Continuous sign language recognition from wearable IMUs using deep capsule networks and game theory ☆

Karush Suri, Rinki Gupta*

Amity University, Sector-125, Noida, Uttar Pradesh, Noida, Uttar Pradesh 201313, India

ARTICLE INFO

ABSTRACT

Sign Language is used by the deaf community all over world. The work presented here proposes a novel one-dimensional deep capsule network (CapsNet) architecture for continuous Indian Sign Language recognition by means of signals obtained from a custom designed wearable IMU system. The performance of the proposed CapsNet architecture is assessed by altering dynamic routing between capsule layers. The proposed CapsNet yields improved accuracy values of 94% for 3 routings and 92.50% for 5 routings in comparison with the convolutional neural network (CNN) that yields an accuracy of 87.99%. Improved learning of the proposed architecture is also validated by spatial activations depicting excited units at the predictive layer. Finally, a novel non-cooperative pick-and-predict competition is designed between CapsNet and CNN. Higher value of Nash equilibrium for CapsNet as compared to CNN indicates the suitability of the proposed approach.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Human motion analysis has been studied for various applications using different sensing technologies [1–5]. For instance, human gait may be analysed to localize the phases in a gait that may be used for health monitoring or navigation purpose. The use of an inertial measurement unit (IMU) is very common in human gait analysis because of its low-cost and wearability. An IMU consists of a tri-axial accelerometer and a tri-axial gyroscope that measure the acceleration and turn rate along the *x*-, *y*- and *z*-axis of the sensor. Hence, an IMU provides the capability to assess the motion of an object in three-dimensional cartesian space. Measurement in three-dimensional coordinates also plays a significant role in accurate recognition of hand gestures. Hand-gesture recognition has been applied for designing human-machine interfaces [1,2]. Again, IMUs are useful for such system because they may be processed to develop real-time systems that could be used to control a computer interface during a presentation [1] or control a virtual-reality head-mounted display based on real-time tracking of upper limb [2].

Hand motion analysis has also been studied for the development of assistive technology such as sign language recognition. Sign language predominantly involves the use of various hand postures and motions. Consequently, different sensing technologies have been employed in analysis of human motion for recognition of the signed word or sentence. These include depth sensors [3], videos [4] as well as wearable sensors such as surface electromyogram (sEMG) and IMU. Signs may be

---

recognized from images or video frames by acquisition of the hand posture at an appropriate angle, followed by image segmentation and prediction [3,4]. However, images may not always be suitable for gesture recognition since their acquisition system may not be wearable and images may be affected by illumination condition, foreground focus and background complexity. Hence, wearable sensors may be used for acquiring data while signing. Surface electromyogram measures the muscle potentials when different muscles are activated to perform a specific sign. Several features have been proposed for sEMG signals that may distinguish one hand activity from another. Although they are non-invasive and provide complementary capability as compared to IMUs for hand gesture recognition; sEMG are easily affected by factors such as sensor placement, motion artefacts and subject variability. IMUs estimate the position and orientation of the hand with minimum invasiveness and have a compact design. In this work, an algorithm is proposed for recognition of sentences signed according to the Indian sign language using data from IMU placed in a forearm armband.

While conversing in sign language, a signer performs gestures in a sequence which result in the formation of a complete sentence. The signs may be evaluated in continuation by means of position and orientation of the hand based on the signals recorded with IMUs. Then, the signal patterns may be analysed by making use of advanced deep models and by exploiting the hierarchical structure of sign language itself to predict the performed sign and hence, translate the signed gestures in the sentence in any verbal language such as English. Recognition of gestures from the recorded signals has been performed using conventional machine learning as well as deep learning techniques [5,6]. Most commonly found algorithms in literature are Artificial Neural Networks (ANNs), which are multilayer perceptrons based on the concept of stacked Restricted Boltzmann Machines (RBMs). ANNs provide a simple framework for layer-based structured classification by making use of feed-forward action and backpropagation algorithm [7]. Introduction of multiple layers in ANN tends to make the network deeper in order to enhance its learning. This gives rise to Deep Neural Networks (DNNs). However, as a result of multiple stacked layers in the structure, ANNs and DNNs suffer from vanishing gradients during training [8]. A suitable alternative to tackle this issue is provided by Recursive Neural Networks (RNNs). RNNs employ recursive layers which can be used time and again for the purpose of classification and generation as well [9]. These are often used to classify sequential data as a result of their recursive nature. Most of the times RNNs are used as generational models which provide higher training times and complex algorithms in collaboration to augment the training process [10]. This generates a need for models capable of improved learning and accurate predictions. Convolution layers for feature learning address this problem. Successive convolution and pooling of data in subsample space augment feature learning, which results in accurate classification. Convolution Neural Networks (CNNs) make use of these layers in a hierarchical manner [11]. Based on two-dimensional convolution, CNNs are excessively used in image processing applications for digit and object recognition [11]. CNN have also been applied on one-dimensional data and shown to yield good classification accuracy of 97.5% [12]. In classification of objects using images, CNNs may provide misclassification when the orientation of the image is altered or when there is a deviation from the expected spatial structure of pixels [13]. Thus, one must aim for equi-variance by making use of Capsule Networks (CapsNet) as proposed by Hinton et'al, which make use of nested group of neurons to construct a capsule [14]. CapsNet decreases layers and errors thereby increasing the prediction accuracy by making use of higher dimensional spacing in capsules. Also, dynamic routing between capsules can further be used to enhance recognition [15]. So far, CapsNet has been applied only for image-based recognition and its utility for multiple data types and dimensions is not reported in literature to the best of our knowledge.

Advanced models for classification are useful only if they optimize the performance. Performance of multiple state-of-the-art classifiers may be compared in terms of quantitative parameters such as classification accuracies, number of false predictions and learning rate. Alternatively, interactive games may be designed wherein the classifiers compete with each other to present better performance. For instance, General Adversarial Networks (GANs) optimize their choice of strategy in order to yield better generational and classification results [16]. Similar to GANs, multi-layered structures may comprise of different learning games which allow the model to study the data better as a result of competition [17].

In this work, the problem of sign language translation is addressed by making use of a novel CapsNet architecture consisting of one-dimensional convolution and dynamic routing. Signed sentences comprising of gestures recorded using a custom-designed wearable IMU device are classified using the CapsNet architecture consisting of capsules having dimensions higher than input data. The details of the recorded dataset and the proposed CapsNet architecture are explained in Section 2. Performance of the architecture is compared to the foundational CNN having similar trainable hyper-parameters. Furthermore, validation of accurate predictions is obtained by Game Theory consisting of construction of a non-cooperative pick game in which both the models compete with each other to offer a correct prediction of the input sample by picking the correct class, as explained in Section 3. The single-strategy competition serves as the basis for independent performance and sample-by-sample comparison allowing both the models to reach their best response. The results obtained with actual data are presented in Section 4 and Section 5 contains the concluding remarks.

2. Signed sentence recognition using capsnet

2.1. Data corpus

The dataset of IMU signals was constructed by making use of the GY-80 multiboard and Arduino UNO board, which contains the ATmega328P microcontroller. The GY-80 board consists of tri-axial accelerometer ADXL345 and gyroscope L3G400D, which measure acceleration (in m/s^2) and turn rate (in deg/s), respectively at 100 Hz. Both accelerometer and

Fig 1. (a) GY-80 sensor board, (b) Placement of experimental apparatus.

gyroscope have 3 degree-of-freedom each, giving a total of 6 channels of data. Fig. 1a depicts the GY-80 board consisting of IMU instruments. The setup is placed on wearable bands including straps. This is done for convenient usage of the apparatus and a compact design which would prevent any hindrances during hand motion. The setup is placed on the frontal side of the forearm, below the elbow as depicted in Fig. 1b. The sensors were calibrated for bias and scale factor. Accelerometer data was corrected by using the 12-parameter estimation method [18]. On the other hand, rotational signals were corrected by subtracting bias values and then scaling the sequence [19].

Signals from IMU sensors were recorded from a total of 10 different subjects out of which 5 are female and 5 are male. All the subjects fall in the age range of 21–49 years and 2 subjects were left-handed which makes the constructed dataset heterogeneous with respect to subject's age, gender and dominant arms. Signs were recorded from the Indian Sign Language (ISL) in a sequence, resulting in the formation of a complete sentence [20]. A total of 20 sentences were gathered with each sentence consisting of 2–4 signs spaced approximately 1 s apart. Each subject performed 10 repetitions of a sentence. In ISL, sentences follow a predefined structure. Interrogative words, number quantities and negations are generally signed at the end of the sentence whereas subjects and objects are signed at the beginning. For instance, for the sentence 'I want water', the gesture corresponding to 'I' is performed first followed by gesture corresponding to 'want' followed by gesture corresponding to 'water'. However, in the interrogative sentence 'Where is the Clinic?', gesture corresponding to 'Clinic' is performed first followed by gesture corresponding to 'Where'.

The recorded signals are further processed using a moving median filter to remove any spurious overshoots in the signals introduced because of the recording setup. Euler Angles '*E*' are obtained from accelerometer and gyroscope using (1) (2) and (3), respectively, as

$$A_\alpha = a\tan\frac{A_x}{\sqrt{A_y^2 + A_z^2}},\tag{1}$$

$$A_\rho = a\tan\frac{A_y}{\sqrt{A_x^2 + A_z^2}},\tag{2}$$

and

$$G(\alpha, \rho, \theta) = \int_0^T G(x, y, z)dt.\tag{3}$$

In (1) and (2), the roll (A_α) and pitch (A_ρ) angles are estimated using accelerations, A_x, A_y and A_z measured along *x*-, *y*- and *z*-axes, respectively. In (3), $G(\alpha,\rho,\theta)$ indicates the Euler angles estimated from turn rates $G(x,y,z)$ and '*dt*' indicates the sampling interval and '*T*' denotes the total duration of time. The Euler angles evaluated from accelerometer and gyroscope are combined in a complementary filter to evaluate the final angles as

$$E = \beta * G(\alpha, \rho, \theta) + (1 - \beta) * A(\alpha, \rho, \theta).\tag{4}$$

Here, '*β*' indicates the filter coefficient which is selected empirically as 0.85. Euler angles along with the pre-processed accelerometer and gyroscope signals are used in the CapsNet algorithm for classification, as explained below.

2.2. CapsNet architecture

The use of capsule theory addresses the need for accurate learning algorithms with a fool-proof structure. Two dimensional CaspNet eliminates the problem of rotational confusion present in CNN wherein ambiguous spatial relationships result in a poor performance of the model. In the case of one-dimensional convolution, capsule theory aids in providing a better mapping between the convolved feature set. Nesting of several convolutional layers within a layer gives rise to vectorized structures known as capsules. These capsules act as higher dimensional entities during feature learning by making use of
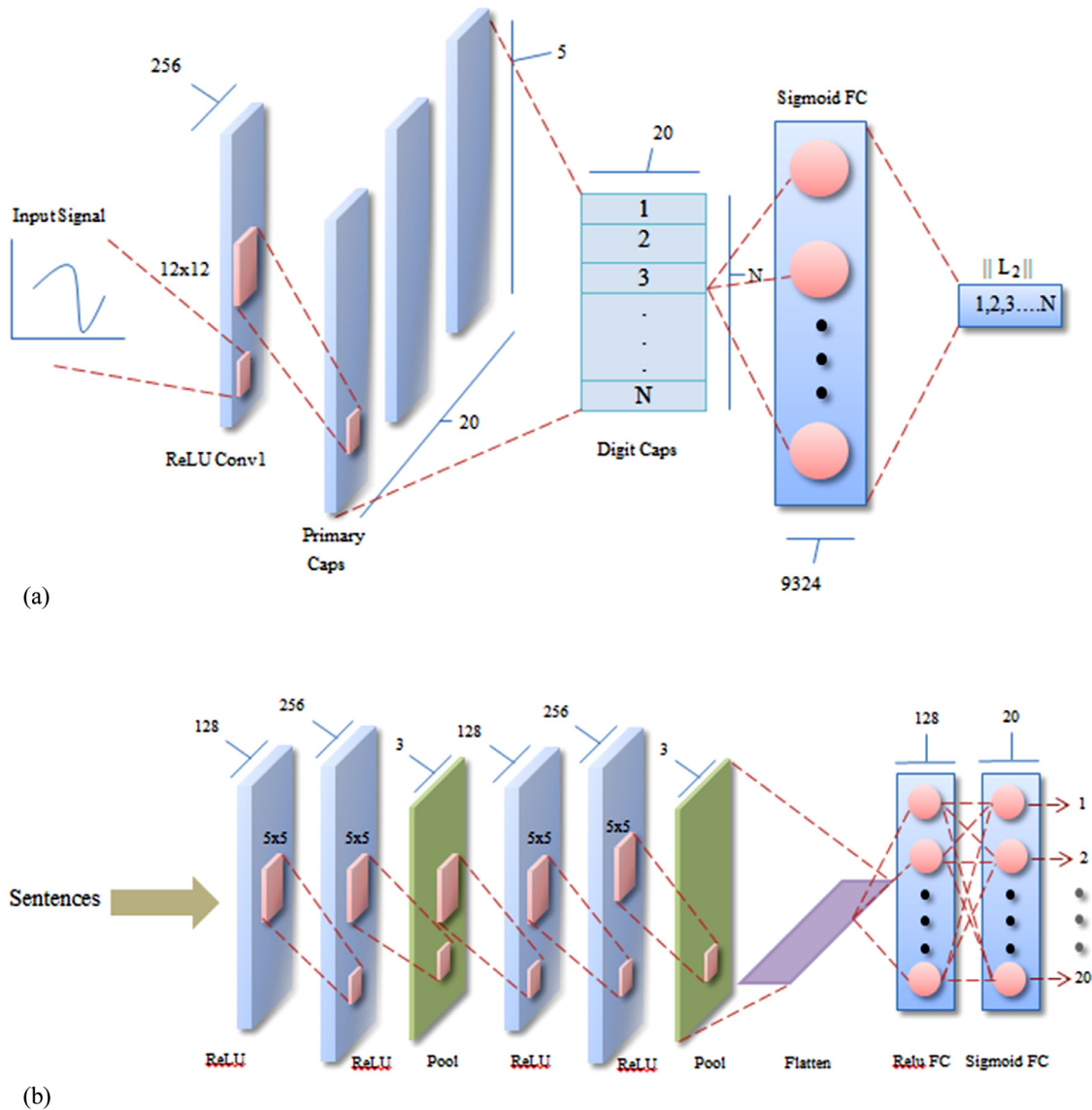
Fig 2. (a) One-dimensional CapsNet architecture for signed sentence recognition. (b) One-dimensional CNN architecture.

non-linear activations. Unlike the foundational CNNs, excessive convolution takes place in the layers and the need for pooling the outputs into a sub-sample space is prevented. Fig 2(a) provides the architecture for the one-dimensional CapsNet used in the recognition of IMU signals in contrast to the architecture of CNN which is depicted in Fig 2(b). In the CapsNet model, convolution of samples at nested primary capsule layer is followed by the Digit capsule (caps) layer. Digit caps layer acts as the higher dimensional feature space for the convolved values to be learnt. The digit caps layer accepts inputs from all the capsules in the previous layer. Non-linear activations at both the primary and digit caps layer are provided by means of the squash function as proposed in the original work on CapsNet [14]. Connections between these two layers are dynamic and are governed by the usage of dynamic routing. Based on the number and intensity of units excited in the next layer, data is routed for learning. This process of routing is governed by logits or simply probabilities denoting the coupling of two successive capsules between two consecutive layers. The coupling coefficients are determined iteratively by making use of softmax, also referred to as 'softmax routing'. These coefficients are refined over iterations in order to provide agreements which are used to link capsules in the previous layer to higher order capsules in the next layer. A conventional network also consists of a mask which is used in the reconstruction of the target vector. However, in the case of gesture recognition sequence generation is not the primary objective and would only add to training time. Thus, the need for a mask is eliminated and a fully connected layer is used for dimensionality reduction and prediction.

The CapsNet architecture used in this work for the recognition of IMU signals consists of a 12×12 convolutional filter for both the capsule layers with a stride of 1. A total of 256 filters have been used for convolution. The Primary Caps layer has 20 channels with 5 dimensional capsules indicating that each capsule consists of 5 convolutional filters of size 12×12 with a stride of 2. Here, the stride value is increased to 2 keeping in mind the increasing number of convolution coefficients being delivered from the previous layer. Each primary capsule receives the input of all 256×144 1st convolutional layer (Conv1) units. Both the layers are activated with rectified linear unit (ReLU) activation. The ReLU activation has been selected empirically. The Digit Caps layer receives 20×1 capsule outputs from the Primary Caps layer as 5 dimensional vectors. Each capsule in the Digit Caps grid shares its weight with other capsules. A total of 10 dimensions are accumulated per class in the Digit Caps layer. In the case of routing, all the logits are initialized to zero, thus indicating equal probability for the capsule output to be routed to the next unit. For prediction purpose a fully connected Sigmoid layer consisting of 9324 units is inserted. One-dimensional scalar outputs from the Digit Caps layer are received by the fully connected units which lead to the prediction of the correct class. Optimization of the model is conducted by making use of the Adam optimizer with Amsgrad gradient optimization [21]. For a fixed learning rate of 0.001, Adam provides faster and efficient optimization in comparison to Sigmoid Gradient Descent (SGD) and RMSProp optimizers.

The CNN architecture depicted in Fig 2(b) is used for comparing the performance of the proposed CapsNet architecture. As shown in Fig 2(b), a filter size of 5×5 has been used because it provides the best precision or the considered dataset. Successive convolution layers consisting of 128 and 256 filters have been used which extract 2000×128 and 128×256 feature values respectively. These values are then accommodated in a subsample pool with a pool size of 3. Reducing the pool size helps the model preserve information in a reduced dimensional space such that it can be readily extracted and delivered to the next layer. Hence, the subsample pool size is selected as 3. Similarly, the 128 and 256 convolutional filters in the convolutional layers are followed by a subsample pool of pool size 3. A flattening layer has been added in order to reduce the dimensionality of the incoming information values and make them compatible with the fully connected perceptrons [22]. A total of 2000 samples of observations are fed to the fully connected layers consisting of 128 and 12 hidden units. The first layer has an excitation of ReLU while the output layer has a sigmoid activation in order to quantize the predictions in accordance with the class values. Optimization and loss parameters of the CNN model are kept same as that of the CapsNet architecture for a uniform basis of comparison.

The performance of the proposed CapsNet algorithm is assessed in terms of quantitative measures such as classification accuracies, variation of loss function and the number of false predictions and the results are compared to the conventional CNN classifier. Moreover, the concept of game theory is utilized to build a non-cooperative pick game to compare the performance of the proposed CapsNet with the conventional CNN classifier, as explained in the following section.

3. Non-cooperative CapsNet games

Making use of validation techniques has always been good practice for comparing different types of models. Custom methods suitable to the environment and models are preferred in comparison to the conventional methods [21]. Optimized models may be compared using competitive or non-competitive games [23]. Here, a non-cooperative pick game is constructed for comparing and validating the performance of different CapsNet architectures and the conventional CNN. The proposed non-cooperative pick game is designed using the following parameters:

Players: CapsNet and CNN architectures are selected as the primary players for the game. However, altering the structural parameters of the CapsNet leads to the introduction of new players in the competition.

Strategies: In a standard non-cooperative game, each player can play more than one strategy and optimize its selection of the best strategy for performance. For the considered models, these strategies can be distinguished on the basis of hyperparameters. However, for the purpose of uniform evaluation and consistent optimization in the given set of parameters, each player is constrained to play only one single pre-defined strategy on the basis of its optimizer.

Actions: Actions consist of the set of rules and steps that govern the procedure of the game. Players engaging in the non-cooperative game interact with each other by means of these steps. For the proposed non-cooperative pick game, actions of players are defined on the basis of the following three theorems-

Theorem-1. *The competing model learns and optimizes upon set $(D_i)^{train}$ by yielding predictions c_i corresponding to each sample in $(D_i)^{train}$ where $c_i \in R$ and $i = \{1,2,....,n\}$.*

Here, $(D_i)^{train}$ is the set of all the samples to be used for training of the model (hereby referred to as the training dataset), 'R' is the set of classes from which model will pick the most appropriate value and 'n' is the number of samples in the training dataset. Optimization of the model is carried out during the training phase wherein the model picks most appropriate class corresponding to each sample in the training dataset. Successive iterations of this process result in the learnt weights which indicate the peak performance of the model.

Theorem-2. *The competing model predicts upon set $(D_i)^{test}$ by yielding predictions c_j corresponding to each sample in $(D_i)^{test}$ where $c_j \in R$ and $j = \{1,2,....,m\}$.*

Here, $(D_i)^{test}$ is the set of all the samples to be used for testing of the model (hereby referred to as the testing dataset) and 'm' is the number of samples in the testing dataset. The model makes predictions on the testing samples by picking a class from the set of classes once its performance has been optimized.

Theorem-3. *The model is said to have won the competition if $c_j = c_{true}$ and for its competitor $c_j \neq c_{true}$ where c_j, $c_{true} \in R$. In any other case the competition is declared to be a draw.*

The above theorem clearly indicates the rules for deciding the winner of the game. If exactly one of the competing models picks the correct class, i.e.- the selected class is same as the true class then that model is said to have won the competition. In any other cases such as multiple models selecting the correct class or none of the models selecting the correct class the game concludes in a draw between all the models. If there are 'x' players competing in a game then the condition 'P' for a player to win the game is mathematically expressed by Eq. (5).

$$P(1) : \left(c_j\right)_1 = c_{true}; P(2) : \left(c_j\right)_2 = c_{true}; \ldots\ldots\ldots P(x) : \left(c_j\right)_x = c_{true} \tag{5}$$

These can be grouped together in a set denoted by W represented by Eq. (6).

$$W = \{P(1), \ P(2), \ldots\ldots P(x)\} \tag{6}$$

Now, for a player 'z' to win the game, as depicted in Eq. (7).

$$\exists! z : W(z); \ z \in \{1, 2, \ldots x\} \tag{7}$$

Thus, there is exactly one player 'z' in the set of possible winning criteria 'W' which satisfies the condition. However, the conditions for the game to end up as a draw are expressed by Eqs. (8)–(10).

$$\exists z : W(z); \ z \in \{1, 2, \ldots x\} \tag{8}$$

$$\vee z : W(z); \ z \in \{1, 2, \ldots x\} \tag{9}$$

$$\vee! z : W(z); \ z \in \{1, 2, \ldots x\} \tag{10}$$

These are combined together and represented by Eq. (11).

$$\exists z \,|\vee z| \vee! z : W(z); z \in \{1, 2, \ldots x\} \tag{11}$$

The game comes out to be a draw if all or none or more than one player in the competition pick the correct class. Thus, winning of the constructed non-cooperative pick game is an exclusive task which depends not only on the behaviour of the model's performance but also on the probability that the competing models fail.

Algorithm 1 highlights the procedure for the constructed novel non-cooperative pick game. It makes use of the training and validation phases of both the networks which act as players to the constructed game. A total 'n' number of predictions are made by iterating over all the samples in the training dataset by making use of the 'predict' (pred) procedure. The procedure is a simple function consisting of prediction operations of the networks. Each model selects the most appropriate class as per its strategy on the basis of learnt weights. $(c_i)_{CNN}$ denotes the ith prediction offered by CNN and $(c_i)_{CapsNet}$ denotes the ith prediction offered by CapsNet. During the validation phase, the operation of the game begins by making a total of 'm' predictions on the testing dataset. Over each iteration these predictions are compared with the $(c_j)_{true}$ which indicates the true value of the class corresponding to i^{th} sample. In case the picked class satisfies the winning criteria then the corresponding model is declared as the winner. In any other case the game results in a draw. Since a total of 'm' iterations lead to 'm' assessments of the winning criteria, a total 'm' number of games are utilized to compare the relative performances of CNN and CapsNet with respect to each other.

Algorithm 1 Proceudre of non-cooperative pick game.

---

for $i = 1{:}n$
    $(c_i)_{CNN} \leftarrow \text{pred}((D_i)^{train})$
    $(c_i)_{CapsNet} \leftarrow \text{pred}((D_i)^{train})$
for $j = 1{:}m$
    $(c_j)_{CNN} \leftarrow \text{pred}((D_i)^{test})$
    $(c_j)_{CapsNet} \leftarrow \text{pred}((D_i)^{test})$
    if $((c_j)_{CapsNet} == (c_j)_{true}$ & $(c_j)_{CNN} \neq (c_j)_{true})$
      Winner $\leftarrow$ CapsNet
    else if $((c_j)_{CNN} == (c_j)_{true}$ & $(c_j)_{CapsNet} \neq (c_j)_{true})$
      Winner $\leftarrow$ CNN

---

4. Results and discussion

4.1. CapsNet recognition

The proposed one-dimensional CapsNet model is applied on the IMU signals to perform classification of the signed sentences. The performance of the proposed model is compared to the conventional state-of-the-art CNN having similar hyper-parameters. Both the models make use of the same loss function, categorical cross-entropy. However, final layer activation
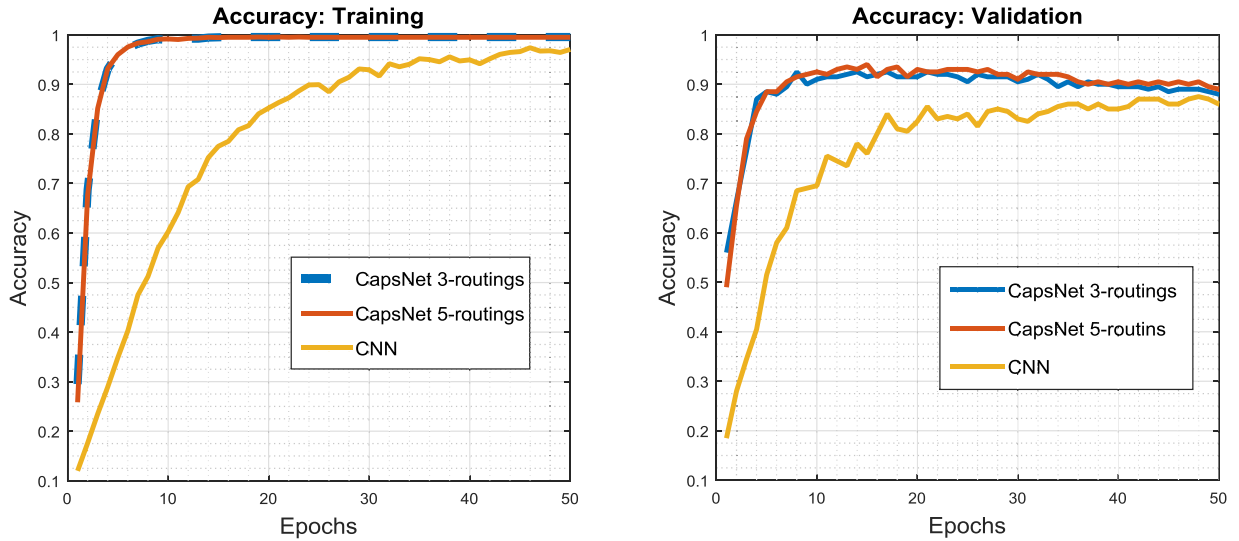
Fig 3. Average accuracy value variation over 50 iterations for (a) Training and (b) Validation.
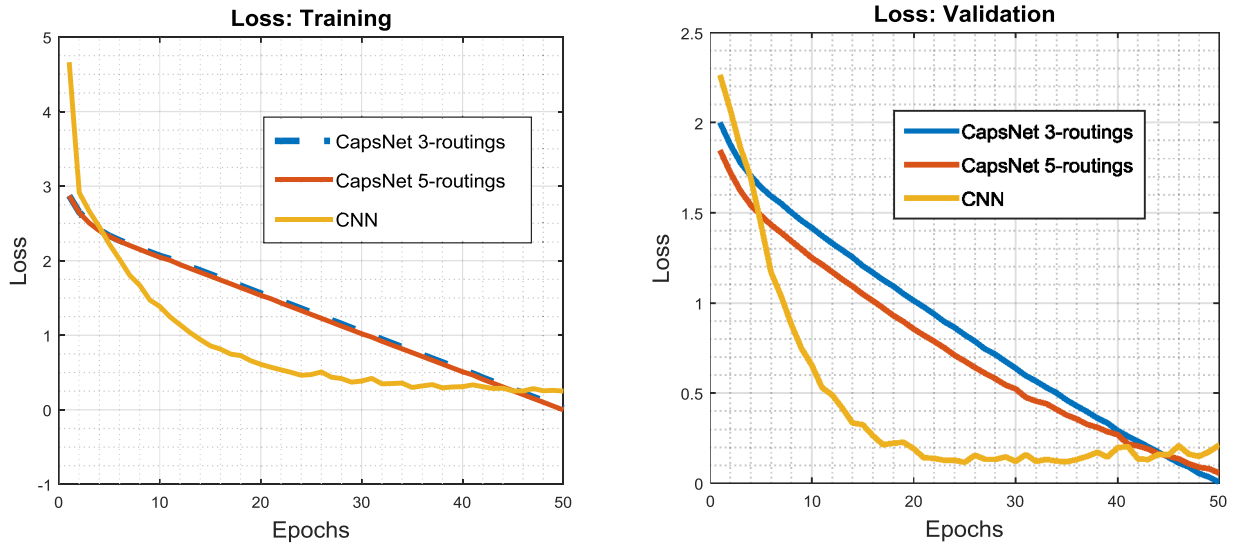

Fig 4. Optimization loss variation (categorical crossentropy) over 50 iterations for (a) Training and (b) Validation.

for CNN is empirically determined to be softmax, since it yields better learn weights over successive iterations. Also, different number of routings are used in the CapsNet architecture to assess the variation of performance of the classifier when the number of dynamic connections is increased or decreased. Fig 3a and b depict the training and validation accuracies of the models over 50 iterations respectively. Learning of CapsNet exceeds that of CNN as a result of faster response in improvement over the learnt weights. As observed in Fig. 3, while the training accuracies for 3 or 5 routings overlap, during validation, the model architecture with 3 routings performs better in comparison to 5 connections and CNN. Peak accuracy values for the CapsNet architecture with 3 routings are observed to be 99.72% during training and 94.00% during validation. Fig. 4a and b highlight the same variation by means of optimization loss during training and validation, respectively. CapsNet with 3 routings depicts better performance with approximately 0.01 units of loss during training and validation. Another informative measure to assess the predictive behaviour of these models is the number of false predictions. A low value of optimization loss need not necessarily indicate accurate predictions over all the batches. Since loss is calculated as the average over all the batches, misclassification of values between batches tends to be neglected in this process. Fig. 5a and b indicate the number of false predictions over iterations during training and validation, respectively. Fewer false predictions in the case of 3 routings architecture are observed (8 during training and 12 during validation) as compared to false predictions for 5 connections and CNN.
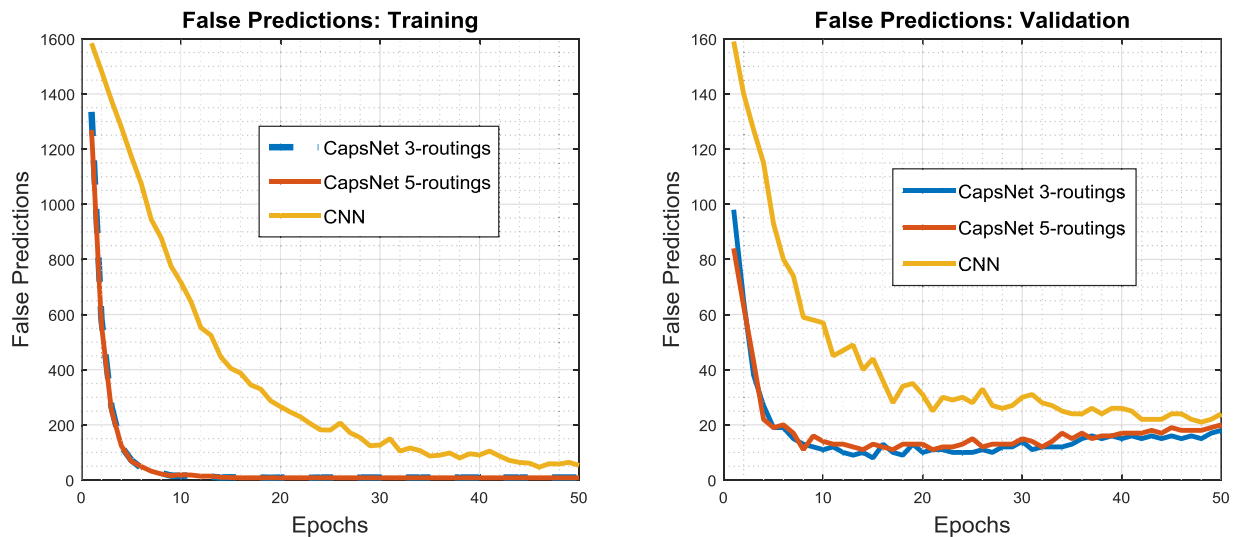
Fig 5. False predictions variation over 50 iterations for (a) Training and (b) Validation.
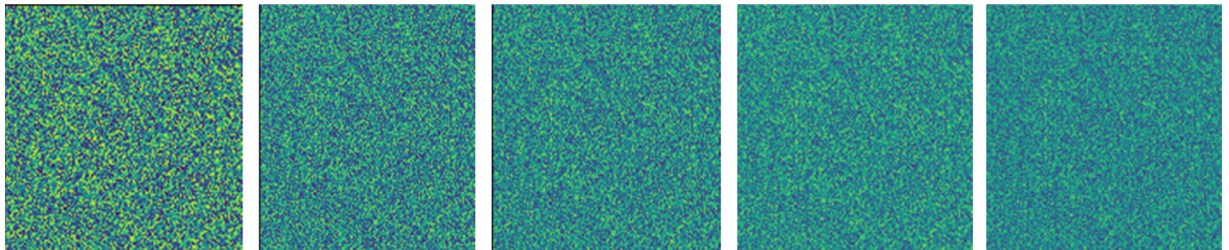

Fig 6. Learnt activations at the final layer of CNN over (a) 1 iteration, (b) 10 iterations, (c) 20 iterations, (d) 30 iterations and (e) 40 iterations.
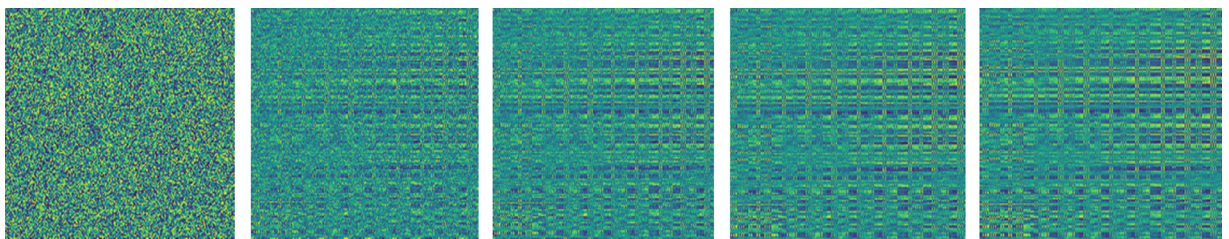

Fig 7. Learnt activations at the final layer of CapsNet over (a) 1 iteration, (b) 10 iterations, (c) 20 iterations, (d) 30 iterations and (e) 40 iterations.

Analysing learnt activations for the model plays a significant role in the evaluation of learning as these weights tend to change over each iteration [24]. Figs. 6 and 7 indicate the improvement in weights after every 10th iteration for CNN and CapsNet architecture. Activation of the last layer tends to improve by virtue of the learnt weights. Higher intensity of the colour gradient depicts highly activated units whereas lower gradients indicate lower values. As seen from Fig. 7a to e, the multiplicative values modify over passing iterations producing a regular pattern in Fig. 7e, which depicts the weights in the spatial domain and a definite structure in the background for CapsNet architecture. This definite structure in Fig. 7(e) indicates the activated units in the layer. However, no such regular pattern is observed to emerge in the case of CNN (Fig. 6a to e). Thus, sharp spatial activations with a higher intensity of the colour gradient in Fig. 7 indicate better optimized weights of CapsNet in comparison to CNN. Accurate predictions are a cause of modified activations at the end of the training phase. Comparing weights of CapsNet to CNN, the CapsNet architecture is expected to yield accurate predictions and enhanced learning as a result of the highly activated units arising from dynamic routing between nested units. This is not observed in a conventional CNN due to the absence of nested units and a routing technique.

4.2. Non-cooperative games

The proposed two-player game designed as explained in Section III, allows the participating models to choose the most appropriate class from the set of all classes based on the information learnt during training. Both the models optimize their performance by learning the values of the weights in order to perform correct predictions. Interactions between players are limited to the extent that they compete with each other during each validation phase. A total of 3 games are played, each between two sets of players at a given time. A simple three-player game would also be equivalent to these games. However, such a scenario would not provide details on the relative performance and would require the use of three-dimensional space for visualization. Each game of the three games consist of the two players. These games based on their players may be mathematically summarised as,

$$Game-1-P(CNN) : \left(c_j\right)_{CNN} = c_{true}; P(CapsNet-3) : \left(c_j\right)_{CapsNet-3} = c_{true} \qquad (12)$$

$$Game-2-P(CNN) : \left(c_j\right)_{CNN} = c_{true}; P(CapsNet-5) : \left(c_j\right)_{CapsNet-5} = c_{true} \qquad (13)$$

$$Game-3-P(CapsNet-3) : \left(c_j\right)_{CapsNet-3} = c_{true}; P(CapsNet-5) : \left(c_j\right)_{CapsNet-5} = c_{true} \qquad (14)$$

where $(c_j)_{CNN}$ denotes the class picked by CNN corresponding to the jth sample in the validation set, $(c_j)_{CapsNet-3}$ denotes the class picked by CapsNet with 3-routings corresponding to the jth sample in the validation set and $(c_j)_{CapsNet-5}$ denotes the class picked by CapsNet with 5-routings corresponding to the jth sample in the validation set. Collective representation of these games in set notation is expressed by Eq. (15).

$$W_1 = \{P(CNN), P(CapsNet-3)\}; W_2 = \{P(CNN), P(CapsNet-5)\}; W_3 = \{P(CapsNet-3), P(CapsNet-5)\} \qquad (15)$$

where W_1, W_2 and W_3 indicate games 1, 2 and 3, respectively. Once both the models in a game have reached their best responses, their relative performances may be evaluated by means of Nash Equilibrium which is defined as followed [25],

Definition-1 (Nash Equilibrium). Nash Equilibrium is defined as a stable state involving the interaction of different players, in which no player can gain a unilateral change of strategy if the strategies of other players remain unchanged.

Definition-1 indicates that once the strategy of a player is fixed, it cannot be altered with respect to the strategy of the other player. The idea of Nash Equilibrium corresponds to the point where both the players in the constructed games have adopted and optimized the single selected strategy. It denotes that point in the game where both the players are at their best with respect to each other's performance. Nash Equilibrium is an essential indicator in determining the extent of optimization between the strategies. Once the strategies have been optimized the models are said to have their best responses during the interaction. In this work, the players are CNN and CapsNet with 3- or 5-routings. The strategy of the players is the optimization of the weights that happens during the training phase. The pick games are played during the testing phase. Relative performance between the two competing models in games is thus assessed by visualizing their best responses. Fig 8 indicates the best responses for both the models in games with respect to each other. Best response here refers to the probability that the class selected by the model is same as the true class. The point of intersection on the best response plot denotes the Nash Equilibrium for the corresponding game. Both the players have adopted and optimized their single selected strategy in order to reach the point of Nash Equilibrium. Of the two models CNN and CapsNet, the CapsNet architecture depicts a higher probability of winning in comparison to CNN. CapsNet with 3-routings achieves the best performance in all of its games with a response of 0.94 when compared to CapsNet with 5-routings and CNN that achieve their best performance with responses of 0.93 and 0.88, respectively.

Now, the computational complexity and time required by the considered models is compared. Each iteration of the training phase takes 180 s for the conventional CNN, 480 s $\pm$ 1 μs for CapsNet with 3-routings and 520 s for CapsNet with 5-routings. Although, the time required by CapsNet for optimization is higher than CNN, however during validation, CapsNet predicts values faster and more accurately than CNN as a result of increased weight optimization. Since optimization is absent in the validation phase, it only takes 9 ms (45 μs per sample) and 10 ms (50 μs per sample) for CapsNet with 3- and 5- routings to make the predictions, respectively. Thus, each game, in the case of 3-routings, takes 45 μs to complete as result of the models being pre-trained.

Table 1 summarizes the performance of CapsNet models in comparison to CNN. CapsNet with 3-routings achieves the highest performance for recognition of the signed sentences from the IMU signals with an optimized loss of 0.01, false predictions as 12, accuracy of 94% and Nash Equilibrium for non-cooperative games at 0.94. CapsNet with 5-routings achieves a 4.50% improvement in recognition and non-cooperative games when compared to CNN. Improved results of the CapsNet architecture in comparison to the conventional CNN highlight the appropriateness of dynamic routing with nested layers in the capsule theory. In the future, these capsules can be further modified as stages in a hierarchical network such as master-slave models and parallel architectures. This would help solve problems related to under-fitting.
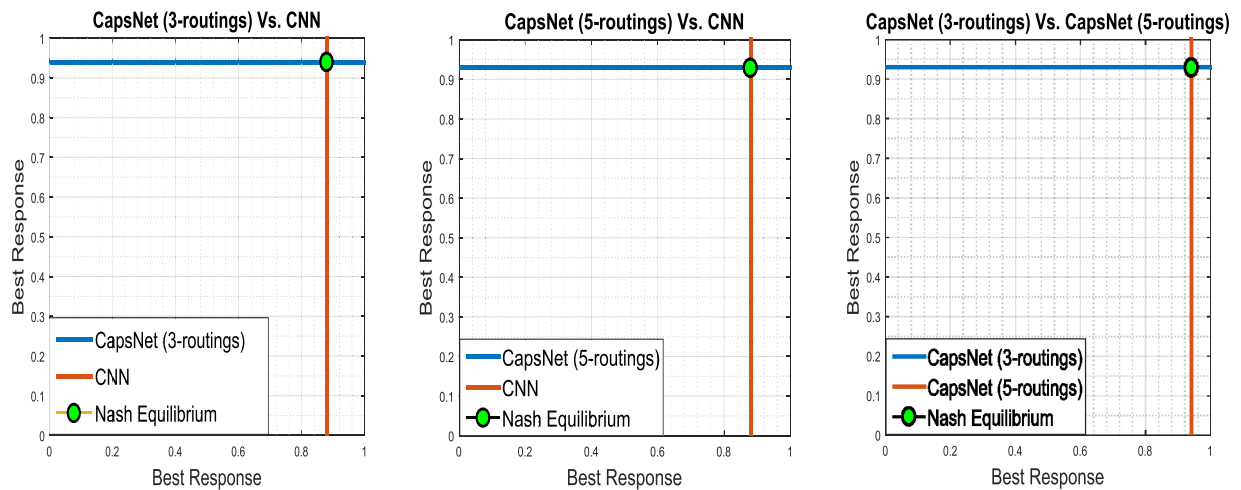
Fig 8. Nash equilibrium obtained between best responses for (a) CNN Vs. CapsNet (3 routings), (b) CNN Vs. CapsNet (5 routings) and (c) CapsNet (5 routings) Vs. CapsNet (3 routings).

Table 1
Performance comparison of CapsNet architecture with CNN.

Architecture	Optimization loss		Misclassified outputs		Classification accuracy		Nash equilibrium	
	Training	Validation	Training	Validation	Training	Validation	Training	Validation
CNN	0.07	0.11	139	24	93.00%	87.99%	0.93	0.88
CapsNet 3 routings	**0.01**	**0.01**	**8**	**12**	**99.72%**	**94.00%**	**0.99**	**0.94**
CapsNet 5 routings	0.01	0.02	8	15	99.56%	92.50%	0.99	0.93

5. Conclusion

In this work, recognition of sentences signed according to the Indian sign language is performed using signals recorded from a custom-designed wearable IMU device. The wearable IMU device consists of a tri-axial accelerometer and a tri-axial gyroscope. Additionally, the orientation of the hand is also estimated. The sentence recognition is carried out using a novel one-dimensional CapsNet architecture. The proposed model makes use of Capsule theory consisting of nested convolutional layers linked to each other by dynamic routing. The performance of the proposed CapsNet architecture is compared with its predecessor model, a conventional CNN.

Improved classification accuracy of 94% is observed for CapsNet with 3-routings in comparison to CapsNet with 5-routings and CNN, which yield 92.5% and 87.99% classification accuracy, respectively. In terms of recognition loss, the 3-routing model of CapsNet has a minimized loss of 0.01 in comparison to 0.02 and 0.11 for the 5 routing CapsNet model and CNN, respectively. Thus, CapsNet with 3-routings is better optimized in contrast to the other two models. Learning of the architecture is validated by observing spatial activations depicting excited units at the final layer. The CapsNet model presents a uniform pattern of activated convolutional perceptrons in its spatial weight activations. This is found to be absent in the case of CNN indicating less excitation. Furthermore, a non-cooperative pick game is constructed for assessing the relative performance of the models. The game is constrained to single strategy adoption which is optimized by means of optimization function of the model. Both the models optimize their performances with respect to each other in order to attain Nash Equilibrium. The model with higher recognition accuracy at its Nash Equilibrium is declared to be more accurate in the competition. CapsNet architecture presents a better value of the best response at Nash Equilibrium asserting the suitability of the proposed approach.

Declaration of Competing Interest

None.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compeleceng.2019.08.006.

References

[1] Sikeridis, D. and T.A. Antonakopoulos, "An IMU-based wearable system for automatic pointing during presentations", Image Processing & Communications, vol. 21, no. 2, pp. 7–18.
[2] Zhu K, Shi L. Motion control in VR — real-time upper limb tracking via IMU and flex sensor. IEEE; 2016. p. 1–5.
[3] Chophuk P, Pattanaworapan K, Chamnongthai K. Fist American sign language recognition using leap motion sensor. In: 2018 International workshop on advanced image technology (IWAIT). IEEE; 2018. p. 1–6.
[4] Mahesh M, Jayaprakash A, Geetha M. Sign language translator for mobile platforms. In: 2017 International conference on advances in computing, communications and informatics (ICACCI). IEEE; 2017. p. 1–6.
[5] Kaneishi D, Matthew RP, Tomizuka M. A SEMG classification framework with less training data. In: 2018 40th Annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE; 2018. p. 1–6.
[6] Zheng W-L, Lu B-L. Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. IEEE Trans Auton Ment Dev 2015;7(September(3)):162–75.
[7] Sun Y, Lo B. An artificial neural network framework for gait-based biometrics. IEEE J Biomed Health Inform 2019;23(May(3)):987–98.
[8] Suri K, Gupta R. Classification of hand gestures from wearable IMUs using deep neural network. In: 2nd International conference on inventive communication and computational technologies (ICICCT). IEEE; April 2018. p. 45–50.
[9] Wöllmer M, Kaiser M, Eyben F, Schuller B, Rigoll G. LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework. Image Vis Cmput 2013;31(February(2)):13–163.
[10] Suri K, Gupta R. Transfer learning for sEMG-based hand gesture classification using deep learning in a master-slave architecture. In: 3rd International conference on contemporary computing and informatics (IC3I). IEEE; October 2018. p. 1–6.
[11] Wu M, Chen L. Image recognition based on deep learning. In: 2015 Chinese automation conference. IEEE; 2016. p. 1–6.
[12] Li D, Zhang J, Zhang Q, Wei X. Classification of ECG signals based on 1D convolution neural network. In: 2017 IEEE 19th International conference on e-health networking, applications and services (Healthcom). IEEE; 2017. p. 1–6.
[13] Engstrom L, Tran B, Tsipras D, Schmidt L, Madry A. In: A rotation and a translation suffice: fooling CNNs with simple transformations; 2017. p. 1–6.
[14] Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. In: 31st Conference on advances in neural information processing systems (NIPS 2017); 2017. p. 1–11.
[15] Xiang C, Zhang L, Tang Y, Zou W, Xu C. MS-CapsNet: a novel multi-scale capsule network. IEEE Signal Process Lett 2018;25(Decemebr(12)):1–14.
[16] Oliehoek FA, Savani R, Gallego-Posada J, van derPol E, de Jong ED, Groß R. GANGs: generative adversarial network games. Cornell Lib April 2017:1–9.
[17] Schuurmans D, Zinkevich M. Deep learning games. In: 30th Conference on neural information processing systems (NIPS 2016); 2016. p. 1–9.
[18] Wu Q, Wu R, Han F, Zhang R. A three-stage accelerometer self-calibration technique for space-stable inertial navigation systems. Sensors 2018:1–16.
[19] Chen Y, Aktakka EE, Woo J-K, Ajafi KN, Oldham K. Error contributions during MEMs gyroscope calibration by chip-scale micro-stage with capacitive motion sensor. In: 2018 IEEE/ASME International conference on advanced intelligent mechatronics (AIM). IEEE; 2018. p. 1–6.
[20] Faculty of Disability Management and Special Education (FDMSE). Indian sign language. Ramakrishna Mission Vivekananda University; 2018.
[21] Alzubi JA. Optimal classifier ensemble design based on cooperative game theory. Res J Appl Sci Eng Technol 2015;11(12):1336–43.
[22] Kingma DP, Ba JL. Adam: a method for stochastic optimization. In: Internaional conference on learning representations (ICLR); 2015. p. 1–15.
[23] Andersen P-A, Goodwin M, Granmo O-C. Deep RTS: aGame environment for deep reifnorcement learning in real-time strategy games. In: International conference on computational intelligence and games (CGIS). IEEE; 2018. p. 1–8.
[24] Sünetci S, Ateş HF. Semantic image segmentation with deep features. In: 2018 26th Signal processing and communications applications conference (SIU). IEEE; 2018. p. 1–6.
[25] Chakravorty S, Ghosh D. Nash equilibrium strategy for non-cooperative games with interval type-2 fuzzy payoffs. In: 2018 IEEE International conference on fuzzy systems (FUZZ-IEEE). IEEE; 2018. p. 1–6.

Mr. Karush Suri is pursuing his masters in engineering in electrical and computer engineering from the University of Toronto, Ontario. He completed his Bachelor of Technology in Electronics and Communication Engineering from Amity University Uttar Pradesh, India. His research is concerned with the application of deep learning and machine intelligence for task specific learning such as hand motion analysis and gesture recognition.

Rinki Gupta received her Ph.D. in signal processing from Indian Institute of Technology Delhi, India. Currently, she is a faculty at Amity School of Engineering and Technology, Amity University Uttar Pradesh, India. She is heading a funded research project and has worked on several others. Her research interests include human motion analysis, multi-sensor data fusion and audio signal processing.