A Project Report

On

APPLICATION OF DEEP LEARNING AND GAME THEORY FOR SIGN LANGUAGE RECOGNITION USING WEARABLE SENSORS

Submitted to

Amity University Uttar Pradesh

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

Electronics and Communication Engineering

By

KARUSH SURI

under the guidance of

Dr. Rinki Gupta

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY
AMITY UNIVERSITY UTTAR PRADESH

April-May 2019

DECLARATION

I, Karush Suri, student of B.Tech (*Electronics and Communication Engineering*) hereby declare that the project titled "Application of Deep Learning and Game Theory for Sign Language Recognition using Wearable Sensors" which is submitted by me to the Department of Electronics and Communication Engineering, Amity School of Engineering and Technology, Amity University Uttar Pradesh, in partial fulfillment of requirement for the award of the degree of Bachelor of Technology (*Electronics and Communication Engineering*), has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

Noida

01/05/2019

Karush Suri

CERTIFICATE

On the basis of declaration submitted by Karush Suri, student of B. Tech (*Electronics and Communication Engineering)*, I hereby certify that the project titled "Application of Deep Learning and Game Theory for Sign Language Recognition using Wearable Sensors" which is submitted to the Department of Electronics and Communication Engineering, Amity School of Engineering and Technology, Amity University Uttar Pradesh, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (*Electronics and Communication Engineering*), is an original contribution with existing knowledge and faithful record of work carried out by him under my guidance and supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Noida

01/05/2019

Dr. Rinki Gupta
Assistant Professor
Dept. of Electronics and Communication Engineering
Amity School of Engineering and Technology
Amity University Uttar Pradesh

DECLARATION FORM

I, Karush Suri, student of B.Tech (*Electronics and Communication Engineering*), A2305115034, 2015-2019, Department of Electronics and Communication Engineering, Amity School of Engineering and Technology, Amity University, Uttar Pradesh, Noida hereby declare that I have gone through project guidelines including policy on health and safety, policy on plagiarism etc.

01/05/2019.

Noida.

ACKNOWLEDGMENT

The author would like to thank the volunteers who helped in recording the data. The author also recognizes the funding support provided by the Science & Engineering Research Board, a statutory body of the Department of Science & Technology (DST), Government of India (ECR/2016/000637).

ABSTRACT

Sign Language is used by the deaf community all over world. Internationally, various research groups are working towards the development of an electronic sign language translator to enhance the accessibility of a signer. By employing intelligent models and wearable devices such as inertial measurement units (IMUs), continuous signs leading to the formation of a complete sentence can be recognized effectively. 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 IMU records tri-axial acceleration and turn rate, and orientation of the sensor is evaluated using a complementary filter. All the signals are used in the proposed deep learning network for learning and recognition of the signed sentences. The performance of the proposed CapsNet architecture is assessed by altering dynamic routing between capsule layers. Performance of the model is compared to that of foundational convolutional neural networks (CNNs) in terms of accuracy, loss, false predictions and learnt activations. The proposed CapsNet yields improved accuracy values of 94% (for 3 routing) and 92.50% (for 5 routings) in comparison to CNNs which yield 87.99%. Improved learning of the architecture is also validated by spatial activations depicting excited units at the predictive layer.

The work also presents a novel one-dimensional Convolutional Neural Network (CNN) array architecture. The signals recorded using the IMU device are segregated on the basis of their context, such as whether they correspond to signing for a general sentence or an interrogative sentence. The array comprises of two individual CNNs, one classifying the general sentences and the other classifying the interrogative sentence. Performances of individual CNNs in the array architecture are compared to that of a conventional CNN classifying the unsegregated dataset. Peak classification accuracies of 94.20% for general sentences and 95.00% for interrogative sentences achieved with the proposed CNN array in comparison to 93.50% for conventional CNN assert the suitability of the proposed approach.

For the purpose of evaluating relative performance and competitive nature of models, a novel non-cooperative pick game is constructed. The game presents a pick-and-predict competition between CapsNet and CNN constrained to a single strategy adoption. Both models compete with each other in order to reach their best responses. Higher value of Nash equilibrium for CapsNet as compared to CNN indicates the suitability of the proposed approach.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1: INTRODUCTION

Human motion analysis has been studied for various applications using different sensing technologies [1-9]. For instance, human gait may be analysed to localize the phases in a gait that may be used for health monitoring or navigation purpose [1,2]. 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 [3, 4]. 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 [3] or control a virtual-reality head-mounted display based on real-time tracking of upper limb [4].

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 [5], videos [6] as well as wearable sensors such as surface electromyogram (sEMG) and IMU [7,8]. 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 [5,6]. 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 measure 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 [8]. Although they are non-invasive and provide complementary capability as compared to IMUs for hand gesture recognition [7], 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 [9-10]. 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 [11]. 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 [12]. 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 [13]. 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 provides higher training times and complex algorithms in collaboration to augment the training process [14]. 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 [15]. Based on two-dimensional convolution, CNNs are excessively used in image processing applications for digit and object recognition [15, 16]. CNN have also been applied on one-dimensional data and shown to yield good classification accuracy of 97.5% [17]. 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 [18]. 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 [19]. 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 [20]. 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 [21]. 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 [22].

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 Chapter 3. Performance of the architecture is compared to the foundational CNN having similar trainable hyper-

parameters. Furthermore, validation of accurate predictions is obtained by constructing 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 Chapter 4. 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. Architecture for the 2x1 CNN array is explained in Chapter 5. The results obtained with actual data are presented in Chapter 6 and the Conclusion section contains the concluding remarks.

CHAPTER-2: REAL-TIME PROCESSING

Gesture recognition has gained significant importance in the past decade. With the advent of advanced biomedical techniques and unsupervised learning algorithms, progress of the analysis of hand motion signals is evident. Hand motion may be studied in the form of position, orientation and the intensity of the muscle. Of these, the orientation aspect contains significant processing and gives a concise idea about the rotation of the limb. It is essential to accurately evaluate the variations in rotational aspect of the forearm. Moreover, a real time evaluation of the gesture being performed plays a significant role in making the process more dynamic and suitable to the needs of the subject. Combining these elements of limb movements may help in language translation, arm amputee treatment and gesture controlled devices. Such a human-machine interface can successfully deliver prosthetic tools, smart security systems, optimized language translators and health tracking bands.

The project proposes a real time system capable of analyzing motion of various hand gestures. Study of these gestures may be carried out in the form of signals. The system makes use of economical sensors such as Intrinsic Measurement Units (IMUs) capable of evaluating the motion of the limb through its orientation. An accurate estimation of hand movement can be obtained by acquiring signals from these low cost sensors such as accelerometer and gyroscope. Acquisition of these signals can then be carried out prior to processing. Processing of the acquired signals consists of orientation estimation determining the angular movements of the limb in real time. Validation of results can be carried out by comparing previously recorded gestures to that performed by the subjects in real time.

Such a system makes use of only the rotational aspects for determining the orientation of the limb and is further optimized in real time. Advances in the design may be carried out by assessing intensity of the muscles in the form of

surface Electromyography (sEMG) and recognizing gestures performed by the subject using deep algorithms capable of learning the most intricate aspects of data. The project depicts the process of hand gestures using real-time IMU setup.

Development of a real time system capable of assessing hand gestures is useful in developing assistive technology, such as sign language translation and in developing human machine interfaces for example in gaming applications. A real time acquisition algorithm would play a significant role in obtaining meaningful data from multiple sensors and processing it as per the need of the application. Following are the main aspects of the proposed real-time system-

- Acquisition of signals from triaxial accelerometer and triaxial gyroscopes for different hand motions.
- Processing of the acquired data in real time for pre-processing and Orientation and/or position estimation.
- Assess the estimated orientation to determine the type and nature of the gesture performed.
- Validating the results for various hand gestures recorded with multiple subjects.

The pre-processing step being the key step makes use of time-stamp correction followed by median filtering. Corresponding to each signal, unwanted time stamp values at the serial port are cleared in order to assert continuity of the signal. Once the signal obtained is continuous in nature and free from any spurious jumps, filtering of the samples is carried. In order to remove any peak overshoots in the form of noise, median filtering is carried out on samples replacing the overshoot with the median value of its neighbourhood samples.

CHAPTER-3: CAPSNET RECOGNITION

3.1 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 gyroscope have 3 degree-of-freedom each, giving a total of 6 channels of data. Fig 1.a 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 1.b. The sensors were calibrated for bias and scale factor. Accelerometer data was corrected by using the 12-parameter estimation method [29]. On the other hand, rotational signals were corrected by subtracting bias values and then scaling the sequence [30].

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

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. Signs were recorded from the Indian Sign Language (ISL) in a sequence, resulting in the formation of a complete sentence [31]. A total of 20 sentences were gathered with each sentence consisting of 2-4 signs spaced approximately 1s 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’. Fig 2. presents the structure and the signs constructing the 20 sentences used for sign language recognition.

(a) Assertive sentence

(b) Interrogative sentence

Fig 2. Structure and signs used in sentences

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

$$A_\alpha = atan\frac{A_x}{\sqrt{A_y^2+A_z^2}}, \qquad (1)$$

$$A_\rho = atan\frac{A_y}{\sqrt{A_x^2+A_z^2}}, \qquad (2)$$

and

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

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

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

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.

3.2 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 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 3. provides the architecture for the one-dimensional CapsNet used in the recognition of IMU signals. 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 [19]. 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.

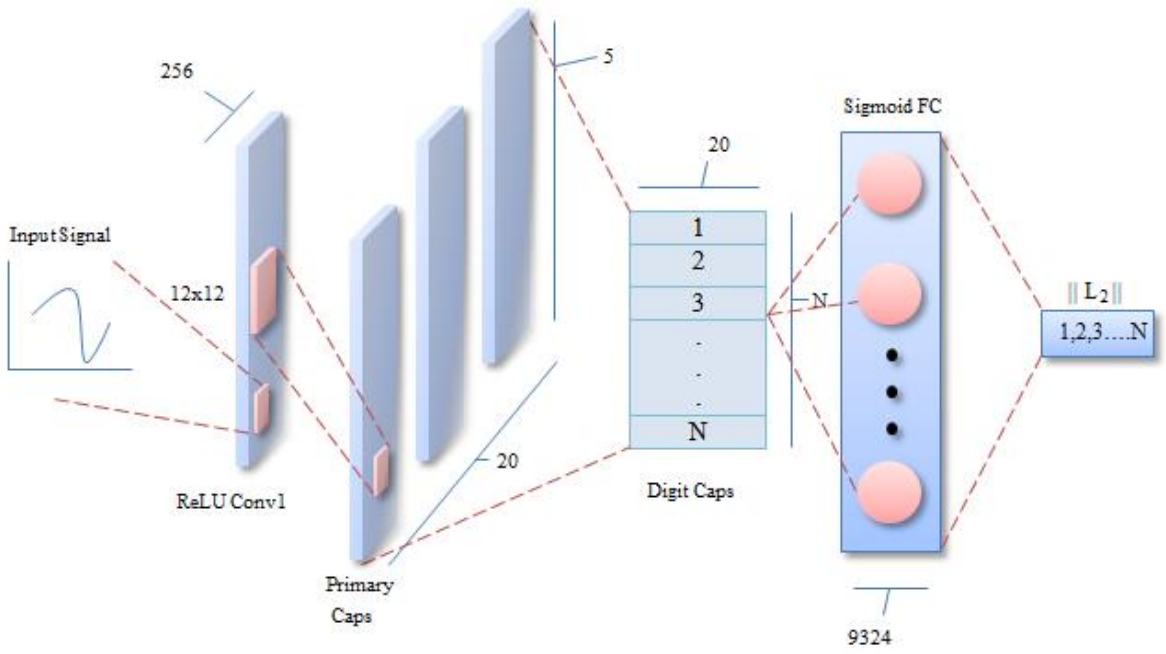


Fig 3. One-Dimensional CapsNet Architecture for Signed Sentence Recognition

The CapsNet architecture used in this work for the recognition of IMU signals consists of a 12x12 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 12x12 with a stride of 2. Each primary capsule receives the input of all 256x144 1st convolutional layer (Conv1) units. Both the layers are activated with rectified linear unit (ReLU) activation. The Digit Caps layer receives 20x1 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 [32]. Finally, 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.

CHAPTER-4: 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 [33]. Optimized models may be compared using competitive or non-competitive games [34]. Here, a non-cooperative pick game is constructed for comparing 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 hyper-parameters. 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* $c_i \epsilon R$ *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* $c_j \epsilon R$ *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* $c_j, c_{true} \epsilon R$. *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.

$$P(1): (c_j)_1 = c_{true}\ ;\ P(2): (c_j)_2 = c_{true};\ .\ .\ .\ .\ .\ .\ .\ .\ .\ P(x): (c_j)_x = c_{true} \qquad (5)$$

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

$$W = \{P(1), P(2), .\ .\ .\ .\ .\ .\ .\ .\ P(x)\} \qquad (6)$$

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

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

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 is expressed by Eq. 8, 9 and 10.

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

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

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

These are combined together and represented by Eq. 11.

$$\exists z \mid \vee z \mid \vee !z : W(z)\ ;\ z \in \{1,2,.\ .\ .\ .\ x\} \qquad (11)$$

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 optimization of the model's performance but also on the probability that the competing models fail.

Algorithm 1.

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 ← CapsNet

else if $((c_j)_{CNN} = = (c_j)_{true}$ & $(c_j)_{CaspNet} \neq (c_j)_{true})$

Winner ← CNN

Algorithm 1 highlights the procedure for the constructed novel non-cooperative pick 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. Each model selects the most appropriate class as per its strategy on the basis of learnt weights. $(c_i)_{CNN}$ denotes the i^{th} prediction offered by CNN and $(c_i)_{CapsNet}$ denotes the i^{th} prediction offered by CapsNet. During the validation phase, the operation of the game begins by making a total '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 '*n*' iterations lead to '*n*' assessments of the winning criteria, a total '*n*' number of games are utilized to compare the relative performances of CNN and CapsNet with respect to each other.

CHAPTER-5: 2x1 CNN ARRAY ARCHITECTURE

The proposed array architecture as depicted in Fig 4. is a 2x1 array consisting of 2 CNNs. Each CNN operates on a particular subset of data, i.e. either questions or sentences. These subsets have been manually created prior to the training phase and split into training and validation data. Both the CNNs have identical structure for convenience. Each network consists of 9 layers, these being convolutional layers, pooling layers, flattening layers and fully-connected perceptron layers. The convolution layer consists of 128 and 256 filters respectively, each having a size of 5x5 and activation as Regularized Tangent (Relu) The pooling subsample space has a pool size of 3. Convolutional and pooling layers together make the feature learning network. The recognition network consists of fully-connected (FC) multi-perceptron layers. The first layer in the recognition network consists of 128 hidden units whereas the number of units in the final layer depend on the number of classes. Since there are 12 general sentences in the IMU dataset, CNN-sentence consists of 12 units in the final layer. Similarly, 8 questions in the IMU dataset make up for 8 units in CNN-questions. Each of the output layers in the CNN array are activated by means of sigmoid activation. Flattening of learned entities is carried out for the purpose of dimensionality reduction followed by predictions offered by the recognition network. Optimizer used for the process of recognition is *RMSprop* initialized to a learning rate of 0.0001 and no decay. Operation of the array is governed by sequential classification of input data. IMU dataset is subdivided into subsets of sentences and questions. Each of these subsets consists of data from all the 10 subjects having 10 repetitions of each class. Thus, the sentence subset has a total of 10*10*12 = 1200 signals as samples (corresponding to 12 classes) whereas the questions subset has 10*10*8 = 800 signals as samples (corresponding to 8 classes). Each of these subsets is further split into training and validation data with 90% of the samples falling in the training set. CNN-sentences is therefore trained on 1080 signals and validated on 120 signals whereas CNN-questions is trained on 720 signals and validated on 80 signals. Since there are multiple classes for each CNN in the array, the use of categorical cross entropy as a suitable loss function is employed. Equations (12) and (13) denote the expression for categorical cross entropy on the total number of samples ‘n’ in a training dataset.

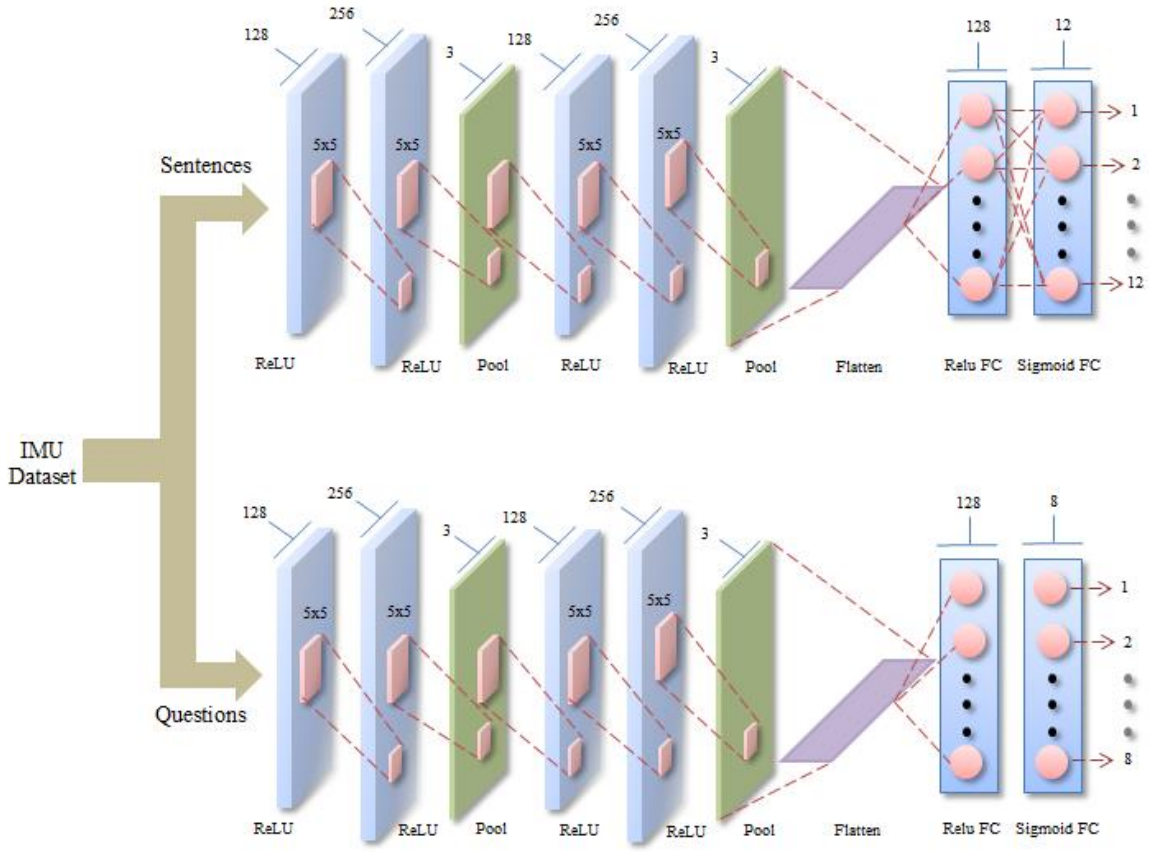


Fig 4. Proposed 2x1 CNN Array Architecture for Sign Language Recognition

$$H(p,q) = -[(p(x_1)\,log\,q(x_1)) + (p(x_2)\,log\,q(x_2)) + \ldots.(p(x_n)\,log\,q(x_n))], \quad (12)$$

$$H(p,q) = -\sum_{i=1}^{n}[(p(x_i)\,log\,q(x_i))]. \quad (13)$$

Here, x_i denotes the i^{th} sample in the subset, $p(x_i)$ the class probability corresponding to the i^{th} sample and $q(x_i)$ the likelihood corresponding to the i^{th} sample. The loss in multi-layered networks may not depend on incorrect classes. However, the gradient of the loss function does affect incorrect classes. This leads to an optimization in the learned weights over each passing iteration. Gradient of the loss function can be expressed as the derivative with respect to the i^{th} sample as given in (14), This can further be simplified to give,

$$\frac{\partial \mathrm{H(p,q)}}{\partial \mathrm{x_i}} = -\frac{\partial [\mathrm{p(x_i)} \log (\mathrm{q(x_i)})]}{\partial \mathrm{x_i}} . \qquad (14)$$

$$\frac{\partial H(p,q)}{\partial x_i} = -p(x_i)\frac{1}{q(x_i)}\frac{\partial q(x_i)}{\partial x_i}. \qquad (15)$$

Equation (15) indicates the gradient updates taking place for each sample in the training set. True classes representing a finite value of $p(x_i)$ are taken into account. Values corresponding to false classes represent a zero value of $p(x_i)$. Categorical representation of predictions is thus, organized as 1's and 0's indicating true and false classes, respectively.

CHAPTER-6: GEAR2.1

Presentation and visualization of the program from a user's standpoint is essential. Obtained predictions can be presented to the user using a graphical interface which would clearly indicate the performance and accuracy of learning models in real-time. Such an interface would also help the user in convenient usage of the program instead of forcing him to go through the complete program.

Gesture Evaluator and Recognizer (GEAR2.1) is the designed Graphical User Interface (GUI) which allows the user to interact with the constructed algorithms. The algorithms can then offer their predictions by obtaining and assessing the gesture data in real-time. GEAR2.1 is made easy for daily usage and is a simple interface with minimum overheads. All the associated files with the GUI are stored in one single directory which the program makes use of in order to predict gestures in real-time. Fig 5.a provides the complete view of the interface. The interactive buttons allow the user to carry out specific actions such as read the description, import all the AI files and start the competition between the two AIs, CNN and CapsNet in real-time once the IMU system is connected and online.

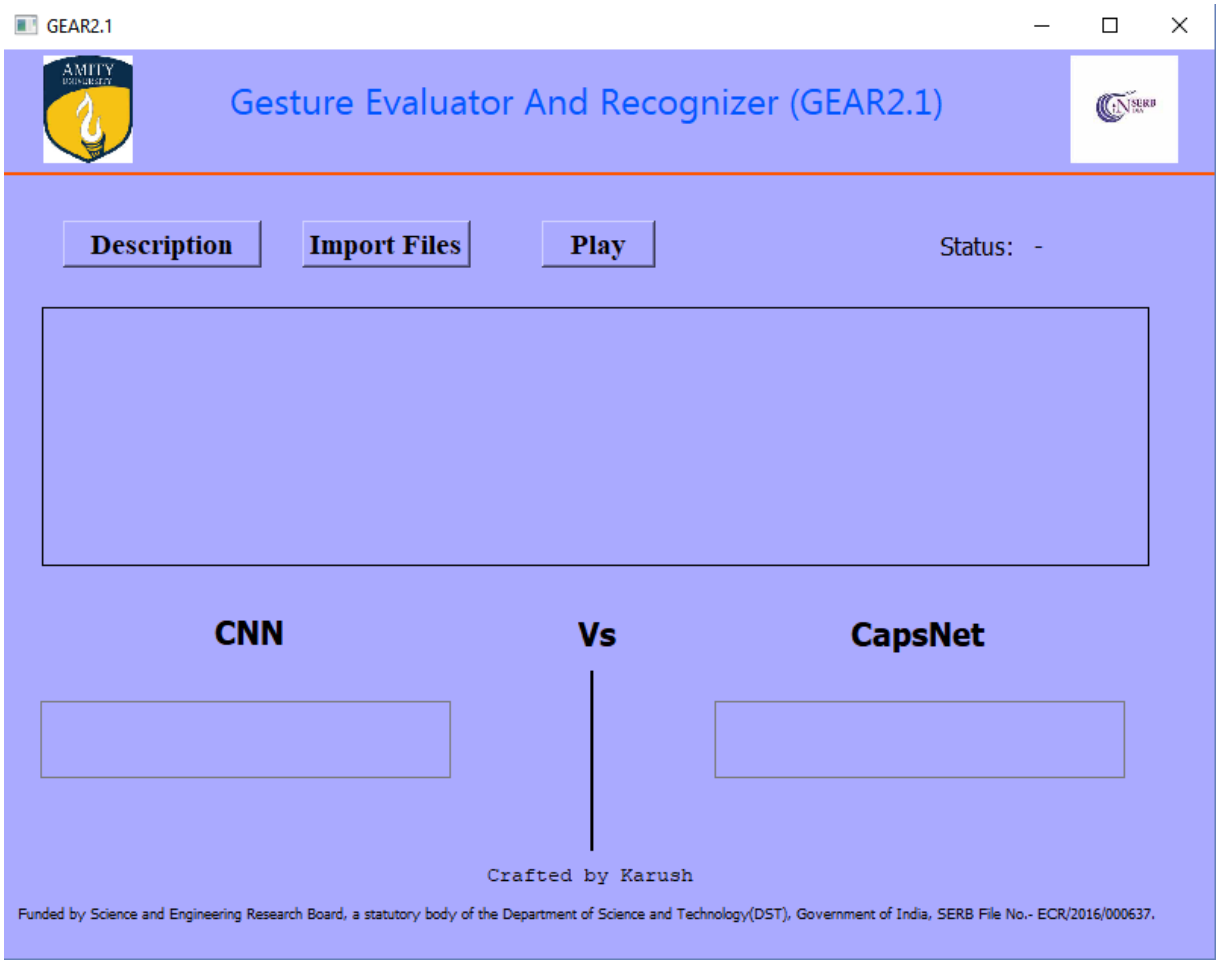


Fig 5.a Usage of GEAR2.1 in real-time

The user may start by reading the description about the software. The description provides the user with the steps by which GEAR2.1 operates. By clicking on the description button the steps appear inside the display area (outlined by black). Following is the message prompted along with the steps-

GEAR2.1 recognizes sentences from Indian Sign Language by making AIs compete with each other.

To play the Game follow these simple steps-

1. Click on ‘Import to import’ the AI files.

2. Select the sentence you wish to perform from the sentence list.

3. Click on ‘Play’ and perform the sentence with gaps between each gestures.

4. When finished, keep your hand in the final position.

Fig 5.b provides the response obtained on clicking the ‘Description button’.

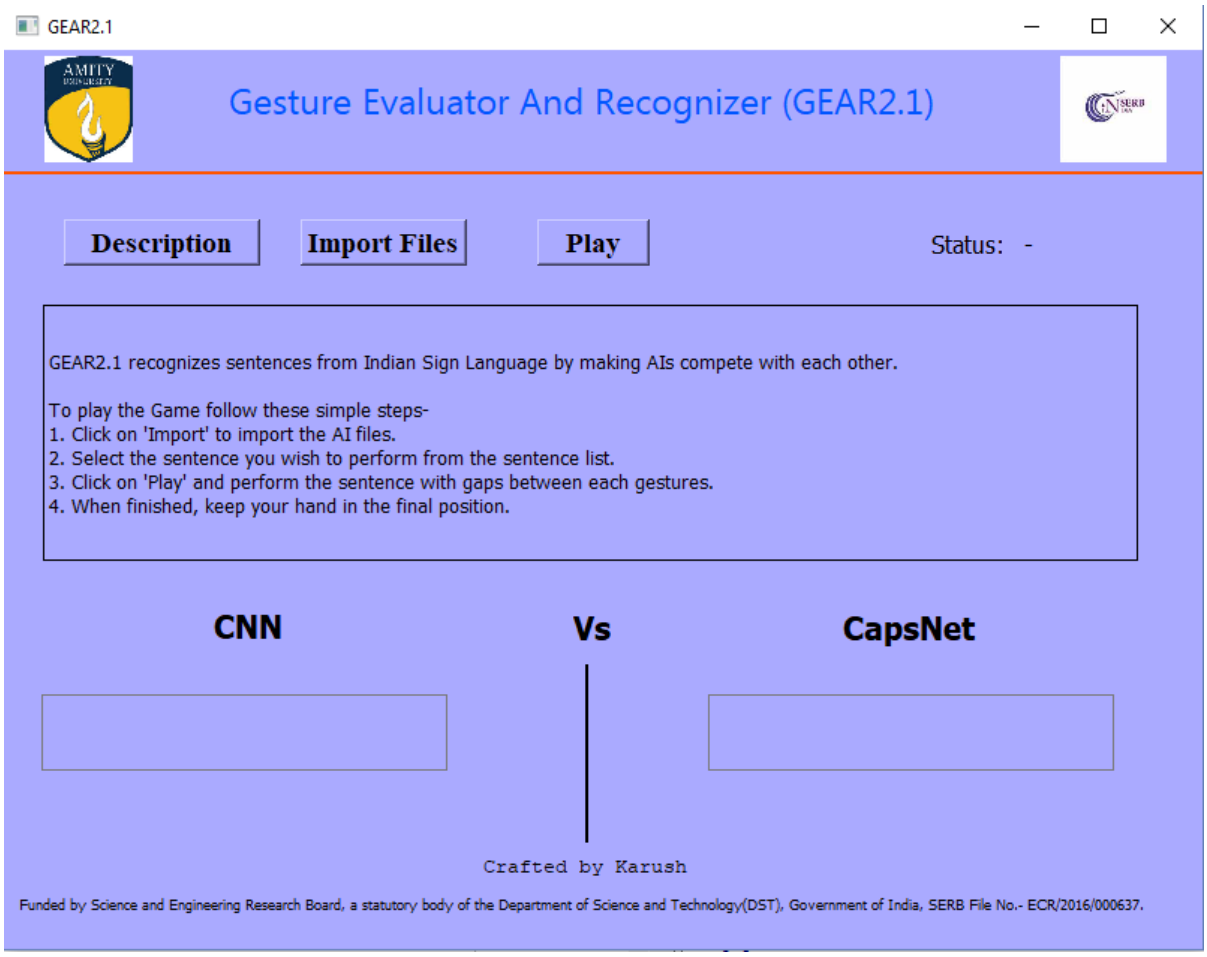


Fig 5.b Description provided by GEAR2.1 to the user

Once the user has read the description and understood the steps, he/she can proceed to import the associated files. The files imported here consist of the AI models, CNN and CapsNet and their weights. So, a total of 4 files are imported, CNN, optimized weights of CNN, CapsNet and optimized weights of CapsNet. All these files are used

to recognize the sentence by virtue of the IMU signals. The weights activate the imported AI which then predicts the result based upon its optimization.

Fig 5.c denotes the message provided to the user upon clicking the Import button. The status bar at the top right corner of the GUI reads imported, indicating that the necessary files have been imported.

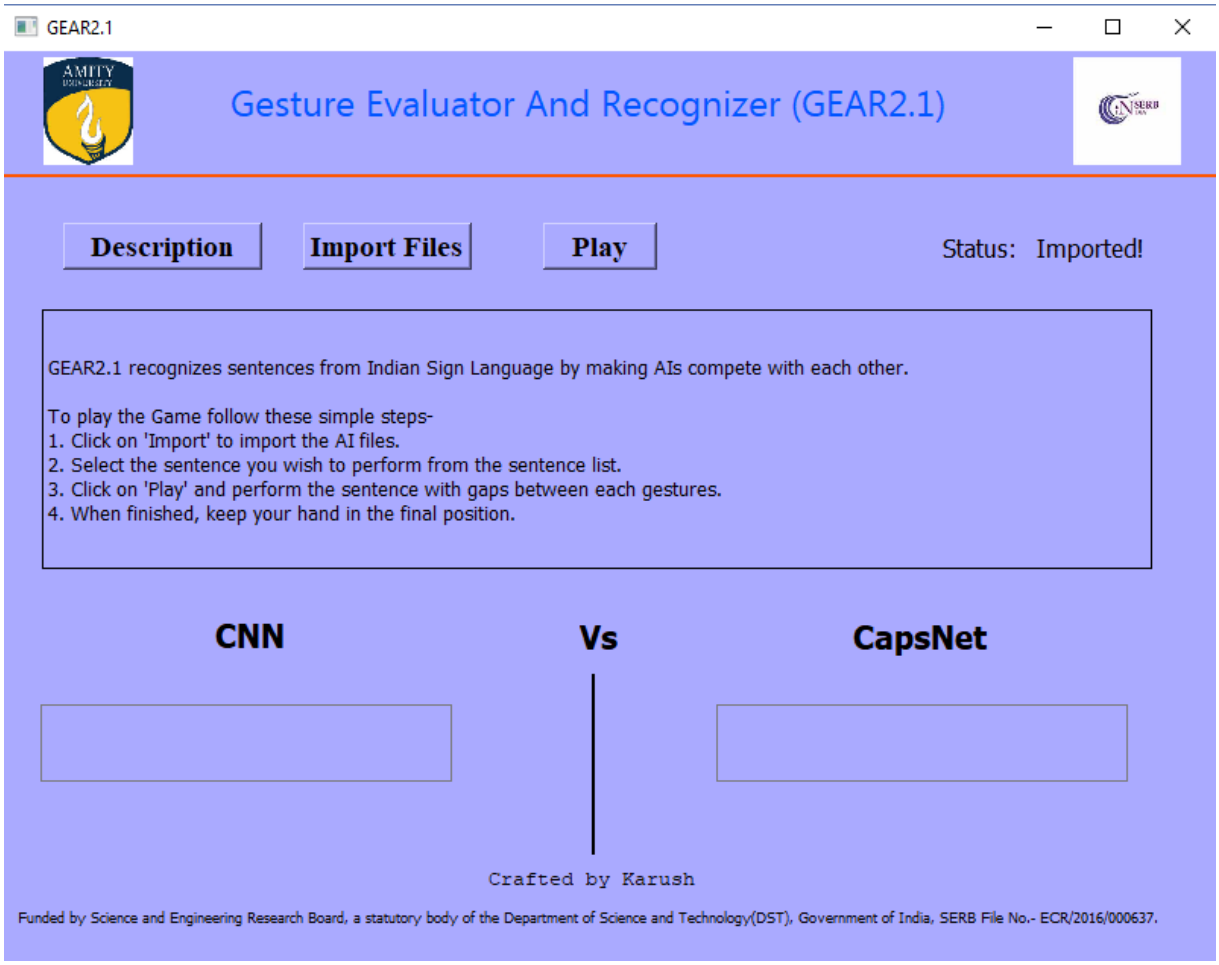


Fig 5.c Status bar updated upon clicking the Import files button

Completion of imports allows the user to begin the game. The play button located centrally governs the action of the game. Once the button has been clicked, it opens the serial port for communication. Stream of data representing the motion of the hand by virtue of IMUs flows in and starts getting updated in real-time. Values are stored in

an array of finite shape which is then processed for any abrupt spikes representing noise in the channel.

Stream of data collected is then fed into the pre-trained and optimized AIs which are set to compete with each other. Inputting the array into the algorithm begins the testing phase and both the models offer their predictions. The model whose prediction matches that of the true label wins the game and its competitor loses. However, the game makes an exception in cases when both the models give correct predictions or wrong predictions altogether. In such cases, the game is declared to be a draw.

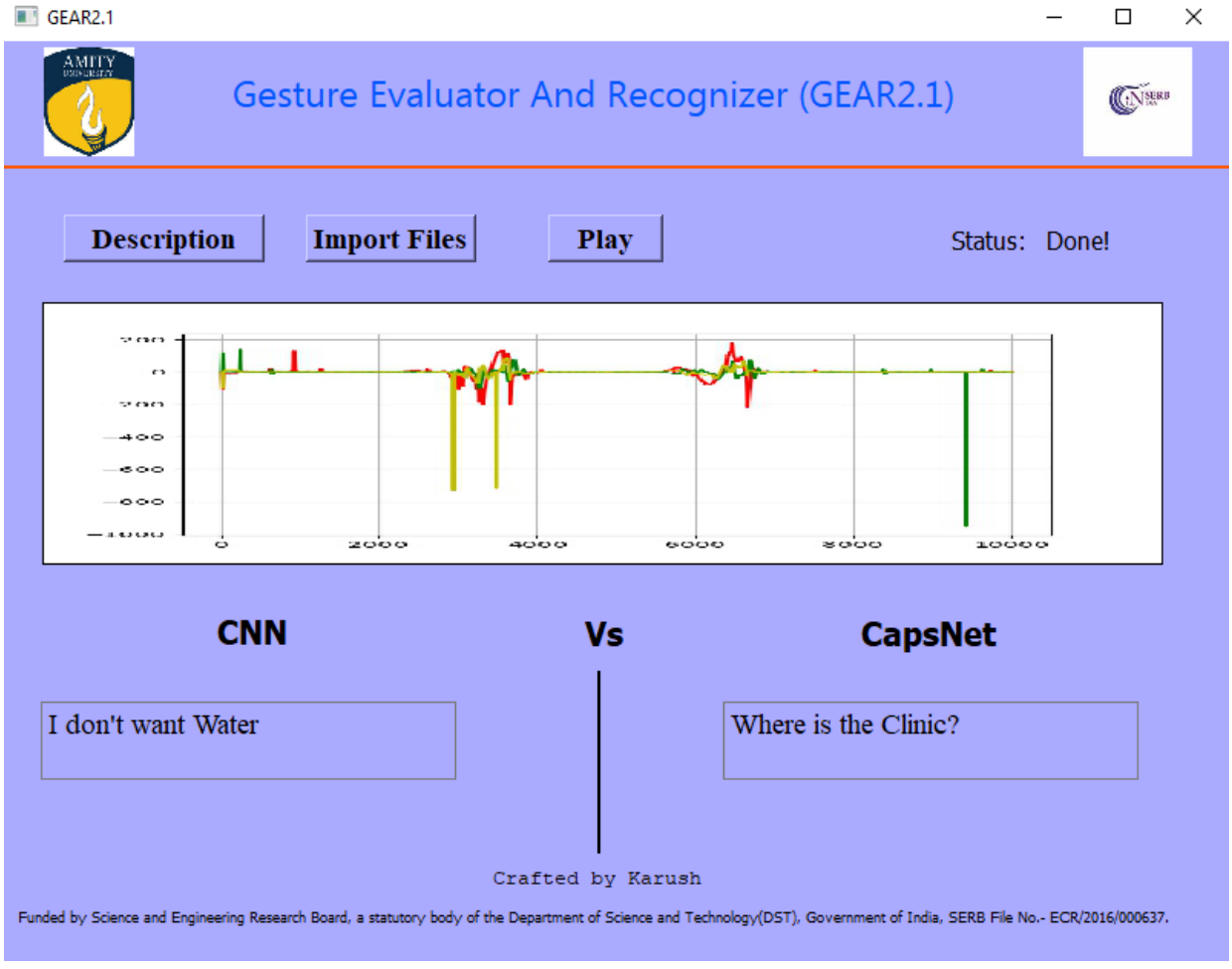


Fig 5.d Final window upon output of the Play button

Fig 5.d presents the final window upon output of predictions. The status bar reads ‘Done’ once both the AIs have offered their predictions. The serial port is then closed and the program finishes its execution. Final results are presented to the user in the text display are placed below each of the model’s name. The user can then compare and contrast the prediction of both the AIs with respect to each other in real-time. The user can also see the recorded signal in the main display window (with the black outline). Signal displayed here is that of gyrometer depicting the turn rate. During regions of activity, the turn rate tends to vary from a constant value and the graph moves up or down depending upon the direction of the turn. Turn rate denotes the rotational aspects of the gesture such as for how long the arm was rotated and at what specific angles.

CHAPTER-7: RESULTS AND DISCUSSION

7.1 ORIENTATION ESTIMATION

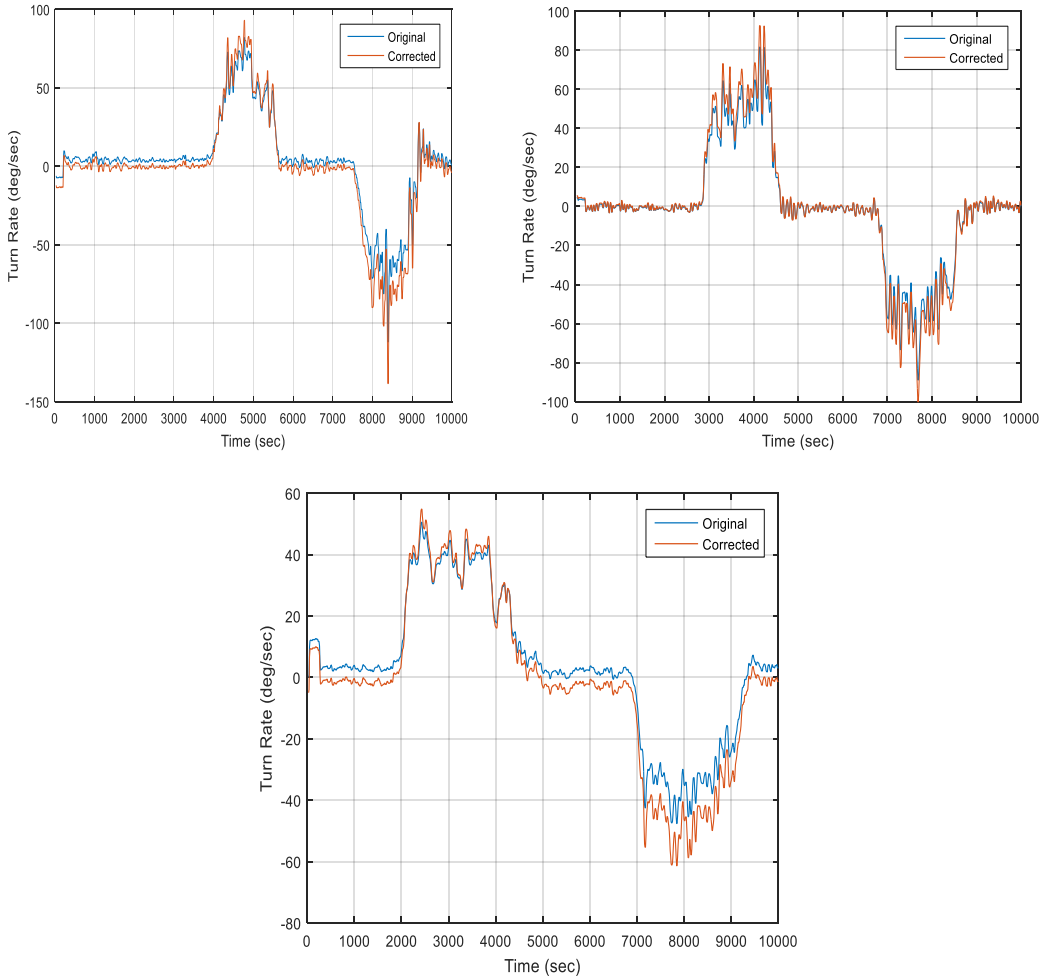


Fig 6. Comparison of original signal with the corrected signal prior to the removal of bias value and scaled correction with respect to (a) X-axis, (b) Y-axis, (c) Z-axis.

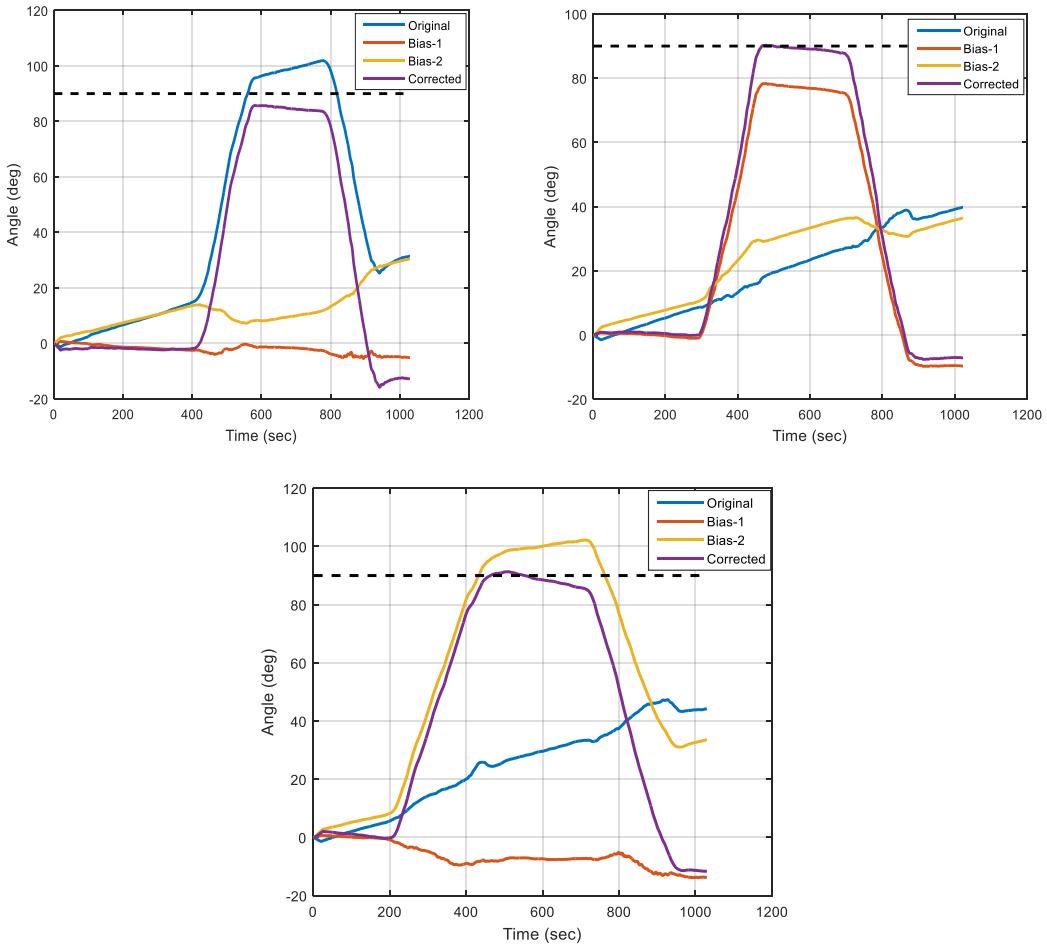


Fig 7. Comparison of original angles with the corrected angle variation along with propagation of bias values with respect to (a) X-axis, (b) Y-axis, (c) Z-axis.

Calibration of rotational aspects results in a more accurate estimation of orientation. Once the bias values have been removed from the original samples and the scaling entity has been multiplied, the signal undergoes a positional and size shift. These changes are visualized in Fig 6. which represents the comparison of the originally recorded signal to that of the corrected sequence. Considering the X-axis, subtraction of the bias estimate brings the initial value of the signal sequence towards 0. This can be asserted from the fact that the initial value of turn-rate should be 0 since there is no motion of the upper limb. Furthermore, scaling of the sequence by a definite value shifts the position of the samples from the non-calibrated trace. Corresponding to each axis, significant shifting is observed which is further used to estimate the angles from sequence values by cumulatively integrating the sampled values.

Cumulative integration of sensor values obtained after correction is carried out. Physical significance of a cumulative summation indicates the area swept by the surface. In the case of one dimensional inputs such as IMUs, cumulative integral yields the orientation in the form of angles along three-dimensional cartesian axes. Comparison of corrected estimation, i.e.- estimation carried out after calibration is carried out with respect to non-calibrated estimation. Fig 7. represents the comparison between calibrated and non-calibrated orientation estimation in the form of angles for each axis. For a given axis, say X-axis, the corrected estimation is more accurate when compared to the non-calibrated estimation. Angular values more closely resemble the angle of the sensor with respect to the X-axis. Non-calibrated values, on the other hand, depict significant amount of deviation from the true angular value of 90 degrees.

7.2 2x1 CNN ARRAY

The proposed CNN architecture is applied on the IMU dataset to classify the signed sentences. For comparison, the conventional CNN with no segregation distinction between general sentences and questions is also applied on the same dataset and the results for classification are compared with the proposed approach. The proposed 2x1 1D array architecture results in an improvement of the recognition process when compared to the conventional CNN. Fig 8.a and Fig 8.b depict the variation of average classification accuracy over 50 iteration for training and validation phases, respectively. The classification accuracies improve as the number of epochs increase and finally stabilizes at around 40 epochs. As seen in Fig 8, at the end of 50 epochs, the classification accuracies achieved by all the three networks is above 90%. However, both the networks in the array, CNN-sentences and CNN-questions depict an increase in the performance, given same hyper-parameters and constraints for validation to both the models. The CNN-questions model achieves a higher classification accuracy in comparison to the accuracies achieved using CNN-sentences and CNN-conventional.

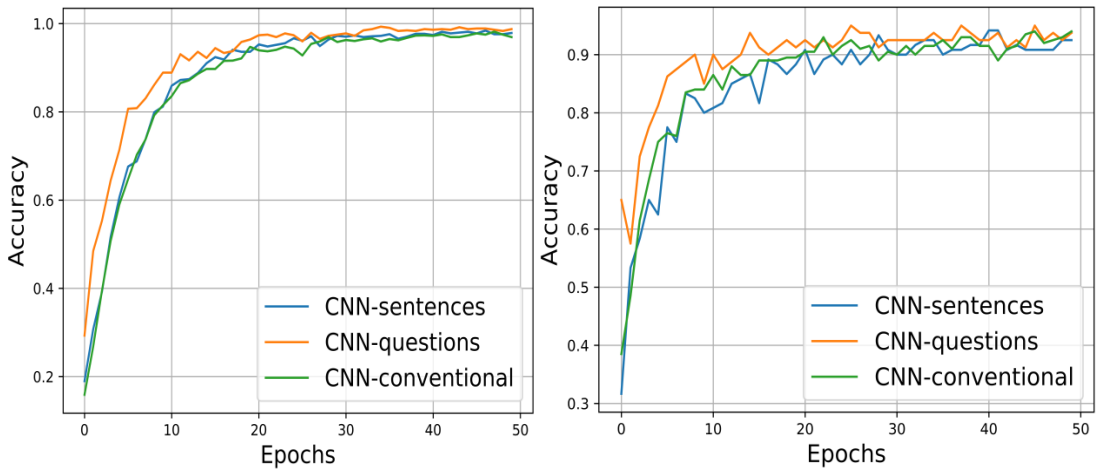


Fig 8. Variation of classification accuracy measured over 50 iterations (a) Training Accuracy, (b) Validation Accuracy

The variation of optimization losses of the considered models is compared during training and validation phases in Fig 9.a and Fig 9.b, respectively. The minimal gradient optimization is observed in the case of CNN-questions is observed in

comparison to the optimization for CNN-sentences and CNN-conventional. Fig 10.a and Fig 10.b represent the variation of false predictions offered by models during training and validation phases respectively. False predictions are employed as a validation metric here primarily for the assessment of optimization loss. A lower gradient initialization at the start of each epoch may not indicate fewer misclassified entities. Thus, falsely predicted values are plotted separately for the evaluation of dynamic optimization. As seen in Fig 10.a and Fig 10.b, false predictions for CNN-questions depict minimum misclassified entities followed by CNN-sentences and CNN-conventional. Thus, corresponding to each metric used for training and validation, CNN-questions presents peak variation of improved recognition followed by CNN-sentences and CNN-conventional. Improvement of CNNs in the 2x1 array in comparison to CNN-conventional assert the suitability of the architecture.

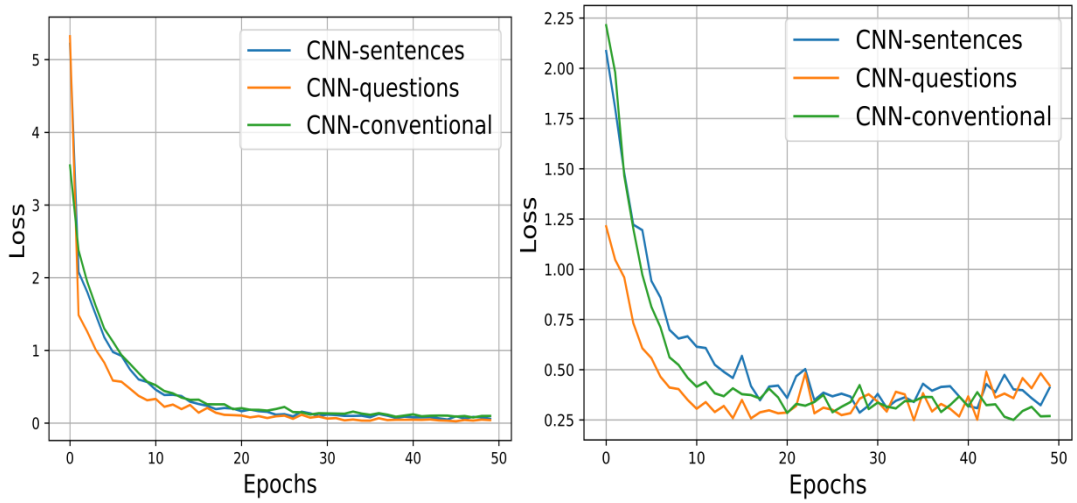


Fig 9. Variation of Loss measured over 50 iterations (a) Training Loss, (b) Validation Loss

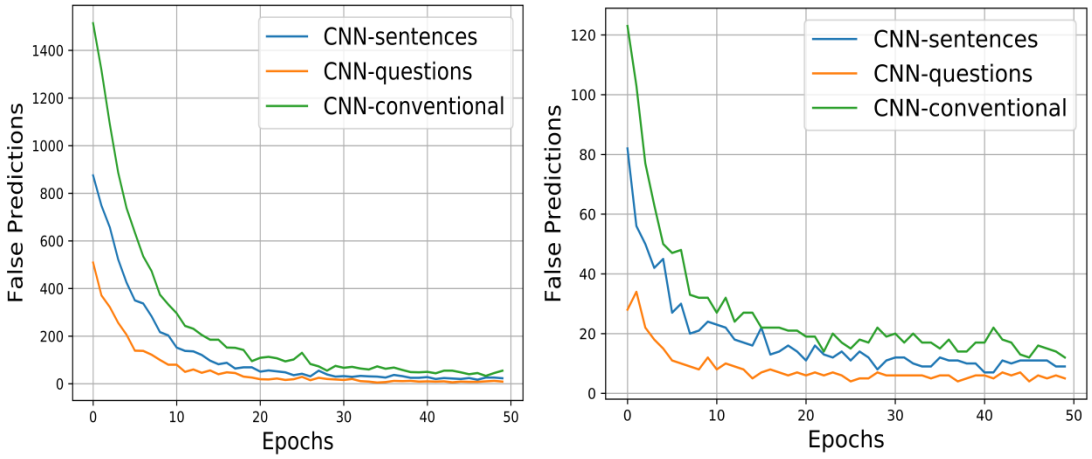


Fig 10. Variation of False Prediction measured over 50 iterations (a) During Training, (b) During Validation

Table I highlights the peak performance values for all the metrics used to assess the models at the end of 50 iterations. Optimization loss presents a peak value of 0.02 and 0.24 during training and validation in the case of CNN-questions which is the highest among the three models followed by CNN-sentences and CNN-conventional. The optimization loss for validation phase are comparable. This may be due to the limited amount of data used for validation. In future, the database will be expanded to include more subjects and more sentences.

Table 1. Peak Performance values for IMU Signal Recognition (obtained over 50 iterations)

Model	**Optimization Loss**		**False Predictions**		**Classification Accuracy**	
	Training	**Validation**	**Training**	**Validation**	**Training**	**Validation**
CNN-sentences	0.04	0.24	17	7	98.42%	94.20%
CNN-questions	0.02	0.24	5	4	99.30%	95.00%
CNN-conventional	0.07	0.25	33	12	98.16%	93.50%

As seen in Table I, the number of false predictions is fewer in the case of CNN-questions (5 for training and 4 for validation) in comparison to CNN-sentences (17 for training and 7 for validation) and CNN-conventional (33 for training and 12 for validation). Average classification accuracy values depict accurate recognition values of 99.30% and 95.00% for CNN-questions during training and validation, followed by 98.42% and 94.20% for CNN-sentence during training and validation and 98.16% and 93.50% for CNN-conventional during training and validation. Higher peak performance values in the case of CNN-questions and CNN-sentences in comparison to CNN-conventional depict the suitability of the proposes array architecture approach.

7.3 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 for CNN is empirically determined to be softmax, since it yields better learn weights over each successive iteration. 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 11.a and Fig 11.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 11., 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 12.a and Fig 12.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 13.a and Fig 13.b indicate the number of false predictions over each iteration 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.

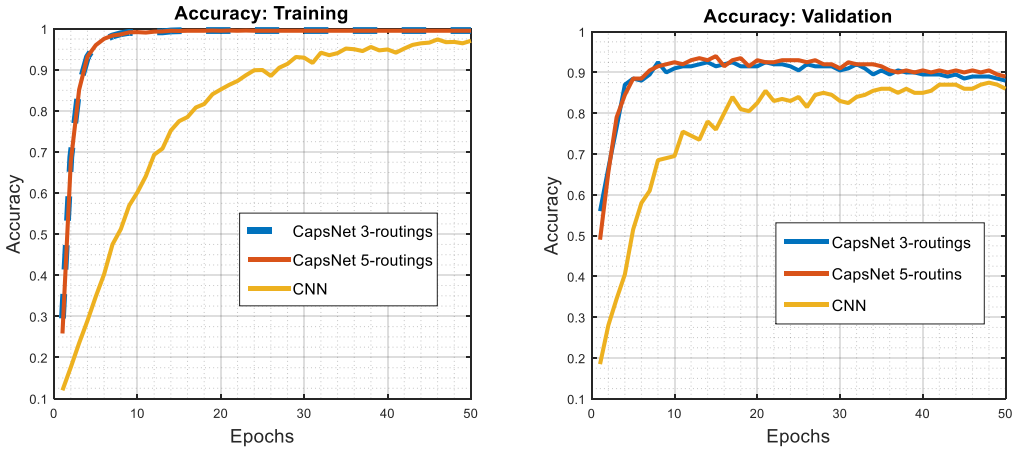


Fig 11. Average Accuracy value variation over 50 iterations for (a) Training and (b) Validation

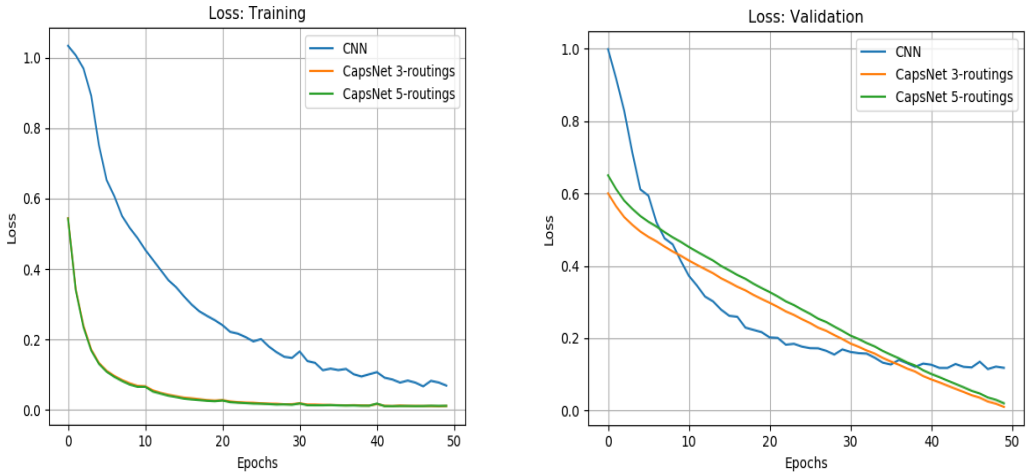


Fig 12. Optimization loss variation (categorical crossentropy) over 50 iterations for (a) Training and (b) Validation

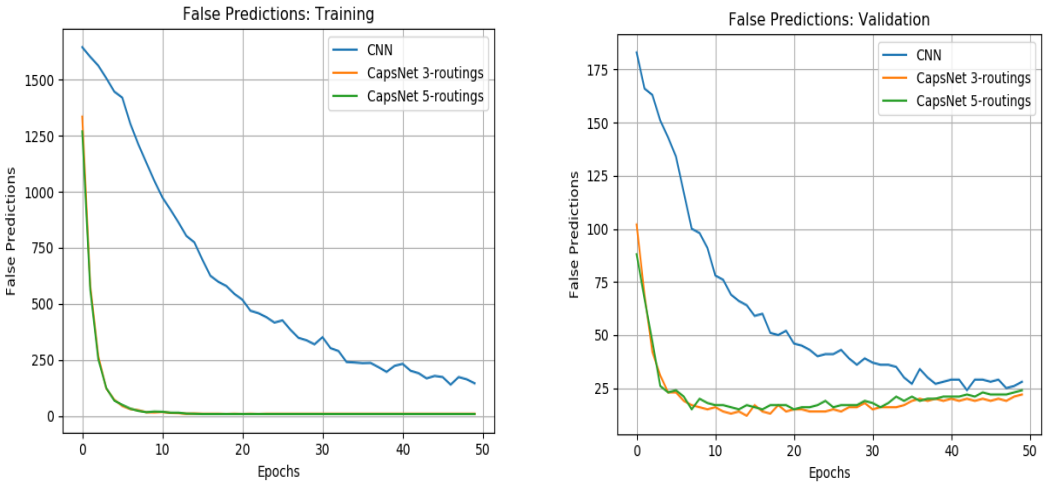


Fig 13. False Predictions variation over 50 iterations for (a) Training and (b) Validation

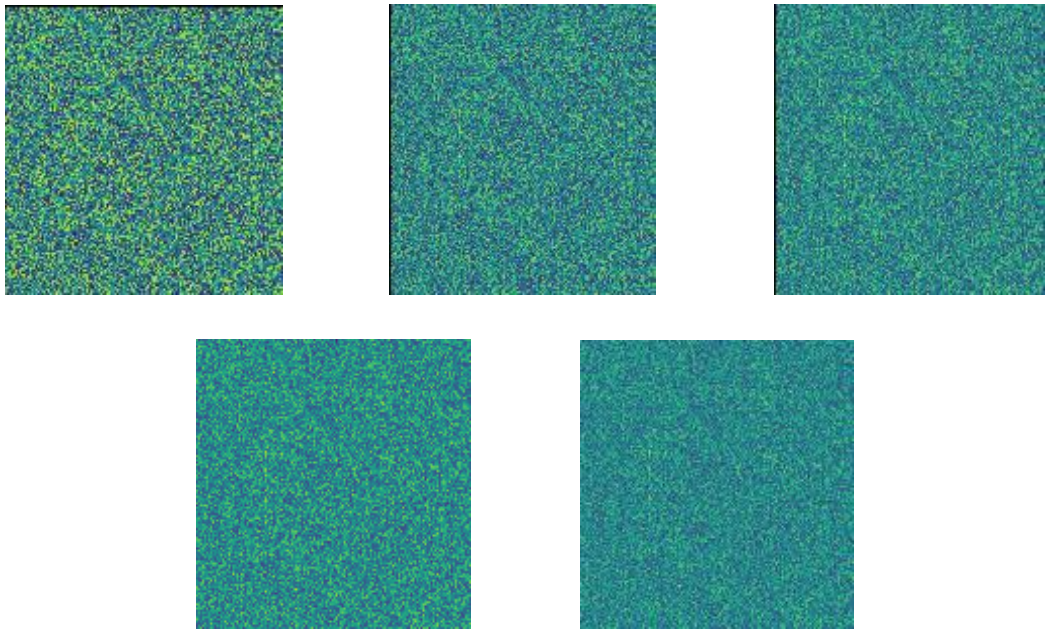


Fig 14. Learnt Activations at the final layer of CNN over (a) 1 iteration, (b) 10 iterations, (c) 20 iterations, (d) 30 iterations and (e) 40 iterations

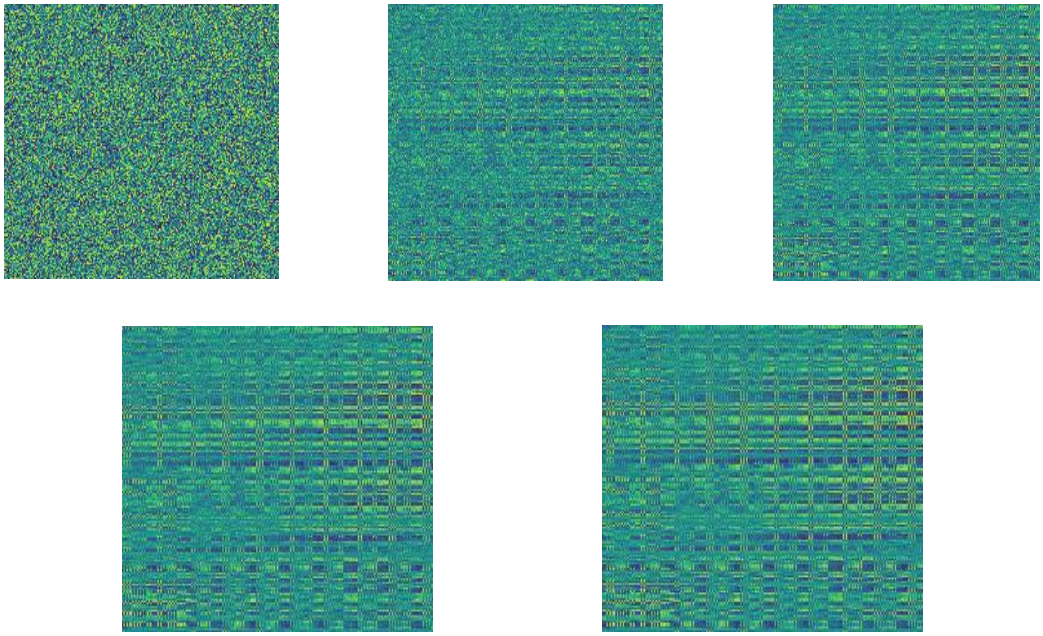


Fig 15. Learnt Activations at the final layer of CapsNet over (a) 1 iteration, (b) 10 iterations, (c) 20 iterations, (d) 30 iterations and (e) 40 iterations

Analysing learnt activations for the model plays a significant role in the evaluation of learning as these weights tend to change over each iteration [35]. Fig 14. and Fig 15. indicate the improvement in weights after every 10^{th} 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 15.a to Fig 15.d, the multiplicative values modify over passing iterations producing a regular pattern in Fig 15.d, which depicts the weights in the spatial domain and a definite structure in the background for CapsNet architecture. This definite structure indicates the activated units in the layer. However, no such regular pattern is observed to emerge in the case of CNN (Fig 14.a to Fig 14.d). 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.

7.4 NON-COOPERATIVE GAMES

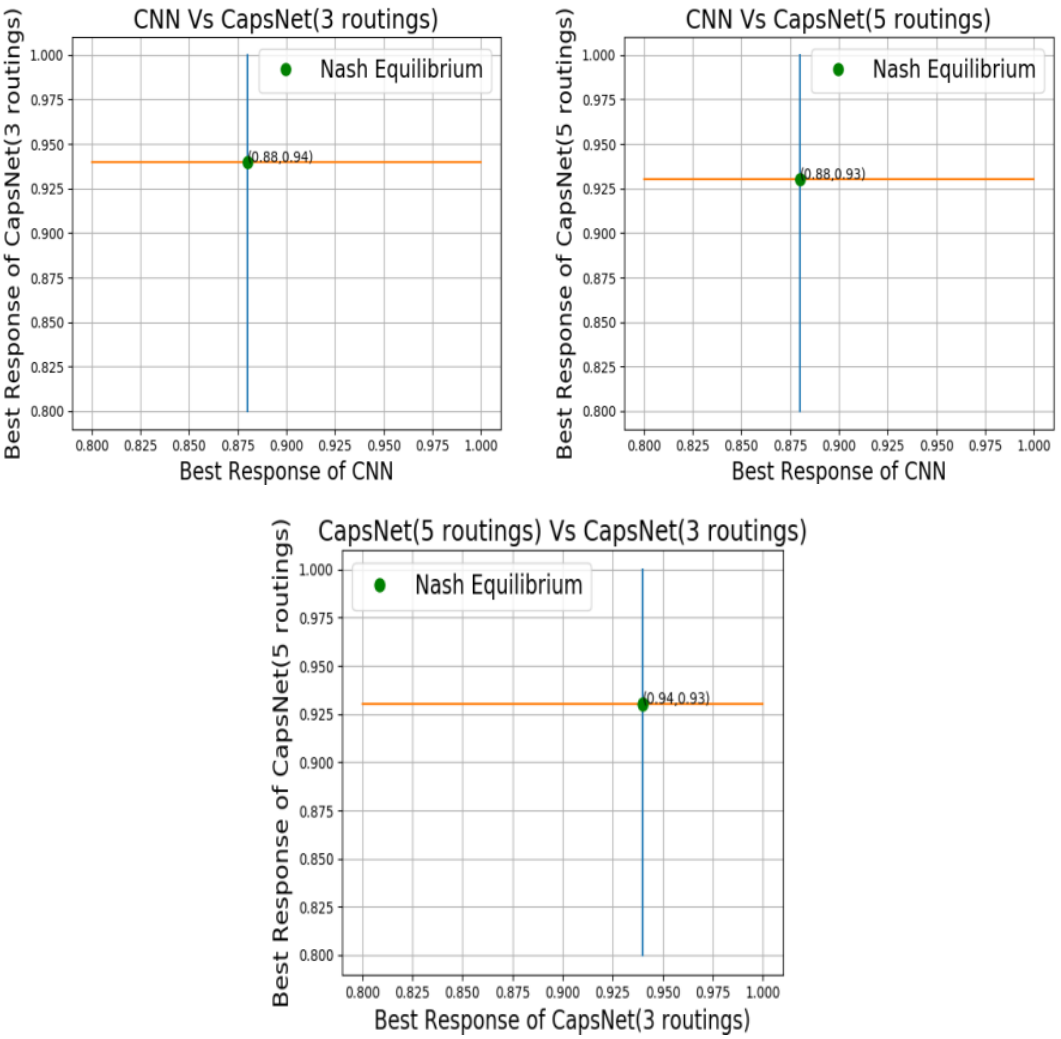


Fig 16. Nash Equilibrium obtained between best responses for (a) CNN Vs. CapsNet (3 routings), (b) CNN Vs. CapsNet (5 routings) and (c) CapsNet (5 routings) Vs. CapsNet (3 routings).

The proposed two-player game designed as explained in Chapter 4, 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,

$$\textit{Game-1- P(CNN): } (c_j)_{CNN} = c_{true} \textit{ ; P(CapsNet-3): } (c_j)_{CapsNet\text{-}3} = c_{true} \quad (16)$$

$$\textit{Game-2- P(CNN): } (c_j)_{CNN} = c_{true} \textit{ ; P(CapsNet-5): } (c_j)_{CapsNet\text{-}5} = c_{true} \quad (17)$$

$$\textit{Game-3- P(CapsNet-3): } (c_j)_{CapsNet\text{-}3} = c_{true} \textit{ ; P(CapsNet-5): } (c_j)_{CapsNet\text{-}5} = c_{true} \quad (18)$$

where $(c_j)_{CNN}$ denotes the class picked by CNN corresponding to the j^{th} sample in the validation set, $(c_j)_{CapsNet\text{-}3}$ denotes the class picked by CapsNet with 3-routings corresponding to the j^{th} sample in the validation set and $(c_j)_{CapsNet\text{-}5}$ denotes the class picked by CapsNet with 5-routings corresponding to the j^{th} sample in the validation set. Collective representation of these games in set notation is expressed by Eq. 19.

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

$$W_3 = \{P(CapsNet\text{-}3), P(CapsNet\text{-}5)\} \quad (19)$$

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 [36],

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.

The idea of Nash Equilibrium corresponds to the point where both the players in the constructed games have adopted and optimized the single adopted strategy. Once the strategies have been optimized the models are said to have their best responses during the interaction. Relative performance between the two competing models in games is thus assessed by visualizing their best responses. Fig 16. 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.

Table 2. Performance Comparison of CapsNet architecture with CNN

	Optimization Loss		**Misclassified Outputs**		**Classification Accuracy**		**Nash Equilibrium**	
Architecture	**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

Table 2 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.

CONCLUSION

Recognition of sign language has gained popularity as one of the primary applications of gesture recognition. Intelligent algorithms in corroboration with wearable sensors pave the way for accurate recognition of signs in the form of sentences. In this work, recognition of sentences signed according to the Indian sign language is performed using signals recorded from a wearable IMU device. The database of IMU signals is recorded for 20 different sentences, with 10 subjects performing 10 repetitions of each sentence. The sentence recognition is carried out using a novel one-dimensional CapsNet architecture. Performance of the model is assessed when 3 or 5 dynamic routings are used in the CapsNet architecture. The performance of the proposed CapsNet architecture is also compared with the conventional CNN in terms of quantitative measures such as classification accuracy, evolution of loss function and number of false predictions. Improved accuracy value 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% accuracy values, respectively. Learning of the architecture is validated by observing spatial activations depicting excited units at the final layer.

The work also presents a novel one-dimensional CNN array architecture for the recognition of sentences signed according to the Indian Sign Language. Signals for the signed sentences are recorded by using a custom designed wearable IMU device. The dataset is split into general sentences and questions. Two CNNs in the array recognize the general sentences and questions separately. Performance of the proposed array architecture is compared to a conventional CNN which is trained to recognize any sentence from the complete dataset. The number of false predictions is less for CNN-array, which is 17 and 5 for CNN-sentences and questions, respectively, as compared to conventional CNN where it is 33 during training. During validation, peak classification accuracy values of 95.00% for CNN-questions and 94.20% for CNN-sentences in the deep array architecture in comparison to 93.50% for the conventional CNN validate the suitability of the proposed approach.

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. CapsNet architecture presents a better value of the best response at Nash Equilibrium asserting the suitability of the proposed approach.

FUTURE PROSPECTS

In future we wish to expand the database by including more subjects and more sentences. The Indian Sign Language is a vast system of communication and has various sentences in the form of assertions and questions. We wish to cover these sentences as well. We also aim to decrease the pause between signs to achieve continuous sign language recognition using even more optimized algorithms in real-time. Furthermore, we wish to deploy the constructed CapsNet architecture using cloud platform for public usage. This way, research groups and deep learning enthusiasts would be able to study and make use of the algorithm for future work.

REFERENCES

[1] Luan Van Nguyen and Hung Manh La, “A Human Foot Motion Localization Algorithm Using IMU”, 2016 American Control Conference (ACC), IEEE, July, 2016, pp. 4379-4384.

[2] Mohammed M. Hamdi, Mohammed I. Awad, Magdy M. Abdelhameed and Farid A. Tolbah, “Lower Limb Motion Tracking Using IMU Sensor Network.”, 2014 7th Cairo International Biomedical Engineering Conference, IEEE, December, 2014, pp. 28-33.

[3] Dimitrios Sikeridis and Theodore A. Antonakopoulos, “An IMU-based Wearable System for Automatic Pointing during Presentations”, Image Processing & Communications, vol. 21, no. 2, pp.7-18.

[4] Kaiyuan Zhu, Liang Shi, “Motion Control in VR — Real-time Upper Limb Tracking via IMU and Flex Sensor”, IEEE, 2016, pp. 1-5.

[5] Mehrez Boulares, Mohamed Jemni, “3D motion trajectory analysis approach to improve Sign Language 3D-based content recognition”, International Neural Network Society Winter Conference (INNS-WC 2012), Elsevier, 2012, pp. 133-143.

[6] Purva C. Badhe, Vaishali Kulkarni, “Indian Sign Language Translator using gesture recognition algorithm”, International Conference on Computer Graphics, Vision and Information Security (CGVIS), IEEE, November, 2015.

[7] Jian Wu, Lu Sun, and Roozbeh Jafari, “A Wearable System for Recognizing American Sign Language in Real-Time Using IMU and Surface EMG Sensors”, IEEE Journal of Biomedical and Health Informatics, Vol. 20, No. 5, September 2016, pp. 1281-1290.

[8] Angkoon Phinyomark, Rami N. Khushaba and Erik Scheme, “Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors”, Sensor, MDPI, June, 2018, pp. 1-17.

[9] Suncheol Kwon, Jung Kim, “Real-Time Upper Limb Motion Estimation From Surface Electromyography and Joint Angular Velocities Using an Artificial

Neural Network for Human–Machine Cooperation", IEEE Transactions on Information Technology and Biomedicine, IEEE, May, 2011.

[10] Wei-Long Zheng and Bao-Liang Lu, "Investigating Critical Frequency Bands and Channels for EEG-Based Emotion Recognition with Deep Neural Networks", IEEE Transactions on Autonomous Mental Development, Vol. 7, No. 3, September 2015, pp. 162-175.

[11] Naser El-Sheimy, Kai-Wei Chiang, and Aboelmagd Noureldin, "The Utilization of Artificial Neural Networks for Multisensor System Integration in Navigation and Positioning Instruments", IEEE Transactions on Instrumentation and Measurement, Vol. 55, No. 5, October 2006, pp. 1606-1615.

[12] Karush Suri, Rinki Gupta, "Classification of Hand Gestures from Wearable IMUs using Deep Neural Network", 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT), IEEE, April 2018, pp. 45-50.

[13] Martin Wöllmer, Moritz Kaiser, Florian Eyben, Björn Schuller, Gerhard Rigoll, "LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework", Elsevier, Image and Vision Cmputing, Vol. 31, Issue 2, February, 2013, pp. 13-163.

[14] Karush Suri, Rinki Gupta, "Transfer Learning for sEMG-based Hand Gesture Classification using Deep Learning in a Master-Slave Architecture", 3rd International Conference on Contemporary Computing and Informatics (IC3I), IEEE, October 2018.

[15] Meiyin Wu, Li Chen, "Image recognition based on deep learning", 2015 Chinese Automation Conference, IEEE, January, 2016.

[16] Matthew Y. W. Teow, "Understadning neural networks using a minimal model for handwritten digit recognition", 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS), IEEE, October, 2017.

[17] Dan Li ; Jianxin Zhang ; Qiang Zhang ; Xiaopeng Wei, "Classification of ECG signals based on 1D convolution neural network", 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), IEEE, October, 2017.

[18] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, Aleksander Madry, “A Rotation and a Translation Suffice: Fooling CNNs with Simple Transformations”, December, 2017.

[19] Sara Sabour, Nicholas Frosst, Geoffrey E. Hinton, “Dynamic Routing Between Capsules”, 31st Conference on Advances in Neural Information Processing Systems (NIPS 2017), pp. 1-11.

[20] Canqun Xiang ; Lu Zhang ; Yi Tang ; Wenbin Zou ; Chen Xu, “MS-CapsNet: A Novel Multi-Scale Capsule Network”, IEEE Signal Processing Letters, Volume: 25, Issue: 12 , Dec. 2018.

[21] Frans A. Oliehoek, Rahul Savani, Jose Gallego-Posada, Elise van der Pol, Edwin D. de Jong, Roderich Groß, “GANGs: Generative Adversarial Network Games”, Cornell Library, April 2017, pp. 1-9.

[22] Dale Schuurmans, Martin Zinkevich, “Deep Learning Games”, 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain., pp. 1-9.

[23] Lukun Wang, “Recognition of Human Activities Using Continuous Autoencoders with Wearable Sensors”, Sensors, February 2016.

[24] Zhenyu He, Lianwen Jin, “Activity Recognition from acceleration data Based on Discrete Consine Transform and SVM”, IEEE International Conference on Systems, Man, and Cybernetics, October, 2009.

[25] Wen-Chang Cheng and Ding-Mao Jhan, “Triaxial Accelerometer-Based Fall Detection Method Using a Self-Constructing Cascade-AdaBoost-SVM Classifier”, IEEE Journal of Biomedical and Health Informatics, Vol. 17, No. 2, March 2013

[26] M.Murugappan, “Electromyogram Signal Based Human Emotion Classification using KNN and LDA”, IEEE International Conference on System Engineering and Technology (ICSET), 2011.

[27] Pinky Paul, Thomas George, “An Effective Approach for Human Activity Recognition on Smartphone”, IEEE International Conference on Engineering and Technology (ICETECH), March, 2015.

[28] Kilian Forster, Samuel Monteleone, Alberto Calatroni, Daniel Roggen, Gerhard Troster, “Incremental kNN classifier exploiting correct - error teacher for activity

recognition", Ninth International Conference on Machine Learning and Applications, 2010.

[29] Qiuping Wu, Ruonan Wu, Fengtian Han and Rong Zhang, "A Three-Stage Accelerometer Self-Calibration Technique for Space-Stable Inertial Navigation Systems", Sensors, August, 2018, pp. 1-16.

[30] Mark Looney, "A Simple Calibration For MEMS Gyroscopes", EDN Europe, July 2010, pp. 28-31.

[31] Faculty of Disability Management and Special Education (FDMSE), "Indian Sign Language", Ramakrishna Mission Vivekananda University, 2018.

[32] Diederik P. Kingma , Jimmy Lei Ba , "Adam: A Method for Stochastic Optimization", Internaional Conference on Learning Representations (ICLR), arXiv, 2015, pp. 1-15.

[33] Juan C. Burguillo, "Using game theory and Competition-based Learning to stimulate student motivation and performance", Elsevier, Computers and Education, Vol. 55, Issue 2, 2010, pp. 566-575.

[34] Per-Arne Andersen, Morten Goodwin, Ole-Christoffer Granmo, "Deep RTS: AGame Environment for Deep Reifnorcement Learning in Real-Time Strategy Games", International Conference on Computational Intelligence and Games (CGIS), IEEE, 2018, pp. 1-8.

[35] Bolei Zhou, Agata Lapedriza1, Jianxiong Xiao, Antonio Torralba1 and Aude Oliva1, "Learning Deep Features for Scene Recognition using Places Database", 27th International Conference on Advances in Neural Information Processing Systems (NIPS 2014), pp. 1-9.

[36] Tansu Alpcan, Lacra Pavel, "Nash equilibrium design and optimization", International Conference on Game Theory for Networks, IEEE, May 2009.

APPENDIX A. *List of ISL Sentences*

Table 3. List of ISL Sentences

Class Label	English Sentence	ISL Translation
1	**I Need Help**	**I+Need+Help**
2	**She Needs Help**	**She+Need+Help**
3	**He Needs Help**	**He+Need+Help**
4	**They Need Help**	**They+Need+Help**
5	**I want Water**	**I+Need+Water**
6	**I don't want Water**	**I+Need+Water+No**
7	**I want Medicine**	**I+Need+Medicine**
8	**I don't want Medicine**	**I+Need+Medicine+No**
9	**I want one Bread**	**I+Need+Bread+1**
10	**I like Bread**	**I+Like+Bread**
11	**He likes Bread**	**He+Like+Bread**
12	**They like Bread**	**They+Like+Bread**
13	**What is your Name?**	**Your+Name+What**
14	**What are their Names?**	**They+Name+What**
15	**What is your Father's Name?**	**Your+Father's Name+What**
16	**What is the time?**	**Time+What**
17	**Where is the Clinic?**	**Clinic+Where**
18	**Where is the Doctor?**	**Doctor+Where**
19	**Where is the Bank?**	**Bank+Where**
20	**Where is the Meat Shop?**	**Meat Shop+Where**

APPENDIX B. *Project Plan*

AMITY UNIVERSITY
UTTAR PRADESH

AMITY SCHOOL OF ENGINEERING & TECHNOLOGY

Project Synopsis

(B. Tech (ECE)/B. Tech (ECE)+MBA/ B. Tech (ECE)-3C/B. Tech (ECE)-Evening)

Project Guide Allotted (For

Department Use):

Dr. Rinki Gupta

Academic Session: 2018-19

Project Title: Real Time Analysis of Hand Motion using Low Cost IMUs

Project Team:

Programme:-		Year/Semester:-	
S. No.	Enrollment No.	Name	Signature
1.	A2305115034	Karush Suri	

Objectives:

- Analysis of hand motions using data from inertial measurement unit (IMU) consisting of triaxial accelerometer and triaxial gyroscope
- Development of a low cost IMU system for data acquisition during various hand gestures.

- Estimation of orientation of the hand in real time.
- Assessing the performance of the developed algorithm for different hand gestures, different IMU placement and multiple subjects.

Abstract/Project summary (at least 250 words):

Gesture recognition has gained significant importance in the past decade. With the advent of advanced biomedical techniques and unsupervised learning algorithms, progress of the analysis of hand motion signals is evident. Hand motion may be studied in the form of position, orientation and the intensity of the muscle. Of these, the orientation aspect contains significant processing and gives a concise idea about the rotation of the limb. It is essential to accurately evaluate the variations in rotational aspect of the forearm. Moreover, a real time evaluation of the gesture being performed plays a significant role in making the process more dynamic and suitable to the needs of the subject. Combining these elements of limb movements may help in language translation, arm amputee treatment and gesture controlled devices. Such a human-machine interface can successfully deliver prosthetic tools, smart security systems, optimized language translators and health tracking bands.

The project proposes a real time system capable of analyzing motion of various hand gestures. Study of these gestures may be carried out in the form of signals. The system makes use of economical sensors such as Intrinsic Measurement Units (IMUs) capable of evaluating the motion of the limb through its orientation. An accurate estimation of hand movement can be obtained by acquiring signals from these low cost sensors such as accelerometer and gyroscope. Acquisition of these signals can then be carried out prior to processing. Processing of the acquired signals consists of orientation estimation determining the angular movements of the limb in real time. Validation of results can be carried out by comparing previously recorded gestures to that performed by the subjects in real time.

Such a system makes use of only the rotational aspects for determining the orientation of the limb and is further optimized in real time. Advances in the design may be carried out by assessing intensity of the muscles in the form of surface Electromyography (sEMG) and recognizing gestures performed by the subject using deep algorithms capable of learning the most intricate aspects of data.

Methodology to be adopted:-

- Acquisition of signals from triaxial accelerometer and triaxial gyroscopes for different hand motions.
- Processing of the acquired data in real time for pre-processing and Orientation and/or position estimation.
- Assess the estimated orientation to determine the type and nature of the gesture performed.
- Validating the results for various hand gestures recorded with multiple subjects and different IMU placement.

Resource requirement (Hardware & software etc):-

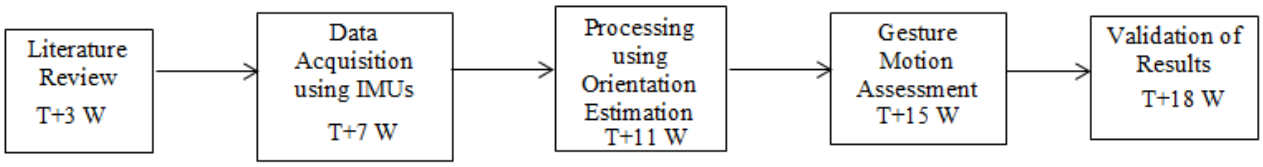
The project involves both hardware and software components.

- Hardware- GY80 multisensor consisting of triaxial accelerometer and gyroscope; Arduino microcontroller board.
- Software- Arduino IDE for acquiring hand motion data, MATLAB (Statistical Toolbox and Signal Processing Toolbox) for signal acquisition and processing.

Justification of the project:- Development of a real time system capable of assessing hand gestures is useful in developing assistive technology, such as sign language translation and in developing human machine interfaces for example in gaming applications. A real time acquisition algorithm would play a significant role in obtaining meaningful data from multiple sensors and processing it as per the need of the application.

PERT Chart/Schedule of project completion:-

T= Start date of the project, W=weeks



Signature(s) of the team member(s):

Date:

COMMENT BY EXTERNAL EXAMINER