Real-Time Translation of Indian Sign Language using LSTM

Ebey Abraham
Department of Computer Science and Engineering
Rajiv Gandhi Institute of Technology
Kottayam, India
ebey97@gmail.com

Akshatha Nayak
Department of Computer Science and Engineering
Rajiv Gandhi Institute of Technology
Kottayam, India
akshanayaki@gmail.com

Ashna Iqbal
Department of Computer Science and Engineering
Rajiv Gandhi Institute of Technology
Kottayam, India
ashnaibl@gmail.com

***Abstract*—Sign language is the only medium of communication for the speech-impaired community while the rest of the population communicate verbally. This project aims to bridge this communication gap by proposing a novel approach to interpret the static and dynamic signs in the Indian Sign Language and convert them to speech. A sensor glove, with flex sensors to detect the bending of each finger and an IMU to read the orientation of the hand, is used to collect data about the actions. This data is then wirelessly transmitted and classified into corresponding speech outputs. LSTM networks were studied and implemented for classification of gesture data because of their ability to learn long-term dependencies. The designed model could classify 26 gestures with an accuracy of 98%, showing the feasibility of using LSTM based neural networks for the purpose of sign language translation.**

Keywords—Indian Sign Language, LSTM, Gesture Recognition, Sensor Glove

I. INTRODUCTION

According to the 2011 census, approximately 7 million people in India are deaf and mute and 1.9 million have speech disability. Among them 63% are not employed and 30% have never attended school [1]. Because of their disability they can only communicate with each other using signs and gestures. While the speech impaired population depend on sign language for communication, the rest of the population communicate verbally. This creates a communication gap. This puts them at a disadvantage and they cannot avail the same education and job opportunities.

The Indian Sign Language(ISL) is a standard set of sign language gestures used by the speech impaired community across India. These ISL gestures can be classified into two categories; static and dynamic. Static gestures are those in which there is no movement of the hands while performing the signs. Most of the gestures for the alphabets in ISL are static signs. Dynamic gestures, on the other hand, involve hand movements while performing the gestures and constitute the majority of the ISL gestures.

A device capable of translating the Indian Sign Language to speech could efficiently bridge the communication gap and thus empower the speech-impaired to communicate with the rest of the population. This would help in improving their social integration and reform their lifestyle.

To be able to translate the Indian Sign Language, the device would need to recognize both the static and dynamic gestures in ISL. While standard neural networks can classify static gestures they cannot be used for classifying dynamic gestures. In dynamic gestures the reading at each time point is dependent on the previous readings resulting in sequential data. Since standard neural networks require that each reading be independent of the other readings they cannot be used for classification of sequential data.

Recurrent neural networks (RNNs) are connectionist models with the ability to selectively pass information across sequence steps while processing sequential data one element at a time [4]. Thus they can model input and/or output consisting of sequences of elements that are not independent. Recurrent neural networks can also simultaneously model sequential and time dependencies on multiple scales [4]. However, it has been observed that gradient descent of error criterion may become increasingly inefficient when the temporal span of dependencies increases [5].

This inefficiency can be overcome by using Long Short-Term Memory (LSTM), which are a special kind of RNN capable of learning long-term dependencies. LSTM is a recurrent network architecture that can learn to bridge time intervals in excess of 1000 steps even in case of noisy, incompressible input sequences, without loss of short time lag capabilities [6]. Due to this ability to learn long-term dependencies, LSTM based neural networks can be used to classify sensor data corresponding to sign language gestures to their respective speech labels and thus help in building a device capable of translating sign language in real-time.

In this paper, we present the design and working of a wearable glove that is capable of translating ISL signs to speech. The glove transmits the data from the embedded sensors to a processing unit like a PC or smartphone via Bluetooth. This data is then classified using an LSTM based neural network to give corresponding text and speech output.

II. RELATED WORKS

There have been primarily two approaches in solving the problem of gesture recognition. They can be broadly classified into the following categories:

1) **Vision Based:** In vision based solutions, sign language gestures are translated to speech using image processing and computer vision techniques. Dutta, Raju K, Kumar G S and Swamy B proposed a system in which feature points of training images are stored in a database[7]. Sign language translation is done by comparing features of input images with the database to find the best match. J. Singha and K. Das proposed a method using Eigen value weighted Euclidean distance for classification of image data of Indian Sign Language[8]. This included skin filtering, hand cropping, feature extraction and classification. Deo, Rangnesh and Trivedi proposed a method of using Hidden

978-1-7281-3694-3/19/$31.00 ©2019 IEEE

Markov models for recognizing dynamic hand gestures[2]. The paper also proposes a CNN-HMM hybrid mode which gives a better recognition accuracy than traditional HMM.

Although these vision based techniques can accurately translate sign language, it has a number of limitations in its practical implementation[3]. The main drawback of using this method is that the light intensity and image resolution affects the accuracy of the translation. This makes it unpredictable in real environments where we cannot control these parameters. Using images for training also reduced the accuracy of the model as extracting features from dynamic gestures is a complex problem[2].

2) **Sensor Based**: The other approach to sign language translation is placing sensors directly on the hand to estimate its orientation while making sign language gestures. Each gesture in the sign language produces a set of sensor values corresponding to that gesture. These sensor values are then used to classify the gestures into corresponding text and speech.

Abhishek, Qubeley and Ho presented a gesture recognition glove based on charge-transfer touch sensors[10]. The hand gestures cause the capacitive sensors to be selectively activated, thus employing a binary detection system. Thus the position of a finger is interpreted as binary 1 when the finger is unbent and binary 0 when the finger is bent. This approach could however only classify static gestures corresponding to letters and numbers and could not be extended to dynamic gestures[10].

Heera, Murthy, Sravanti and Salvi also proposed a sensor based approach using flex sensors and a MPU-6050 module to collect gesture data. The collected data is stored in a database. Sign language translation is done by comparing each action with the database to find the best match and thus recognize the gesture.[9]. Abhijith Bhaskaran K et al. also proposed a similar approach to gesture translation by comparing input with a database to identify the gesture. The difference was that this method used a state space representation of all the gestures instead of directly storing the data[11].

III. Hardware Design

A. Hardware Components

The Adafruit Feather Bluefruit nRF52 acts as the main microcontroller unit and also as a bluetooth module to send data wirelessly. The analog pins A1-A5 are used to read values from the flex sensors. Each of these analog pins reads the voltage value corresponding to a flex sensor. The MPU-6050 is used to understand the orientation of the hand while performing gestures. This is achieved by reading the accelerometer and gyroscope values off the MPU-6050 module. The MPU-6050 communicates with the Adafruit Feather unit through I2C protocol. For this we use the SDA and SCL pins in the MPU-6050 and the Adafruit Feather board. A flex sensor is a variable resistor. The resistance of the flex sensor increases as the body of the component bends. This is used to determine the bending of each finger as varying degrees of bending of the fingers will give different resistance values.

A 3.7V 500mAh Li-Ion battery is used for power supply, which can be directly connected to the Adafruit board. The MPU-6050 module and the voltage divider circuits are powered from the Adafruit board.

B. Voltage Divider Circuit

The flex sensor values are used to describe the extent to which each finger is bent. This is done by using the flex sensor in a voltage divider circuit and reading the voltage drop across the flex sensor using the analog pins of the Adafruit Bluefruit module. The voltage divider circuit is as shown in Fig. 1.

The voltage value V at the junction of the resistors R_1 and R_2 is given as:

$$\mathrm{V} = \frac{3.3R_2}{R_1 + R_2}$$

The flex sensors used for the glove has a resistance value of 10k when left flat and 30k when bent completely. Thus for a constant value of R_2 the maximum voltage value for V will be

$$V_{max} = \frac{3.3R_2}{10k + R_2}$$

and the minimum voltage will be

$$V_{min} = \frac{3.3R_2}{30k + R_2}$$

Thus the range of voltage values for V can be written as

$$V_{diff} = V_{max} - V_{min} = \frac{66R_2}{(10k + R_2)(30k + R_2)}$$

For accurately reading the voltage drop across the flex sensor, the aim is to maximize the value of V_{diff} to be able to precisely differentiate between the varying degree of bending of the fingers. The value of V_{diff} will be maximum when $\frac{d}{dR_2}V_{diff} = 0$.

$$\Rightarrow \frac{d}{dR_2}\left[\frac{66R_2}{(10k + R_2)(30k + R_2)}\right] = 0$$

$$\Rightarrow \frac{66(10k + R_2)(30k + R_2) - 66R_2(2R_2 + 40k)}{[(10k + R_2)(30k + R_2)]^2} = 0$$

$\Rightarrow (R_2)^2 - 300 * 10^6 = 0$
$\Rightarrow R_2 = 17.32k$ (Value of resistor cannot be negative)
$\Rightarrow R_2 \approx 18k$ (Nearest standard resistor value)

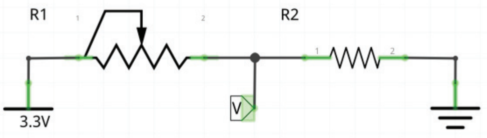

Fig. 1. Voltage divider circuit using flex sensor

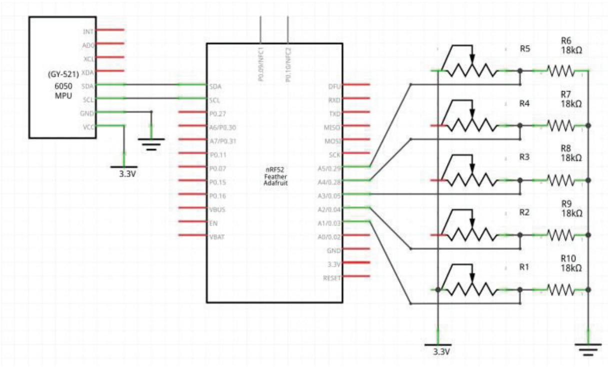

Fig. 2. Circuit diagram of the sensor glove

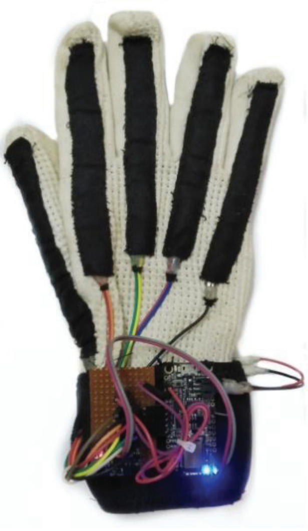

Fig. 3. Prototype of the designed glove

IV. Methodology

A. Reading Data from Hardware

The sensor data corresponding to the gestures is transmitted from the glove to processing device through Bluetooth communication using the Bluefruit LE Python library. The sensor data is sampled at the rate of 50 readings per second.

Since the sampling rate is different from the rate at which the data is transmitted from the micro-controller there needs to be a method to read data from the buffer at regular intervals. For this, threads have been implemented in Python. One thread will be reading data from the controller at the same rate at which it transmits data. This data is stored in a common memory buffer. A second thread will read the data stored in this memory buffer at the rate at which sampling is required, that is at 50 readings per second.

B. Data Collection

For training the deep learning model 40 samples each were collected for 26 commonly used signs in Indian Sign language. Every reading consists of 11 values, 6 of which are obtained from the IMU module, which includes x, y and z axis readings of accelerometer and gyroscope. The other 5 readings are obtained from the 5 flex sensors attached to the fingers of the glove.

Each sample for a sign consists of 50 consecutive readings obtained in a second, which correspond to the 50 time points that define the dynamic motion for a gesture. Thus a single sample consists of 50 rows (time points) and 11 columns (features), resulting in 2D data which is stored in a CSV file. The file name consists of the label for the sign and the time-stamp corresponding to the time of data collection for generating unique file names.

C. Data Preprocessing

The collected data was normalized using feature scaling. The scaling was done over the range of minimum and maximum values possible during normal usage of the glove. The flex sensor values for each sample were scaled between 0 and 1, with the maximum value of 610 set to 1, and the minimum value of 340 set to 0. The accelerometer and gyroscope values, on the other hand, were scaled between -1 and 1, with the maximum value of 32767 set to 1, and the minimum value of -32768 set to -1, for readings of all axes. Normalization of accelerometer and gyroscope values was done on a scale spanning across positive and negative values to retain the sign of readings, and thus, the information about the direction of motion.

For feature scaling of values between a range a to b, for a sample feature X, and minimum and maximum values x_{min} and x_{max} respectively, i^{th} value x_i is normalized as,

$$x_{i\_norm} = a + \frac{(b-a)(x_i - x_{min})}{x_{max} - x_{min}}$$

Thus, for scaling flex sensor values between 0 and 1,

$$x_{i\_norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

For scaling accelerometer and gyroscope values between -1 and 1,

$$x_{i\_norm} = \frac{2(x_i - x_{min})}{x_{max} - x_{min}} - 1$$

D. Neural Network Model

1) **Training**: The dataset was divided into 2 parts. The first part comprising of 90% of the dataset was used as training data. The remaining 10% of the dataset was used as validation data and test data. The training was carried out for 30 epochs with a batch size of 64 data points per each step in an epoch.

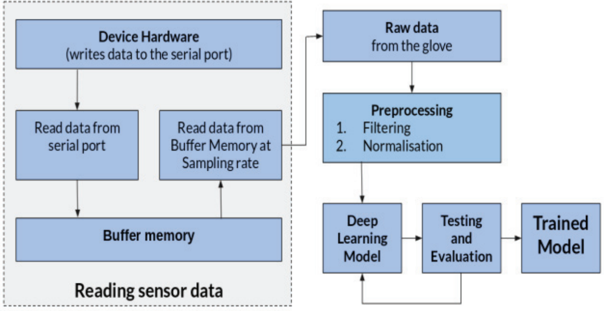

Fig. 4. Software modules

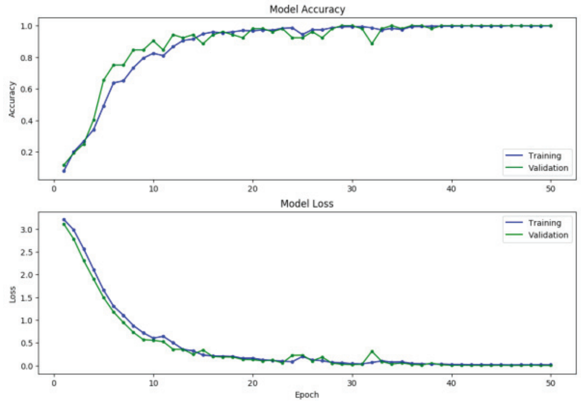

Fig. 5. Loss-accuracy curves

To determine if overfitting or underfitting is occurring, we used loss and accuracy curves. The plots consist of training and validation loss, and training and validation accuracy. Overfitting and underfitting of model is determined as follows –

- If validation loss is higher than the training loss, or if validation loss starts to increase, this indicates that the model is overfitting
- If validation loss is much lower than the training loss, then the model is underfitting
- A model with good fit will have approximately similar training and validation losses

Typically validation loss should be similar to but slightly higher than training loss. As long as validation loss was lower than or even equal to training loss, training was continued further. That is, the number of epochs was increased. The number of epochs was set to that point where the validation started to increase.

The loss and accuracy curve used for determining the number of epochs and dropout values is shown in Fig. 5. After 30 epochs, the validation loss starts to increase. Thus, 30 epochs were considered to be suitable for training a model with a good fit.

2) **Network Architecture**: The deep learning model comprises of LSTM layer, Dense layers, dropout layer and softmax classification layer.

- **LSTM Layer**: The LSTM layer having 50 LSTM units acts as the input layer for the network. This input layer receives a 3D sensor data as input which is an array of 2D data of dimensions 50x11.
- **Hidden Layer**: The network has a single hidden layer which is implemented as a densely connected layer having 100 nodes. The Dense layer performs a dot product of the input value and weight matrix of the layer, followed by application of a linear activation function to it.
- **Dropout Layer**: Following the hidden layer, a dropout layer is added with a dropout value of 40%.
- **Softmax Classifier:** The last layer is a softmax classification layer with 26 nodes, each corresponding to a class in the input dataset. The softmax layer gives the class probabilities corresponding to each label, and the class label corresponding to the highest class probability is considered to be the predicted output of the model.

V. Results

A. Accuracy

On training the model up to 30 epochs, as observed from the loss-accuracy curves in Fig. 5, a near 100% accuracy is achieved for training as well as validation. For evaluation of the trained model, the model was tested on the test data set having 104 samples (10% of dataset). The saved model was loaded from disk, and based on its predictions for the 104 test samples, an accuracy of 98% and a loss value of 0.076 was obtained.

Confusion Matrix: A confusion matrix was plotted to summarize the predictions of our model on test data. In the confusion matrix, each row depicts the values of predicted probabilities for a given actual class, whereas each column depicts the probabilities of different classes being predicted as a particular class. As observed from Fig. 6, presence of all the dark cells in the diagonal indicates that the model predicts most of the class labels correctly and with a high level of confidence. The light-coloured cells in each row indicate the classes with which the model is confusing the class corresponding to that row.

As observed from the confusion matrix, one of the major causes of confusion is the lack of knowledge about the relative position of the glove with respect to the body. Thus a sign which involves motion of the hand in a vertical plane with respect to the body will be interpreted in the same way as the sign involving the motion in horizontal plane with respect to the body if finger orientations and hand motion are similar.

B. Affordability

The designed glove uses low-cost hardware compared to vision based gesture sensing systems which require cameras. It also uses a single right handed glove to predict signs that require two hands and is still able to achieve a high level of accuracy. This is because most of the signs in ISL have sufficient variation in the orientation and motion of the right hand, which is enough for differentiating between various signs.

C. Portability

The glove is capable of transmitting data wirelessly via Bluetooth and operates on a rechargeable lightweight Lithium Ion battery. A mobile device can be used for processing the predictions and for audio output, thus making the glove portable. Use of mobile device avoids the need for extra processing equipment as most users carry a smartphone these days thus adding to its affordability.

D. Usability

The device once trained can be used by anyone whose hand can fit into the glove. The user can also train the glove on custom signs i.e. the user can assign simple signs to commonly used nouns, which are not a part of ISL. This way it can be customized to a user's requirement.

For some ISL signs, the facial expressions, and motion of body parts other than the hand, such as nodding of head, are essential for correct interpretation of the sign. As the glove only takes the hand gesture data into consideration, it cannot be used to classify gestures which involve motion of other body parts.

VI. Conclusion

A gesture sensing glove capable of real-time translation of Indian Sign Language to speech was designed and implemented. The glove uses a combination of flex sensors, gyroscopes and accelerometers to read data corresponding to static and dynamic hand gestures. Bluetooth protocols are used for transfer of data from the glove to the processing device. The gesture data received is then classified into corresponding text and speech outputs using LSTM networks.

The model was trained for 26 words in the Indian Sign Language with 40 samples for each word. The trained model could achieve an accuracy of 98% on the test data. This accuracy can further be improved by training the glove on a larger and more diverse group of people.

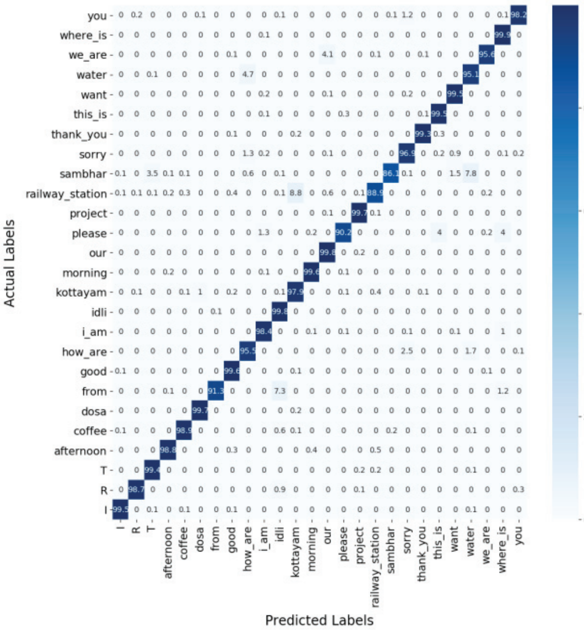

Fig. 6. Confusion matrix for the trained model

The designed glove uses low-cost hardware, making it affordable for the users. Since the device has wireless capabilities, it is portable and can be used with mobile devices for translation of gestures into text and audio.

Although it is specifically targeted at ISL, if trained by the user, it can be used for other sign languages or for general gesture recognition applications too. Further work can be done on finding ways to incorporate the information about the relative position and orientation of hand with respect to the body, to be able to better classify signs.

References

[1] Disabled Persons in India A Statistical Profile, 2016J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] N. Deo, A. Rangesh and M. Trivedi, “In-vehicle Hand Gesture Recognition using Hidden Markov models,” 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 2016, pp. 2179-2184.

[3] S. P. More and A. Sattar, “Hand gesture recognition system using image processing,” 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016, pp. 671-675.

[4] Lipton, Z.C., Berkowitz, J. and Elkan, C., 2015. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.

[5] Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), pp.157-166.

[6] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory.Neural computation, 9(8), pp.1735-1780.

[7] K. K. Dutta, Satheesh Kumar Raju K, Anil Kumar G S and Sunny Arokia Swamy B, “Double handed Indian Sign Language to speech and text,” 2015 Third International Conference on Image Information Processing (ICIIP), Waknaghat, 2015, pp. 374-377.

[8] J. Singha and K. Das, Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique,(IJACSA) International Journal of Advanced Computer Science and Applications, Volume 4, no. 2 , pp. 188-195,July 2013.

[9] S. Y. Heera, M. K. Murthy, V. S. Sravanti and S. Salvi, “Talking hands” An Indian sign language to speech translating gloves,” 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, 2017, pp. 746-751.

[10] K. S. Abhishek, L. C. K. Qubeley and D. Ho, ”Glove-based hand gesture recognition sign language translator using capacitive touch sensor,” 2016 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC), Hong Kong, 2016, pp. 334-337.

[11] K. A. Bhaskaran, A. G. Nair, K. D. Ram, K. Ananthanarayanan and H. R. N. Vardhan, “Smart gloves for hand gesture recognition: Sign language to speech conversion system,” 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA), Kollam, 2016, pp. 1-6.