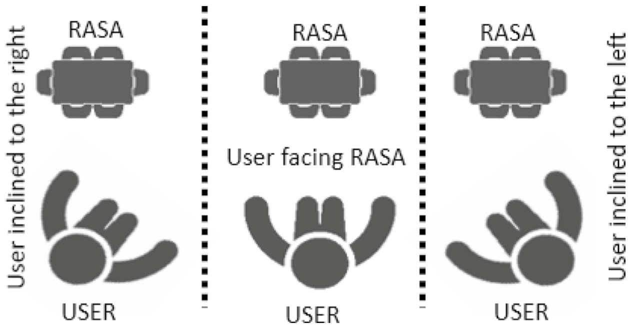


Fig. 4 A schematic of the data collection setup

participating in data collection experiments. An ISL expert performed each sign for each participant in person, and the participants were then asked to repeat the performed sign. The dataset is both user-and environment-independent, as the numerical sensor values are normalized in the pre-processing section. It is important to mention that each sign was completed during a different time period. This occurrence happened naturally because different users in different conditions do not necessarily perform signs at the same speed. Thus, finding a criterion to compare these performed signs regardless of the user's performance speed is an important issue that needs to be addressed in the pre-processing section. The dataset is perfectly balanced, as the amount of data in each class is identical.

2.2 Data Visualization

Iranian sign language is a collection of gesture patterns of the fingers, hand, and forearm. For example, the three top diagrams in Fig. 5 represent the palm position of a specific sign, performed by different users. As seen, the position of just the arm is quite different from person to person. Thus, the pre-processing method and the mechanism of the detection system must be chosen so that the system will be fast enough to detect and classify signs in a minimum amount of time, and also complicated enough to analyze a huge set of time series together to extract features that distinguish different classes. Figure 6 illustrates the position of the palm in three Decatriene planes for a single specific ISL sign performed 10 times by different users. Although the trajectories in each image almost lie in a limited region, different anthropometric characteristics, different speeds of performance, and different geometrical directions in 3D space can lead to totally different variations in this figure. Considering all the issues mentioned above, we hypothesize that coding these time series into pixels and composing pictures for each class using the "State-Image" pre-processing method can lead to fast and accurate results. This hypothesis is evaluated in the next sections.

2.3 The ISL Dataset

In this study, 15 ISL signs were chosen to be studied, and 40 sets of time series were collected for each to make a database of 600 time-series, each with a specific label. Then, the dataset was virtually augmented to make a database of 30,000 data (see Sect. 3.1), and all of these data were turned into images using the State-Image approach. Thus, the final dataset was composed of almost 30,000 images, each representing a specific sign of ISL. To correctly train and evaluate the model, it was necessary to segment our dataset into three parts: Training data, Validation data, and Verification (test) data [44]. In this work, 60% of the total datasets were used for training, 20% for evaluation, and the remaining 20% for testing. The process was done by the standard Sklearn model selection library of python (https://scikit-learn.org/stable/model_selection.html), which uses a train-test split protocol to shuffle and randomly select the data for each group. It should be noted that we segmented the dataset at the beginning of the entire project, meaning that the test (verification) dataset was not used to train any network, not in the optimization process nor the optimal DNN. It was only used to assess the accuracy of networks on an unseen dataset; namely, the models first predicted the labels of these data, and we then compared it to their original label to calculate the verification accuracy for the fitness function. A summary of the properties of this section can be seen in Table 1.

3 Data Pre-Processing

3.1 Data Augmentation

DNNs need a large amount of data to be trained effectively, so only using the small amount of the data collected from the motion capture glove will not lead to promising accuracy in pattern recognition. Therefore, it is necessary to somehow virtually create more data out of the in-hand data with the same label without actually using an experimental setup. This technique is called data augmentation, and it is widely used in data processing. In the field of image processing, data augmentation techniques included rotation, changing brightness, adding noise, mirroring, cropping, etc. [45].

In this paper, three techniques were used for data augmentation: (1) Adding time delay (stagnation frames) to the start/end of each set of time series, (2) Cropping a part of the start/end of each set of the time series, and (3) Adding random Gaussian noise to a randomly chosen collection of datasets. In addition to enlarging the size of the in-hand ISL dataset and affecting the training process positively, the methodology is beneficial in other aspects such as:

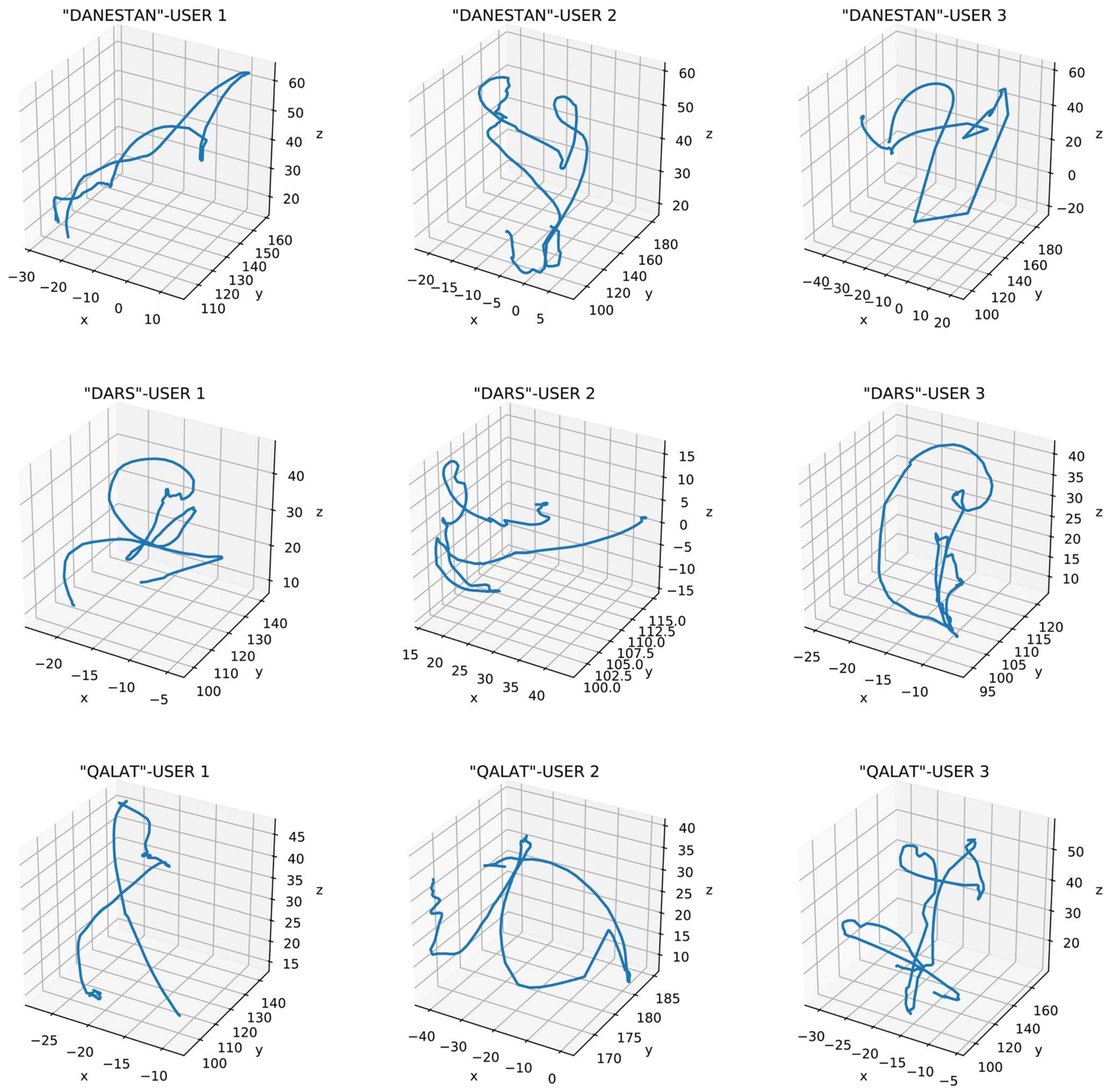


Fig. 5 An example of different palm trajectories made by different users for the same sign

1. Adding stagnation frames to the start/end of each set of the time series increases the robustness in the start/finish detection mechanism against faults. This means that if for any reason the user performs the sign with a time delay at the beginning (start stagnation), or the end of the performance is detected later than it should be (end stagnation), or in general, the duration of the performance is not detected correctly, the system can act more robustly. The stagnation length is up to 20% of the performance length, with a step of 5% added to the beginning and/or the end of the time series.
2. Cropping a different number of frames at the beginning/end of the time series can increase robustness in cases where the start/end detection mechanism detects the beginning later, or the end sooner than it should. It also increases the robustness in cases where the hand position at the beginning/end of the performance is not identical for different users. The cropping length is up to 10% of the performance length, with a step of 5% applied to the beginning and/or the end of the time series.

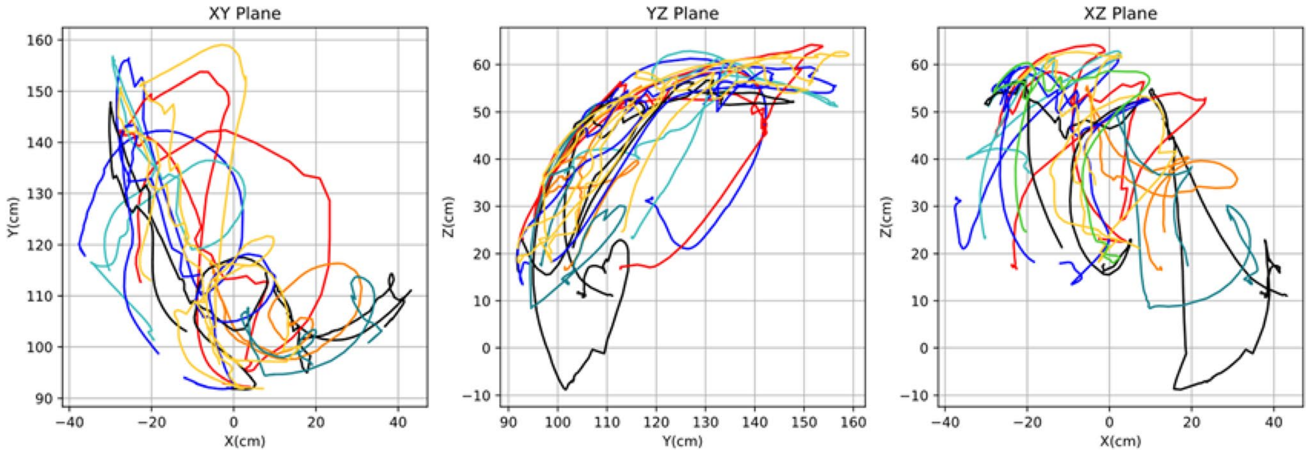


Fig. 6 The position of the palm in 3 Decatriene planes for a single ISL sign performed 10 times by different users

Table 1 A Summary of the Properties of the Collected Dataset

Number of the classes	15	Number of the training data	18000
Number of data collected for each class	40	Number of the validation data	6000
The dataset size after augmentation	30,000	Number of the verification data	6000

Therefore, according to 1 and 2, each data becomes 49 data in total, and after we add another randomly cropped time series, it becomes 50 data out of one.

3. Adding random Gaussian noise can make the system more robust against sensor faults, environmental noise, motion vibrations, etc. Gaussian noise is a commonly observed noise in digital images that is usually caused by sensor faults and poor lighting [46]; thus, it is natural to choose this kind of noise as a data augmentation tool in our application. We applied white Gaussian noise to 20% of the data in each class, with a random signal-to-noise ratio per sample of 35–40 dB.

In summary, it is possible to increase the quality of the training model and robustness of the model in pattern recognition and gesture classification by implementing the mentioned data augmentation methodology.

3.2 State-Image Pre-Processing

The State-Image Method is a novel pre-processing technique that codes every state of the system into pixels of an image. It is used to transform a set of time series into a single image, both corresponding to the same class. In this paper, this technique is implemented in the following steps:

1. Each set of time series is compressed or stretched into a specific pre-defined number of frames, regardless of the duration of the performed sign. Each set is framed into 60 frames using linearly spaced elements. By doing this, the effect of time period differences in different performances is omitted, leading to an increase in the robustness of the system. Almost all of the SL signs were performed in 150–180 frames, with an average of about 160 frames. Since the data transfer rate is unnecessarily large for our application (i.e. collecting data every 16 ms), we applied down-sampling at the preprocessing stage for a few purposes: (1) To keep the dataset easier to store, manage, and process, it is better to omit redundancies in the dataset, and (2) large values of data frames cause significantly large asymmetry in the shape of images produced by the state image method (24 pixels in height, and large values for width). Feeding this kind of data to DNNs is uncommon.
2. Each number in a time series is coded as intensity pixels, ranging from 0 to 255; to do this, in each time series the minimum sensor value is chosen to be 0 and the maximum is chosen to be 255, and every value in between is calculated linearly to have a value in the range of 0 to 255. In other words, for each sensor, we have 60 pixels with a value of brightness between 0 and 255. Therefore, as all these images are one pixel in depth, each time series is coded to an intensity image.
3. The whole image corresponding to a set of time series is made by setting these pixels in a row together, each representing the value of a sensor over time. As we chose 24 parameters to describe the gestures, the images will be 24 pixels in width and 60 pixels in length. Therefore, each set of time series (one sign of ISL) is represented by a 24*60 image. Figure 7 illustrates the proposed

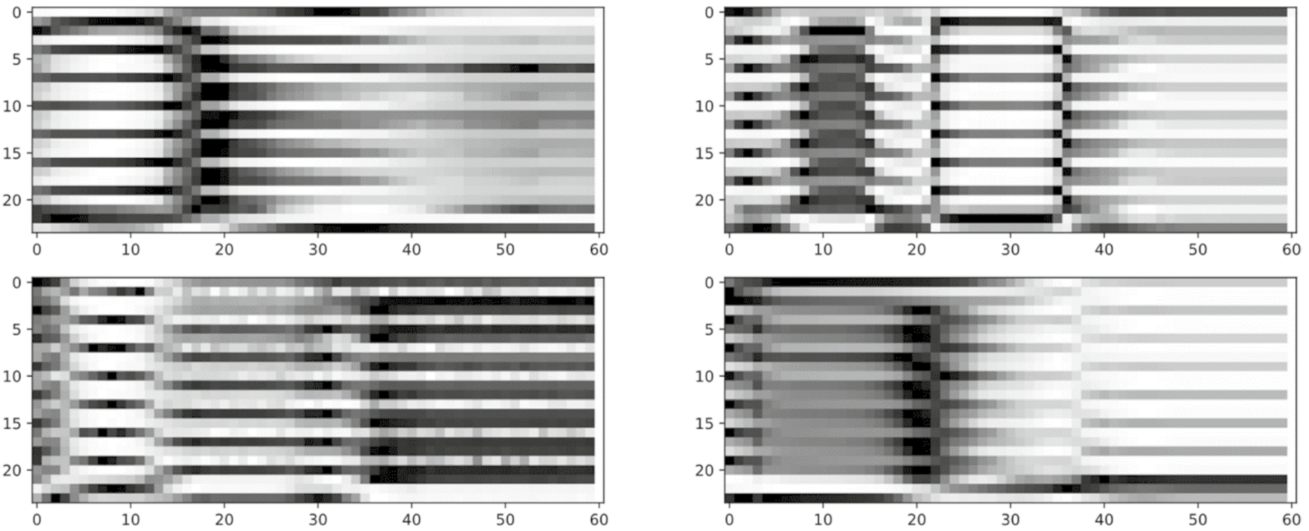


Fig. 7 The diagram of four different dynamic ISL hand gestures turned into images with the State-Image approach. Each image represents a specific sign (ISL class) performed by the user, which is 24 pixels in width (corresponding to 24 states of the system) and 60 pixels in length (corresponding to 60 time frames)

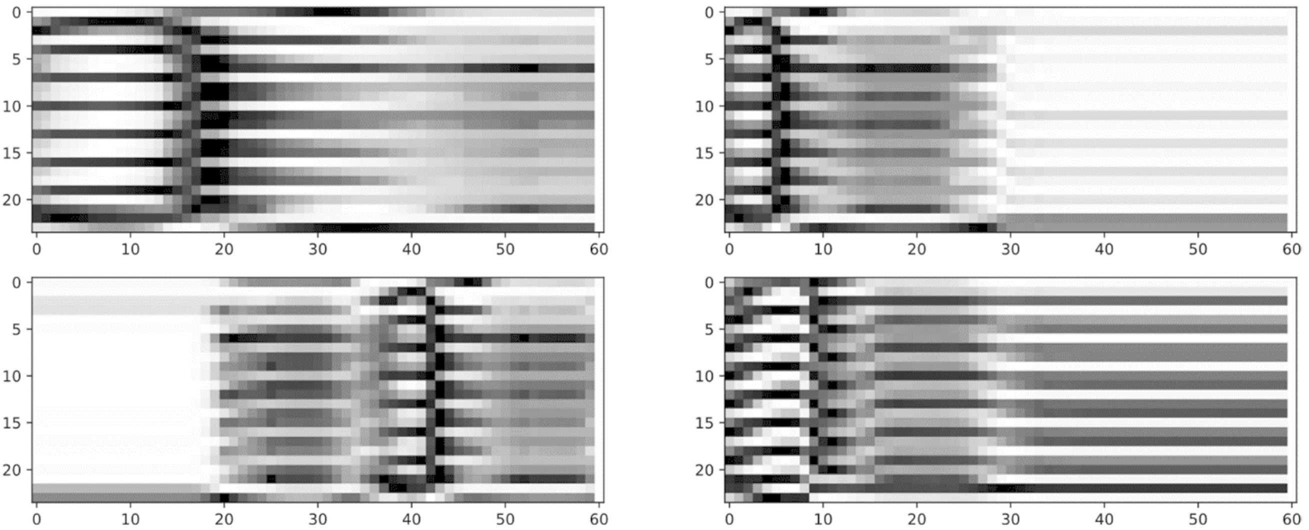


Fig. 8 The illustration of the results of the implemented data augmentation technique. All four images correspond to the same class and all have the same label. The top left shows the original image, and the other three show augmented copies of it

methodology and Fig. 8 depicts an example of the data augmentation process results.

The proposed approach is remarkably beneficial for data pre-processing in these aspects:

1. The dynamic nature of the hand gesture detection becomes static; and thus, much easier to process and recognize without needing to use complex recurrent networks, and avoiding high computational costs and the need for a larger amount of the data. More precisely, each ISL sign is characterized by a dynamic hand gesture over time. This means each sensor's output is a time series, and all these time series together correspond to that sign. The output of the model is not only a function of the current hand and figure positions but is also affected by previous positions. Therefore, the nature of the classification problem is dynamic and needs to be handled by RNNs, like LSTM, which are usually heavy (need more memory and processing power) and need a lot of data to be trained (especially in this case where we have 24 parameters, not just one); but, by using the proposed method all of the time series are coded

together into a static image that represents the 24 time series together. Hence, a good DNN is enough to build an accurate recognition system and there will be no need to gather a large amount of data or deal with much heavier networks and computational costs.
2. DNNs are remarkably powerful tools for image classification and have mostly been developed in this regard. When a hand gesture pattern recognition problem is transformed into an image classification problem using DNNs, we can expect higher classification accuracy in comparison to classic methods like SVM or HMM.
3. Coding each time series into intensity values makes the system insensitive to the pure values of the sensor positions, and it is only the trend of movement, or in better words, the gesture pattern under the scope of the model to extract useful features. That means that it is not important in which spatial position the movement has begun or finished for our model, but rather what the current state of the user's hand and fingers' position are relative to the prior states. In this way, the model acts more robustly against totally different performances caused by different users and different environments.
4. The effect of different performance time durations was omitted due to the equality of the frame numbers in the final processed images.

In summary, by implementing the technique of "State-Image" in the data pre-processing stage, a dataset is built that enables the model to be trained with a lower amount of data and computational cost and to potentially more accurately and robustly classify dynamic hand gestures.

We should note that the proposed methodology can be used for other sign languages as well since no specific characteristics of ISL were used in the preprocessing stage. When these signs are represented as intensity images, they can be classified using DNNs; therefore, we can claim that the proposed method can also be used to recognize other sign languages.

4 Deep Neural Network (DNN)

In this section, we go through the building and proper training of our DNN model. The deep neural network used in this paper is implemented by the Keras library [47] with a TensorFlow backend in python 3 environments.

4.1 Building Layers

The input of the DNN is a set of images with dimensions of 24*60*1 (24 pixels in width, 60 pixels in length, and 1 pixel in depth, which is the value of the intensity).

The first layer is set to be a 2D Convolution layer in which the depth (filter number) and the kernel size are not known and are found through the optimization process. The stride is set to 1 due to the small size of the images and to avoid any probable information loss. Then, a 3×5 Maxpooling layer is set to reduce the size of the total net. Similar to the first layer, another Convolution layer is set with an unknown filter number, kernel size, and a stride number equal to 1, followed by another 3×5 Maxpooling layer. It should be noted that we chose a 3×5 size for the pooling layers as every 3 rows of each image belongs to a single sensor, and the width of the images is considerably higher than their height, so a common size of 2×2 will not be useful. After that, a Flatten layer is set to flatten the output of the previous layers, and finally, a Dense (fully-connected) layer with an output dimension equal to the total class number, which is 15. In the Convolution layers, Relu was chosen as the activation neuron function. Softmax was chosen as the activation neuron function in the Dense layer (i.e., the last layer). Therefore, the output of the DNN is an array of 15 elements with values between 0 and 1, which actually represent the class number of the given data, (i.e., the ISL sign in our case).

One of the most common problems with artificial neural networks is Over-fitting, which means the network is somehow "memorizing" the data (and not "learning" it). This leads to remarkable prediction error rates when dealing with new data that has not been seen during the training. Several techniques can be used to avoid this problem, including the Dropout technique [48]. In this technique, some of the neurons disappear in the total net during each training epoch and the training process continues without them. These neurons return to the total net with their old weights in the next steps, and this cycle continues to the end of the training process. This method has been shown to have promising results in avoiding the Over-fitting problem [44]. In this work, a Dropout with a rate of 0.2 was added to the network to avoid Over-fitting.

4.2 Training The Network

There are several conventional methods for training the DNNs and adjusting its neurons' weights. Two of the most common learning methods are ADAM[2] and SGD[3] [49]. ADAM is usually faster than SGD and it is more robust against noisy and imbalanced data, but in some cases, like when the size of the dataset is not big enough, this can lead to divergence and the training process cannot be completed [50]. In our optimization approach, the networks that are not

[2] Adaptive Moment Estimation.

[3] Stochastic Gradient Descent.

compatible with the ADAM default learning rate (which is 0.001) are omitted automatically.

Since our dataset is perfectly balanced, the loss function (or objective function, or optimization score function) is set to be Categorial Cross-Entropy [51], which means if M is the total class number, $y_{o,c}$ is a binary indicator (0 or 1) showing whether class label c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability observation o of class c, The loss is calculated as (1):

$$Loss = -\sum_{c=1}^{M} y_{o,c} \log(P_{o,c}) \tag{1}$$

Several criteria can be implemented to stop the training process, like defining a maximum number of epochs, a specific training or validation accuracy, etc. As was mentioned in Sect. 4.1, over-training the data can lead to the Over-fitting of the net; and thus, large error rates in test data classification. Over-fitting begins when the validation accuracy starts to fall while the training accuracy is still ascending, and hence, the training process should be terminated. In this paper, the trend of the validation accuracy was observed during the last 15 epochs of the training, and when it ceased to ascend the process stopped. This step was implemented as a Keras callback function called "Early Stopping". This technique, in addition to the Dropout technique, helped us avoid the Over-fitting problem and led to a more reliable model.

5 Optimizing dnn's structure using genetic algorithm

The Genetic Algorithm (GA) is a metaheuristic optimization algorithm developed by Goldberg in 1989. It is inspired by Natural Selection, which is the differential survival and reproduction of individuals due to differences in phenotype. The Genetic algorithm is commonly used to generate high-quality solutions for multi-variable complex optimization and search problems by relying on bio-inspired operators such as *mutation*, *crossover,* and *selection* [52]. GA is often used for static optimization problems where the cost function does not change with time, and it has been shown to be a fast and powerful tool in dealing with real-world problems where the cost function is dependent on many variables, especially ones related to nature.

Deep Neural Networks are also inspired by the mechanism of the human mind for processing and data perception. As there is no explicit mathematical relationship between the structure of the DNN and the final test accuracy, so metaheuristic algorithms should be used to optimize the network. Due to the complex, static, and nature-inspired essence of the defined optimization problem, this paper contributes the Continuous Genetic algorithm as a good choice to find the optimum layer structure of the model as defined in the previous section.

The characteristics of the continuous GA used in this paper are explained in detail in the next section.

5.1 Fitness Function

As previously mentioned, the dataset is segmented into three parts: training data, validation data, and verification (test) data. The final reported accuracy of the neural nets is conducted on the verification data that were not seen during the training and validation processes; therefore, one of the criteria is to compare the different networks. The main goals of the validation data are to (1) establish the point at which the training should be stopped (see Sect. 4.2), and (2) find the optimum neural network structure. In this paper, an optimal DNN is defined as a DNN that produces the highest overall accuracy with minimum trainable parameters. The overall accuracy is defined as the weighted average accuracy on the training, validation, and verification data. In other words, if acc_{trn} is the maximum training accuracy and acc_{vld} is the maximum validation accuracy over all of the epochs, acc_{vrf} is the accuracy over the verification data and p is the number of trainable parameters, then fitness function $\boldsymbol{f}$ of each DNN is defined as follows:

$$f = \frac{acc_{trn} + 2 \times acc_{vld} + 3 \times acc_{vrf}}{6} \times 1000 + \frac{10^6 - p}{1000} \tag{2}$$

We sought to maximize this fitness function. The equation takes all the accuracies into account with different weights, with the test accuracy having the maximum weight. We segmented the dataset at the beginning of the entire project, meaning that the test (verification) dataset was not used to train any network, not in the optimization process nor the optimal DNN. It was only used to assess the accuracy of networks on an unseen dataset; namely, the models predicted the labels of these data, and we compared it to their original label to calculate the verification accuracy for the fitness function. Since DNNs usually reach high accuracies, it is important to take into account small differences in the overall accuracy. As seen in the equation above, a 0.1 percent increase in the overall accuracy is multiplied by 1000, leading to a 100 point increase in the fitness function. On the other hand, a 100,000 increase in the number of trainable parameters decreases the fitness function by 100 points. Therefore, at the end of the optimization process, only networks with the highest accuracies and lowest trainable parameters remained, meaning the networks are accurate, light (need less memory), and fast (need less time to adjust the weights).

Table 2 An Overview of the Specifications of the Implemented GA

Selection	Initial population size	Number of genes in each chromosome	Fitness function
Modified Rank Weighted Random Pairing	32	4	Eq. 2
Stopping Criteria	Number of mutated genes in each generation	Reproduction	Continues/Discrete
Maximum generation number = 25	6	Extrapolation with Crossover	Continues

5.2 Optimization Parameters

The optimization problem in this paper is to find the optimal hyper (structural) parameters of the DNN that lead to the best-trained network. Four hyperparameters of the DNN were chosen as the variables that need to be found during the optimization process: the filter number of the first and second Convolution layer, and the kernel size of the first and second Convolution layer. The filter number was set to change between 10 and 150, and the kernel size was set to change between 1 and 10.

In order to obtain the results in a reasonable time, we fixed the maximum iterations (generations) in the GA optimization process. This number was chosen empirically to be 25. This means that after 25 iterations (generations), the best chromosome is defined as the optimum set of parameters that determine the optimal DNN for our application. Therefore, the stopping criterion for the optimization is reaching the maximum iteration (generation) number. A summary of the specifications of the implemented Genetic algorithm can be seen in Table 2. We have to note that 16 DNN's are trained in each generation of the optimization process (except for the first generation where 32 DNN's are trained), meaning that the algorithm needs to train 432 DNN's in total. Training each neural network is a time-consuming and costly procedure, and because of this, the optimization algorithm requires fast operating GPUs and a great deal of time. Adding another optimization parameter creates the need to expand the generations and/or population of each generation, requiring even more time and cost. The parameters related to the conv. layers gave us the needed degree of freedom to alter the characteristics of the neural networks in a significant manner and deal with the deciding factors of the model.

6 Results and Discussion

The optimization process was run on the online GPU provided by google Colab. The optimal answer obtained from this process is an array of (89, 89, 4, 1), which is the best chromosome at the end of the 25th generation. It is interesting to note that the filter number of the two convolution layers is the same, which is in line with commonly used networks such as VGG-Net. Hence, according to the GA results, we chose the filter number of 89 for the first and second Convolution layer and a kernel size of 4 for the first layers, and 1 for the second layer. Figure 9 presents the diagram of the mean fitness of the process across all generations. The mean fitness is the average fitness of all chromosomes in every iteration.

From Fig. 9, we can see that from the 5th iteration, the average fitness of the chromosomes has a higher value relative to the first few epochs; this means that the optimization process is actually keeping only the chromosomes that have the higher fitness. The mutation effect can be seen in the initial epochs, and it peaks in the 3rd iteration where the mean fitness of that generation is considerably lower than the other generations. The clear effect of the optimization is seen in Fig. 10, where the best chromosome in each generation is getting better over the generations or stays the same. There is a sharp improvement in the fitness of the best chromosome in the first 5 iterations, and then it remains even until it is followed by another improvement in the last iterations. The optimization process has been shown to be effective in finding the networks that have higher accuracies and fewer parameters. The fitness of the best DNN structure in the last

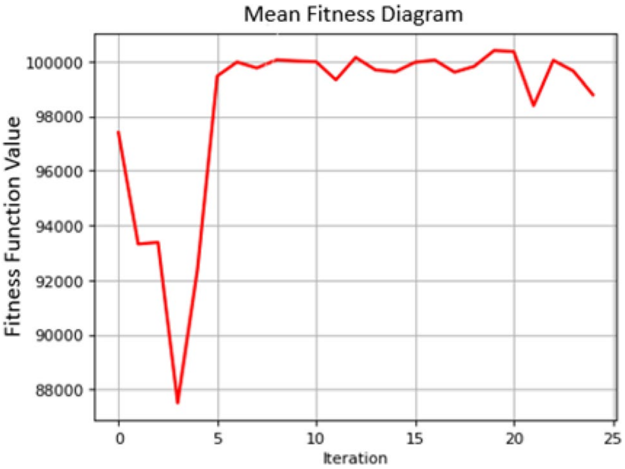


Fig. 9 The diagram of mean fitness over the optimization iterations. Each value indicates the average fitness value of all the chromosomes in an iteration

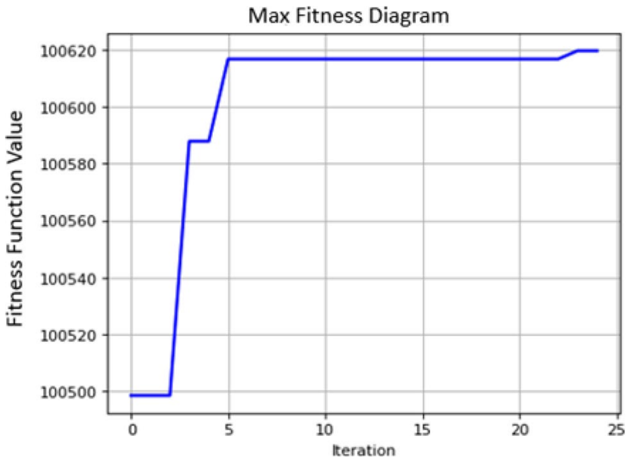


Fig. 10 The diagram of maximum fitness over the optimization iterations. Each value indicates the maximum fitness value of all the chromosomes in an iteration, which is equal to the fitness of the best chromosome of each generation

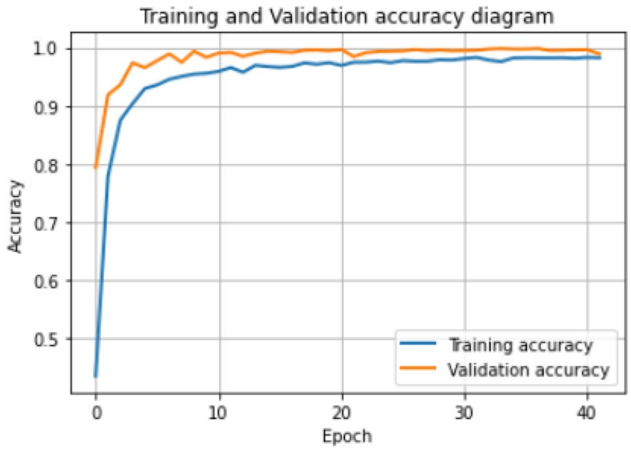


Fig. 11 The diagram of the training and validation accuracy of the optimal DNN over the training epochs

Table 3 The structure of the optimal DNN

Layer (type)	Output Shape	Param #
conv2d_l (Conv2D)	(21, 57, 89)	1513
max_pooling2d_l (MaxPooling2)	(7, 12, 89)	0
conv2d_2 (Conv2D)	(7, 12, 89)	8010
max_pooling2d_l (MaxPooling2)	(3,3,89)	0
dropout_l (Dropout)	(3,3,89)	0
flatten_l (Flatten)	(801)	0
dense_l (Dense)	(15)	12,030
Total params: 21,553		
Trainable params: 21,553		
Non-trainable params: 0		

generation is considerably higher than the initial proposed DNN structure.

The final DNN used for the dynamic ISL classification was built using the optimal parameters. This model had 21,553 parameters (weights). For the optimal model used for our application, we still used 20% of the data for verification, but only 72% for training and 8% for validation (we used the most data for the training stage). In this way, the number of the training data (which is 21,600) will be more than the model trainable parameters, which will reduce the occurrence of over-fitting. The properties of the final optimal DNN that was built and reported are presented in Table 3. The mean accuracy over the verification (test) data, not used in any stage of the training, is 99.7%. This is remarkably satisfying for our application. Although we had other DNNs that had fewer trainable parameters (fewer than 21,553) during the optimization process, these networks were not the optimal answer to our optimization problem since they end up with smaller overall accuracy. Thus, the optimal answer is the best trade-off between the fewest parameters and the highest overall accuracy (defined by Eq. 2). Please note that since half of the population is eliminated in the first iteration of the optimization process, the remaining networks in later generations always have acceptable accuracy.

These results show that the optimization process was quite successful at finding the network, which is absolutely appropriate for our application. Since it is remarkably accurate and also very light to work with, the RASA robot does not need to dedicate a large amount of memory and process to detect the ISL signs in real-time and is able to recognize the signs with an accuracy of more than 99%. Figure 11 shows the trend of the training and validation accuracy throughout the training in this model. The training process was terminated by the Early-Stopping mechanism at the 42nd epoch.

By looking at Fig. 11, we observe that the validation accuracy is always higher than the training accuracy; this might suggest that the validation set consists of easier samples than the training set or the model is overfitted. To assess this, we also trained the final model with a fivefold cross validation method, meaning that we shuffled the dataset and segmented it into 5 equally sized parts. Then, in each iteration, we held one part out for validation, and left the other parts for training the model, and repeated this process 5 times. The results are as follows: Iteration 1, training accuracy: 98.0%, validation accuracy: 99.1%-Iteration 2, training accuracy: 98.3%, validation accuracy: 99.7%-Iteration 3, training accuracy: 98.0%, validation accuracy: 99.6%-Iteration 4, training accuracy: 98.0%, validation accuracy: 99.4%-Iteration 5, training accuracy: 97.1%, validation accuracy: 98.9%. The average training accuracy over all folds is 97.9%, and the average validation accuracy over all folds is 99.3%. Therefore, the same pattern is observed in the cross-validation method,

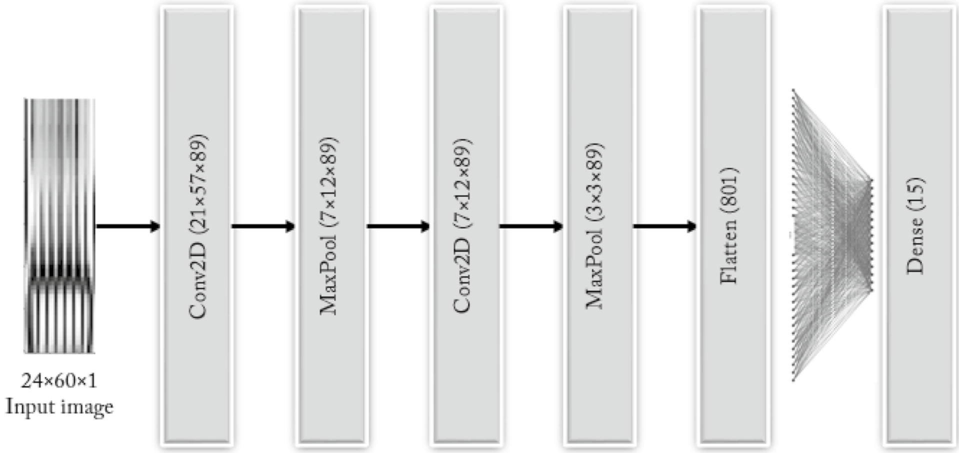


Fig. 12 The Schematics of the optimal DNN structure

Table 4 The confusion matrix of the optimal DNN used over the test dataset

P/A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	*100.00*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	*100.00*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	*99.48*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.52	0.00
3	0.00	0.00	0.24	*99.51*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	*100.00*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	*99.50*	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	*100.00*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*99.77*	0.00	0.00	0.00	0.00	0.00	0.23	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.48	*99.04*	0.00	0.24	0.00	0.24	0.00	0.00
9	0.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*99.01*	0.00	0.00	0.49	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*100.00*	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	*99.75*	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	*100.00*	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.26	*99.74*	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	*99.75*

where the validation accuracy is higher than the training accuracy in all iterations, even though the training and the validation sets are different in each iteration. These results suggest that the premise of having unbalanced sets or overfitting is rejected.

A schematic of the optimal DNN structure is presented in Fig. 12. As can be seen, the structure is composed of four 3D layers and two flat layers. Although the structure seems simple, its accuracy is quite high for our application of recognizing dynamic hand gestures. Table 4 represents the confusion matrix of the optimal DNN used over the test data. The matrix's rows represent the class ID predicted by the model, and its columns represent the real class ID related to the dataset. The diagonal values indicate, in a specific label, how much of the data was classified correctly. As seen, 6 out of 15 labels' classification accuracy is 100%, which means all of the data in these classes were predicted correctly. In the worst cases, which are label numbers 8 and 9, the classification accuracy are 99.0%. The average of the diagonal values is the same as the test accuracy, which is 99.7%.

Table 5 The accuracy and the number of epochs for the optimal DNN with respect to dropout rate

Dropout rate	Accuracy (%)	Number of epochs
0.2	99.7	42
0.35	99.8	57
0.5	99.6	78
0.7	98.4	116

We also trained the optimal network with different dropout values. Table 5 reports the accuracy and the number of epochs with respect to the dropout rate.

Table 6 A comparison between our work and similar research in the literature

Index	Reference	Hardware	Method	Number of classes	Signing type	Accuracy (%)
1	[20]	Leap-Motion	SVM + KNN	26	Static and isolated	79.83
2	[37]	Kinect	DBN + Le-Net	36	Static and continues	98.12
3	[36]	Camera	Convex hull + CNN	36	Static and continues	98.05
4	[41]	Sensor glove	M-KNN	40	Dynamic and isolated	98.90
5	[39]	Camera	HMM	20	Dynamic and isolated	97.48
6	[40]	EMG	Sliding window + DNN	17	Dynamic and isolated	83.23
7	This work	Sensor glove	State-image + DNN	15	Dynamic and isolated	99.7
8	[35]	Camera	VGG-Net/AlexNet + LSTM	450	Dynamic and continues	75.70

According to the mechanism used in this paper to train the optimal network, the logic of this table is reasonable. We can see that the final accuracy of the network does not change significantly with respect to the dropout rate, but the number of epochs the network needs to be trained alters significantly. The dropout technique temporarily deactivates a portion of trainable parameters in each epoch and reactivates them in later epochs. This helps the network avoid overfitting and "memorizing" the data. Increasing this rate will make it harder for the network to train since a larger portion of its parameters are deactivated at each step, and it will take a longer time for it to learn the data. We expect to see better accuracy on any unseen dataset when the network is trained with larger dropout rates. However, too large values can adversely affect the training process and leave the network untrained (i.e. underfitting).

We also tried using different learning rates for the ADAM optimization of the final model. We previously reported an accuracy of 99.7 in 42 epochs when the learning rate is 0.001; changing the value of the learning rate to 0.01 causes the network to diverge, and the DNN cannot be trained. On the other hand, changing the learning rate to 0.0001 and the network will reach an accuracy of 99.8 on the test dataset, but it will need 121 epochs. Therefore, without gaining significant improvement in accuracy, the training process becomes unnecessarily slow. This suggests that it is convenient to choose 0.001 as the order of magnitude for the learning rate.

RASA uses the sign recognition module in an internal architecture along with other processing modules when interacting with human users. The sign recognition process needs to be done fast and requires minimum memory since a large number of other modules are functioning at the same time. Models with a huge number of parameters require a great deal of time to process their input, and they usually possess a large memory, which leads to unacceptable real-time performance. Therefore, it was very important for us to implement a sign detection module that enables RASA to interact with human users in real-time. The model we used is implemented after training and has a relatively small number of parameters. Thus, it only takes a maximum of 2–3 s for it to recognize a sign performed in real-time, even when implemented with all the other functions on RASA's internal computer.

In summary, Table 6 provides a comparison between the proposed methodology and similar works in the literature. Works 1 to 3 obtained good recognition accuracy for a continuous sign performance used static signing (2 & 3), but this is not very useful in ISL. Unlike static signing, studies using dynamic signing (8) were unable to reach a high accuracy even though they deployed a very complex platform using heavy networks and RNNs. However, it should be noted that they tested their system on a large number of classes (450 isolated gesture classes tested by the continuous system). This is one of the limitations of our work, and we aim to increase the size of our dataset in future studies. Our study lies in the dynamic and isolated signing group and has a simpler methodology and higher accuracy compared to other works in this group. The table suggests that our work is less or as complex relative to the other works and yet it yields a higher accuracy, which is clearly due to our optimization process. For example, number 6 also uses DNN for feature extraction in analyzing sequential data. However, there are some important differences to this work; the authors in that study determined the hyperparameters of the network from experience rather than an optimization process, and this could be the reason for the significant difference between the accuracy of the two studies. Moreover, using signal mathematical post-processing techniques (e.g., calculating the signal mean and standard deviation, signal filtering, etc.) and a sliding-window mechanism (rather than the state-image method implemented in this work) is a disadvantage when dealing with real-time operations and leads to inaccurate classification. Comparing results with other studies that do not use sensor globes might be unfair since the quality of the input data from other sensors is much different. The comparison provided in this paper is meant only to deliver the readers preliminary insights on the position

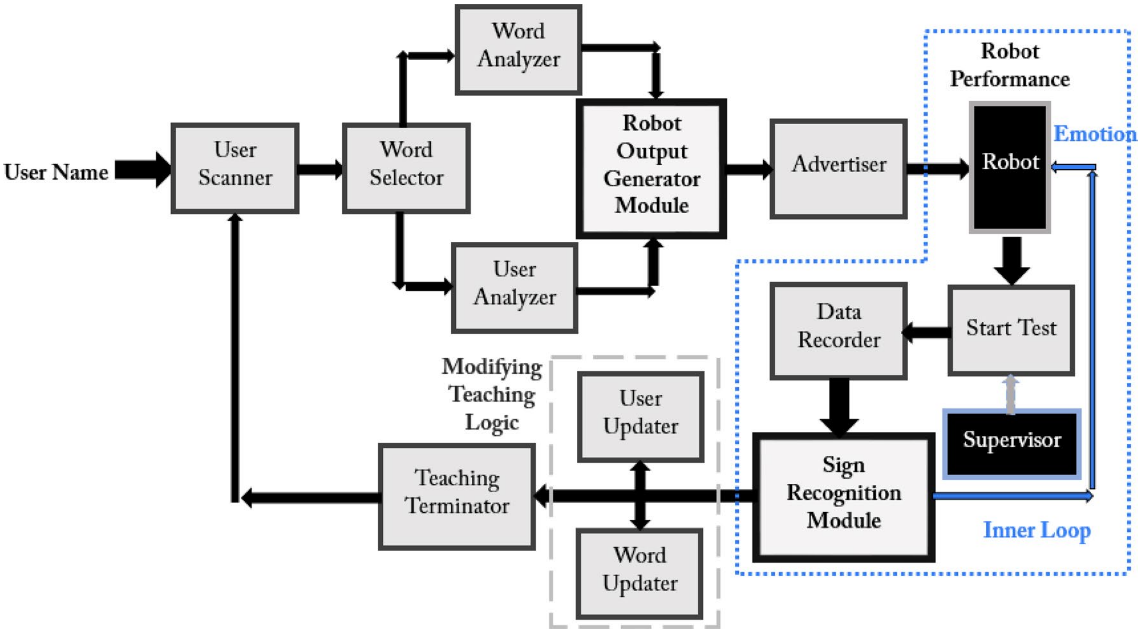


Fig. 13 RASA's adaptive teaching architecture

of this work in the literature and how different our proposed methodology is from similar works.

In this study, we classified 15 ISL signs, which are fewer than the other studies in Table 6. However, if we review the methodology of this work, we see that no stage of the process is dependent on the number of classes. If we wanted to classify more ISL signs (e.g. 20, 30, or more), we would still be able to preprocess the data using the state-image method and create a similar dataset (as the state-image preprocessing has no relationship with data labels). Then, the optimization process would propose a DNN that best classifies the new dataset, probably a network with a more complex structure, a higher number of trainable parameters, and maybe even a different learning method. In fact, the increased diversity and/or complexity in the dataset is compensated by more complexity in the optimal DNN's structure and learning, and we do not expect to see a dramatic performance loss when applying the methodology of this work to a dataset with more classes. To address this issue more reliably, we conducted an experiment to assess the system's performance on a bigger dataset. We collected more data for the previous signs and added five new signs to the original dataset, increasing the number of data in each class to 2250 using the same preprocessing method. The developed dataset consisted of 20 classes and 45,000 data in total. To assess the system's performance on the new dataset, we applied the same optimal DNN (calculated for the original dataset) to the developed dataset without doing the optimization process for the new dataset. All of the parameters of the network remained the same except for the last Dense layer, which was changed from 15 to 20 (since we have 20 classes in this case). Although we did not look for the optimal DNN for the new dataset, the results were still promising. The previous optimal DNN was able to train on the new dataset with an accuracy of 97.5%. This helps prove our claim that increasing the number of classes and/or the number of data in the dataset would not adversely affect the overall system's performance in a dramatic way.

7 Application in Human–Robot Interaction

So far in this paper, we have discussed the mechanism by which RASA can recognize ISL signs and determine performance accuracy. The whole mechanism is integrated into the Sign Recognition Module, which takes the recorded glove sensory data as input and returns the sign class and related accuracy as output. This module is part of a robotic architecture that RASA uses to perform adaptive ISL teaching. In this section, we briefly discuss this architecture and explain how the methodology of this paper is utilized in human–robot interaction.

7.1 Implementation on RASA and Experimental Setup

Figure 13 shows the general architecture designed for RASA to perform adaptive ISL teaching. Describing the aim of each module in this architecture is out of the scope of this paper; however, to provide useful insights on the real-time

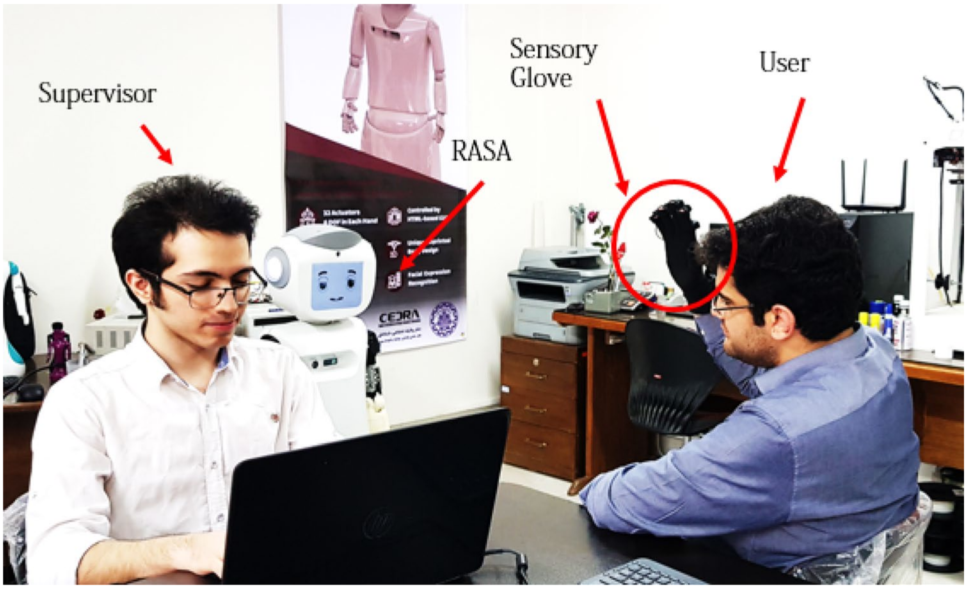


Fig. 14 The experimental setup for the first experiment

application of this paper we will explain how sign language recognition helps RASA to interact with human users.

In the inner loop of teaching, the robot first performs the Word Selected for each session according to the calculated output parameters and waits for the user. These parameters are calculated using a fuzzy-logic based approach that takes user parameters as input [53]. RASA is pre-learnt to perform a specific set of ISL signs using different approaches, all having a point-to-point path generation method for the robot's arm to reach a particular geometry; meaning that it uses a fixed pattern to perform any sign and hence, these gestures are not altered during the teaching process. Although RASA performs ISL signs for the users in order to teach them, it does not learn from the users how to alter its gestures (the only thing that RASA changes is the performance speed and the number of repetition). This waiting time after robot's performance is related to the user's readiness and the length of time for the robot's performance to teach the appropriate words, which is determined by the start test module. A supervisor helps the start test module to initiate. Therefore, a human user determines when the user should begin to perform and also when the performance has finished; this means the segmentation of real-time glove data is not automatic and is done by a human supervisor. This is why we have made our sign recognition process as robust as possible to missing/stagnation frames in the data augmentation stage. Next, the user utilizes the glove to perform the presented word, which the robot then evaluates as successful or not. The Data Recording Module stores the data transmitted from the data collection glove. The Sign Recognition Module determines what word the user has performed and its degree of accuracy using the mechanism described in this paper. Then, according to RASA's assessment on the user's performance, the teaching/user performance is repeated until the user reaches an acceptable performance, or totally fail at that sign. The users hopefully learn how to correctly perform that particular sign when the teaching is terminated.

Figure 14 shows a snapshot of the performed HRI setup, including a user wearing a sensory glove sitting in front of the robot. The supervisor initiates and supervises the teaching process. To enhance the robot's agility in interacting with human users during the teaching sessions, the sign recognition process is conducted in an external server, and the results are sent back to the robot for further processing (however, the whole process can also be implemented on RASA's computer); namely, the data-capturing glove is connected via a cable to the external server (in this case, the supervisor's computer) where the entire recognition process (state-image preprocessing + feeding to the DNN) takes place. Then, the results are transmitted via wi-fi and WebSocket protocol to RASA. Figure 15 gives a schematic view of this setup.

The mentioned robotic architecture shown in Fig. 13 is a fuzzy logic based architecture that makes it possible to adapt the robot's teaching output using four aspects based on the users' past and present performance: (1) What words should be selected for teaching, (2) How many times each word should be repeated in each training session, (3) What should be the speed of the performance of the signs, and 4) What should be the emotional valence of RASA to the user's performance? [53] Therefore, the sign recognition module is a remarkably significant element of this methodology for RASA to learn from the users and adjust its teaching based on their performance.

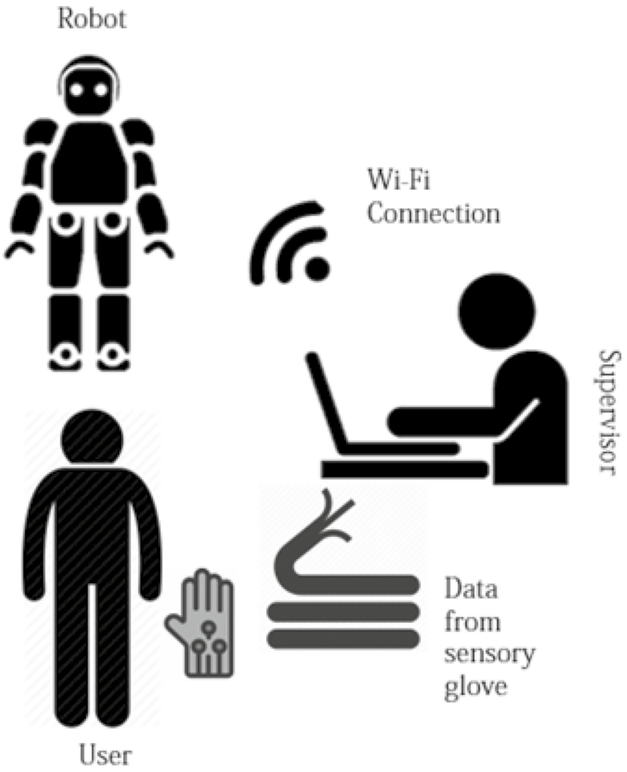
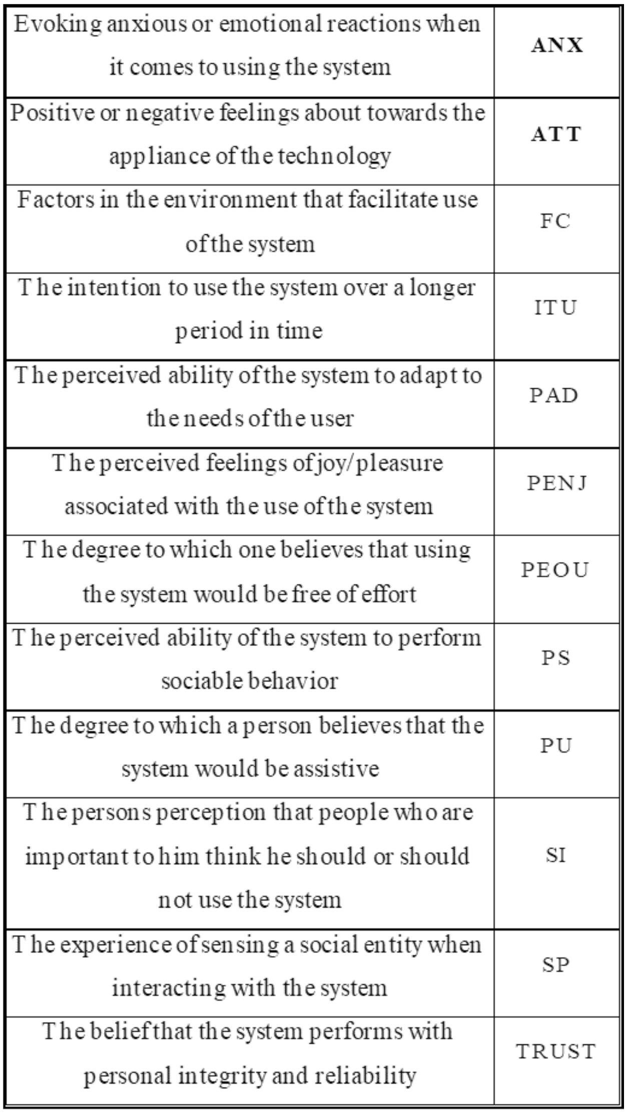


Fig. 15 A schematic of the experimental setup

7.2 HRI Assessment and User Evaluation

To assess whether the proposed methodology works effectively in human–robot interactions, we conducted an experiment that incorporates human users interacting with RASA in teaching sessions; namely, RASA teaches users ISL signs for a fixed amount of time and scores them according to the correctness and accuracy of their performance. It should be noted that the aim of this experiment is only to evaluate the users' attitude and satisfaction with the proposed methodology and to find out whether the robot's ability to recognize ISL signs in real-time is acceptable in the users view.

To be more specific, RASA taught ISL signs, each for 10 min, to 8 human users, who were then asked to fill out the standard Unified Theory of Acceptance and Use of Technology (UTAUT) questionnaire [54]. Each participant used a 5-point Likert scale (i.e., 1: totally negative, 2: negative, 3: neither positive nor negative, 4: positive, and 5: totally positive) to respond to the questions. Different items of the standard questionnaire helped us to evaluate the methodology of this work. The items investigated in the UTAUT standard questionnaire are shown in Fig. 16. Roughly half of the total participants had robotic knowledge, while the other half did not. There was an approximately equal gender balance, and the age range was from twenty to fifty.



Evoking anxious or emotional reactions when it comes to using the system	**ANX**
Positive or negative feelings about towards the appliance of the technology	**ATT**
Factors in the environment that facilitate use of the system	FC
The intention to use the system over a longer period in time	ITU
The perceived ability of the system to adapt to the needs of the user	PAD
The perceived feelings of joy/pleasure associated with the use of the system	PENJ
The degree to which one believes that using the system would be free of effort	PEOU
The perceived ability of the system to perform sociable behavior	PS
The degree to which a person believes that the system would be assistive	PU
The persons perception that people who are important to him think he should or should not use the system	SI
The experience of sensing a social entity when interacting with the system	SP
The belief that the system performs with personal integrity and reliability	TRUST

Fig. 16 Items of the UTAUT questionnaire

We reported four items that directly address the information we needed on the HRI assessment regarding this paper (see Fig. 16). For the ATT item, the users scored 3.75 on average with a standard deviation (STD) of 1.01 on 24 observations (8 participants*3 questions), meaning that the users hold an approximately positive attitude toward the robot's ability to interact with them using SL. For the PENJ item, the users scored 4.03 on average with a STD of 0.64 on 32 observations (8 participants*4 questions), indicating that they perceived on average positive feelings of joy associated with their interaction with RASA. However, for the PEOU item, the users scored 3.38 on average with a STD of 0.93 on 16 observations (8 participants*2 questions), indicating that a significant number of the users were not able to find the proposed interaction mechanism easy to use (this is mainly associated with the data-capturing system). Finally, for the

PU item, the users scored 3.94 on average with a STD of 0.66 on 16 observations (8 participants*2 questions). This shows that the users perceived on average that the mechanism RASA uses to interact with them is useful. For our 8 participants, the Cronbach's alpha for the ATT, PENJ, PEOU, and PU are 0.6, 0.6, 0.4, and 0.5, respectively. It should be noted that the small number of the participants in the HRI section of this study was a serious limitation which causes our preliminary exploratory findings to be considered as a proof for concept and an overall estimation regarding the developed HRI platform by using machine learning algorithms. While taking caution, we cannot make any strong claims based on this limited number of data. All in all, according to the statistical information discussed in this section, we preliminary conclude that the proposed methodology of this paper has been successful in developing an ISL sign recognition module that enables RASA to interact with users in an acceptable manner in terms of robustness, accuracy, and speed.

8 Limitations and Future Works

The work described in this paper is one of the first serious attempts in the domain of integrating Iranian Sign Language (ISL) with robotic systems and is, of course, susceptible to limitations and weaknesses. We conducted this study with the available resources we had at the robotics lab and a limited research grant.

The main limitations of this study were the small number of collected data as our database, the limited number of the chosen signs to be classified, and the lack of the robot's ability to recognize continuous sign in a sentence/phrase. Currently, the RASA robot can only detect each individual chosen word when the start/end time of the performed signs are sent to RASA by the robot operators. The dataset used in this study consisted of 30,000 images. We used several techniques in data preprocessing and training the model to avoid overfitting as much as possible. We also kept the trainable parameters of the final model as low as possible (fewer than the number of our data) to compensate for the size of the dataset and the simplicity of our model. In our next studies, we would like to increase the number of performing samples for each sign in the dataset as well as the number of chosen signs to enrich the vocabularies of the robot. Our future work will also be empowering the robot to recognize the words during online sentences/phrases.

To get a realistic test result, the test data should be obtained from 'other' users, and although the test data is not included in any stage of the entire project, this paper uses data from the same user who created the training data. Also, we have used augmented data to report the final accuracy, which is not normally done. We intend to do a more thorough future experiment with an expanded dataset from new human subjects without presenting the data to the network in the training step.

Furthermore, the data-capturing system in this study is a sensory-glove, which can produce discomfort in users interacting with the system. Although sensory gloves generate more reliable output data compared to other data capturing systems, they are also a disadvantage as they can adversely affect the users' fluency while performing SL signs. As a preliminary exploratory finding, the users' feedback (which we discussed in Sect. 7.2) indicates that the average perceived ease of use of the system is negatively influenced by the data-capturing system, as they might encounter difficulties performing signs normally, in terms of naturality and speed.

Finally, applying the state-image method omits the effect of performance speed as it down-samples all input signals into a 60 frame window. Thus, if two ISL signs have the exact same hand and fingers gestures but differ in performance speed, they would probably be indistinguishable by the state-image method. Also, the data-collection setup can only report the positional and the rotational data of the hand and fingers. We do not have the instrument to capture the forces exerted on the hand and/or fingers' muscles during the performance of the sign; therefore, we have not used them in any stage of the study. However, to the best of our knowledge, performance speed and muscle forces are not deciding factors to determine the meaning (in our application, the class) of ISL signs, and thus, this limitation is not a serious problem for our particular application.

9 Conclusion

In this paper, a platform was proposed for automatic ISL sign/word recognition using a Data Capturing Glove on the RASA robot as a practical application of machine learning in social human–robot interaction. As the first step of this study, the data gathering process, as well as the methods for making the input data, robust with regard to the environmental and users' conditions, was done. Fifteen ISL classes were chosen to investigate the platform's recognition accuracy, and forty performances were recorded for each selected word (i.e., creating a dataset including 600 data). Next, the data was augmented via the methods described in the paper to produce a modified dataset with 30,000 data. It should be noted that the generated data also became more robust to certain deficiencies. Then, we coded all of the input data as an image, turning the dynamic nature of the hand gesture detection problem into a static image processing problem. After describing the preprocessing stage, the main structure of the used DNN and the training method was presented in detail. To optimize the performance of the system for higher overall accuracy and less trainable parameters, four hyperparameters of the network, including the number of filters in the first and second convolution layers as well as the kernel sizes, were innovatively optimized via a Continuous Genetic algorithm. The main characteristics of the optimization algorithm were described in the manuscript. After the optimization process, the final structure of the DNN was determined, and its performance was investigated. We observed a notable mean performance accuracy of 99.7% for our proposed DNN on the test data in our dataset, which is quite promising. The DNN also has slightly less than 21,600 parameters, which helps our robot to detect ISL signs and continue teaching in real-time since the network does not require a large memory or processing power. Therefore, with a simpler methodology, compared to the state-of-the-art practice, we were able to implement a platform that can classify ISL dynamic gestures with quite a high accuracy, it is also remarkably robust against different users and/or environmental conditions. After implementing the sign recognition module in a robotic architecture, we conducted an HRI experiment to assess the system's performance in real-time applications. After an initial statistical analysis on the standard UTAUT model with the small number of the participants (as a preliminary analysis for the proof of concept), it was revealed that the system enabled RASA to interact with human users in an acceptable manner in terms of robustness, speed, and accuracy. The proposed methodology can also be easily implemented to detect any other hand gesture patterns, which allows us to interact with machines in the Human–Computer Interaction (HCI) and Human–Robot Interaction (HRI) contexts.

Acknowledgements This research was supported by the "Iranian National Science Foundation (INSF)" (http://en.insf.org/). We also appreciate the "Dr. AliAkbar Siassi Memorial Grant Award" for their complementary support of the Social & Cognitive Robotics Laboratory.

Authors' contributions All authors contributed to the study conception and design, material preparation, data collection and analysis were performed by Salar Basiri and Alireza Taheri. The first draft of the manuscript was written by Salar Basiri and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Author Alireza Taheri has received research grants from the "Iranian National Science Foundation (INSF)" (Gant No. 98025100).

Data availability All data from this project (videos of the sessions, results of the questionnaires, scores of performances, etc.) are available in the archive of the Social & Cognitive Robotics Laboratory.

Code availability All of the codes are available in the archive of the Social & Cognitive Robotics Laboratory. In case the readers need the codes, they may contact the corresponding author.

Declarations

Conflict of interest The authors Salar Basiri, Ali Meghdari, Mehrdad Boroushaki, and Minoo Alemi declare that they have no conflict of interest.

Ethics approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. Ethical approval for the protocol of this study was provided by the Iran University of Medical Sciences (#IR.IUMS.REC.1395.95301469).

References

1. W. H. O. (WHO). "Deafness and hearing loss." https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed 2019–04–27.
2. Marschark M, Hauser PC (2011) How deaf children learn: what parents and teachers need to know. Oxford University Press, USA
3. Courtin C (2000) The impact of sign language on the cognitive development of deaf children: the case of theories of mind. J Deaf Stud Deaf Educ 5(3):266–276. https://doi.org/10.1093/deafed/5.3.266
4. M. Zakipour, A. Meghdari, and M. Alemi, (2016) RASA: A low-cost upper-torso social robot acting as a sign language teaching assistant, presented at the International Conference on Social Robotics.
5. S. R. Hosseini, A. Taheri, A. Meghdari, and M. Alemi, (2019) Teaching persian sign language to a social robot via the learning from demonstrations approach, Presented at the International Conference on Social Robotics.
6. Meghdari A, Alemi M, Zakipour M, Kashanian SA (2018) Design and realization of a sign language educational humanoid robot. J Intell Rob Syst 95(1):3–17. https://doi.org/10.1007/s10846-018-0860-2
7. Karami A, Zanj B, Sarkaleh AK (2011) Persian sign language (PSL) recognition using wavelet transform and neural networks. Expert Syst Appl 38(3):2661–2667. https://doi.org/10.1016/j.eswa.2010.08.056
8. A. Kiani Sarkaleh, F. Poorahangaryan, B. Zanj, and A. Karami, (2009) A Neural Network based system for Persian sign language recognition, presented at the 2009 IEEE International Conference on Signal and Image Processing Applications, Kuala Lumpur, Malaysia.
9. Starner T, Weaver J, Pentland A (1998) Real-time American sign language recognition using desk and wearable computer based video. IEEE Trans Pattern Anal Mach Intell 20(12):1371–1375. https://doi.org/10.1109/34.735811
10. P. V. V. Kishore, M. V. D. Prasad, C. R. Prasad, and R. Rahul, (2015) 4-Camera model for sign language recognition using elliptical fourier descriptors and ANN, presented at the 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India.
11. Oz C, Leu MC (2011) American Sign Language word recognition with a sensory glove using artificial neural networks. Eng Appl Artif Intell 24(7):1204–1213. https://doi.org/10.1016/j.engappai.2011.06.015
12. S. A. Mehdi and Y. N. Khan, (2002) Sign language recognition using sensor glove, In Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02, Singapore, Singapore: IEEE, https://doi.org/10.1109/ICONIP.2002.1201884.

13. R.-H. Liang, (1998) A real-time continuous gesture recognition system for sign language, In Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan: IEEE, https://doi.org/10.1109/AFGR.1998.671007.
14. A. Agarwal and M. K. Thakur, (2013) Sign language recognition using Microsoft Kinect, presented at the 2013 Sixth International Conference on Contemporary Computing (IC3), Noida, India 8–10 Aug. 2013.
15. Z. Zafrulla, B. Brashear, S. Starner, H. Hamilton, and P. Presti, (2011) American sign language recognition with the kinect, in ICMI '11 Proceedings of the 13th international conference on multimodal interfaces Alicante, Spain, https://doi.org/10.1145/2070481.2070532.
16. S. Lang, M. Block, and R. Rojas, (2012) Sign Language Recognition Using Kinect, Presented at the International Conference on Artificial Intelligence and Soft Computing ICAISC 2012.
17. Zahedi M, Manashty AR (2011) Robust sign language recognition system using ToF depth cameras. World Comput Sci Inf Technol J (WCSIT) 1(3):50–55 (**arXiv:1105.0699**)
18. S. Oprisescu, C. Rasche, and B. Su, (2012) Automatic static hand gesture recognition using ToF cameras, In 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 Aug. 2012.
19. L. E. Potter, J. Araullo, and C. Carter, (2013) The Leap Motion controller: a view on sign language, In Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration Adelaide, Australia, https://doi.org/10.1145/2541016.2541072.
20. C.-H. Chuan, E. Regina, and C. Guardino, (2014) American Sign Language Recognition Using Leap Motion Sensor, In 2014 13th International Conference on Machine Learning and Applications, Detroit, MI, USA: IEEE, https://doi.org/10.1109/ICMLA.2014.110.
21. M. Mohandes, S. Aliyu, and M. Deriche, (2014) Arabic sign language recognition using the leap motion controller, Presented at the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey.
22. H. Brashear, T. Starner, P. Lukowicz, and H. Junker, (2003) Using multiple sensors for mobile sign language recognition, In Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings, White Plains, NY, USA, USA, https://doi.org/10.1109/ISWC.2003.1241392.
23. Yang HD (2014) Sign language recognition with the Kinect sensor based on conditional random fields. Sensors (Basel) 15(1):135–147. https://doi.org/10.3390/s150100135
24. Gao WEN, Ma J, Wu J, Wang C (2000) Sign language recognition based on Hmm/Ann/Dp. Int J Pattern Recognit Artif Intell 14(05):587–602. https://doi.org/10.1142/s0218001400000386
25. S. K. Yewale and P. K. Bharne, (2011) Hand gesture recognition using different algorithms based on artificial neural network, Presented at the 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Udaipur, India.
26. Izzah A, Suciati N (2014) Translation of sign language using generic fourier descriptor and nearest neighbour. Int J Cybern Inf 3(1):31–41. https://doi.org/10.5121/ijci.2014.3104
27. Ansari ZA, Harit G (2016) Nearest neighbour classification of Indian sign language gestures using kinect camera. Sadhana 41(2):161–182. https://doi.org/10.1007/s12046-015-0405-3
28. J. Ye, H. Yao, and F. Jiang, (2004) Based on HMM and SVM multilayer architecture classifier for Chinese sign language recognition with large vocabulary, Presented at the Third International Conference on Image and Graphics (ICIG'04), Hong Kong, China, China.
29. Subashini TS, Nagarajan S (2013) Static hand gesture recognition for sign language alphabets using edge oriented histogram and multi class SVM. Int J Comput Appl 82(4):28–35. https://doi.org/10.5120/14106-2145
30. BPP. Kumar and MB. Manjunatha (2017) A Hybrid Gesture Recognition Method for American Sign Language, Indian Journal of Science and Technology, 10(1) https://doi.org/10.17485/ijst/2017/v10i1/109389
31. O. Koller, S. Zargaran, R. Schlüter, and R. A Bowden, (2016) Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition, In The British Machine Vision Conference (BMVC), York, https://doi.org/10.5244/C.30.136.
32. A. Krizhevsky, I. Sutskever, and G. E. Hinton, (2012) ImageNet classification with deep convolutional neural networks, Presented at the Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1, Lake Tahoe, Nevada.
33. Y. LeCun, Y. Bengio, and G. Hinton, (2015) Deep learning, Nature, vol. 521, p. 436, 05/27/online 2015, https://doi.org/10.1038/nature14539.
34. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
35. Cui R, Liu H, Zhang C (2019) A deep neural framework for continuous sign language recognition by iterative training. IEEE Trans Multimedia 21(7):1880–1891. https://doi.org/10.1109/tmm.2018.2889563
36. M. Taskiran, M. Killioglu, and N. Kahraman, (2018) 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), Athens, Greece: IEEE, https://doi.org/10.1109/TSP.2018.8441304.
37. Tang A, Lu K, Wang Y, Huang J, Li H (2015) A real-time hand posture recognition system using deep neural networks. ACM Trans Intell Syst Technol 6(2):1–23. https://doi.org/10.1145/2735952
38. Oyedotun OK, Khashman A (2016) Deep learning in vision-based static hand gesture recognition. Neural Comput Appl 28(12):3941–3951. https://doi.org/10.1007/s00521-016-2294-8
39. Azar SG, Seyedarabi H (2020) Trajectory-based recognition of dynamic Persian sign language using hidden Markov model. Comput Speech Lang 61:101053. https://doi.org/10.1016/j.csl.2019.101053
40. K. Xing et al., (2018) Hand Gesture Recognition Based on Deep Learning Method, Presented at the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China.
41. 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
42. Dong Y (2018) An application of Deep Neural Networks to the in-flight parameter identification for detection and characterization of aircraft icing. Aerosp Sci Technol 77:34–49. https://doi.org/10.1016/j.ast.2018.02.026
43. Dong Y (2019) Implementing Deep Learning for comprehensive aircraft icing and actuator/sensor fault detection/identification. Eng Appl Artif Intell 83:28–44. https://doi.org/10.1016/j.engappai.2019.04.010
44. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003
45. van Dyk DA, Meng X-L (2001) The art of data augmentation. J Comput Graph Stat 10(1):1–50. https://doi.org/10.1198/10618600152418584
46. Barbu T (2013) Variational image denoising approach with diffusion porous media flow. Abstr Appl Anal 2013:1–8. https://doi.org/10.1155/2013/856876
47. F. Chollet. "Keras." https://keras.io/ (accessed 2019–07–23).

48. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
49. J. Ba and D. P. Kingma, "Adam: A Method for Stochastic Optimization," presented at the the 3rd International Conference for Learning Representations, San Diego, 2015.
50. J. Brownlee. "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning." https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ (accessed 2019–30–07, 2019).
51. "ML Cheatsheet " https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html (accessed 2019–02–08.
52. M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1996.
53. Basiri S, Taheri A, Meghdari A, Alemi M (2021) Design and implementation of a robotic architecture for adaptive teaching: a case study on iranian sign language. J Intell Robot Syst 102(2):48. https://doi.org/10.1007/s10846-021-01413-2
54. Williams MD, Rana NP, Dwivedi YK (2015) The unified theory of acceptance and use of technology (UTAUT): a literature review. J Enterp Inf Manag 28(3):443–488

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Salar Basiri received his B.S. degree in Mechanical Engineering from Sharif University of Technology (SUT), Tehran, Iran in 2018. He received his M.S. degree in Mechatronics from Sharif University of Technology (SUT), Tehran, Iran in 2020. His research interests include Mechatronics, Artificial Intelligence, and Machine Learning.

Alireza Taheri is an Assistant Professor of Mechanical Engineering with an emphasis on Social and Cognitive Robotics at Sharif University of Technology, Tehran, Iran. He is the Head of the Social and Cognitive Robotics Lab at Sharif University of Technology.

Ali F. Meghdari is a Professor Emeritus of Mechanical Engineering and Robotics at Sharif University of Technology (SUT) in Tehran. Professor Meghdari has performed extensive research in various areas of robotics; social and cognitive robotics, mechatronics, bio-robotics, and modeling of biomechanical systems. He has been the recipient of various scholarships and awards, the latest being: the 2012 Allameh Tabatabaei distinguished professorship award by the National Elites Foundation of Iran (BMN), the 2001 Mechanical Engineering Distinguished Professorship Award from the Ministry of Science, Research & Technology (MSRT) in Iran, and the 1997 ISESCO Award in Technology from Morocco. He is the founder of the Centre of Excellence in Design, Robotics, and Automation (CEDRA), an affiliate member of the Iranian Academy of Sciences (IAS), a Fellow of the American Society of Mechanical Engineers (ASME), and the Founder and Chancellor of Islamic Azad University-Fereshtegaan International Branch (for students with special needs; primarily the Deaf).

Mehrdad Boroushaki received his B.Sc. degree in Electronics engineering from Gilan Univer-sity, Iran in 1989. He also received his MS.c. and Ph.D. degrees in Mechanical engineering from Sharif University of Technology, Tehran, Iran, in 1994 and 2002, respectively. He is currently associate professor of energy engineering at the Sharif University of Technology. His research in-terests include the application of machine learning in mechanical engineering and energy engineering systems fields, identification, optimization and control of complex nonlinear systems using intelligent systems, and renewable energy systems. He is also a Senior Member of the IEEE since 2014.

Minoo Alemi received her Ph.D. in Applied Linguistics from Allameh Tabataba'i University in 2011. She is currently an Associate Professor of Applied Linguistics at the Islamic Azad University, West-Tehran Branch. She is the co-founder of Social Robotics in Iran, a title she achieved as a Post-Doctoral research associate at the Social Robotics Laboratory of the Sharif University of Technology. Her areas of interest include discourse analysis, interlanguage pragmatics, materials development, and RALL. Dr. Alemi has been the recipient of various teaching and research awards from Sharif University of Technology, Allameh Tabataba'i University, Islamic Azad University, and Int. Conf. on Social Robotics (ICSR-2014).