Realization and Implemenation of Communication for Hearing Impaired and Inarticulate People

Dr. Sajjad ur Rehman
Electrical Engineering Department
Namal University Mianwali
Mianwali, Pakistan
sajjad.rehman@namal.edu.pk

Dr. Majid Ali
Electrical Engineering Department
Namal University Mianwali
Mianwali, Pakistan
majid.ali@namal.edu.pk

Dr. Ihsan Ullah
Engineering Informatics Department
National University of Ireland
Galway, Ireland
ihsan.ullah@nuigalway.ie

Tanzila Iram
Electrical Engineering Department
Namal University Mianwali
Mianwali, Pakistan
tanzila2-18@namal.edu.pk

Muhammad Faizan Ikram
Electrical Engiering Department
Namal University Mianwali
Mianwali, Pakistan
faizan2018@namal.edu.pk

Abstract— **Natural language processing is one of the most growing fields of research. It deals with human-computer interaction and natural languages. Humans share ideas and thoughts with people around them using speech and hearing abilities. But this is not the case for hearing impaired and inarticulate people. Through sign recognition, communication is possible with hearing impaired and inarticulate people.**

In this paper, we have proposed a mobile application-based system for recognizing sign language and converting it into text and speech, which provides smooth communication between the deaf-mute community and normal people, thereby reducing the communication barrier between them. Our proposed system will consist of a mobile application that would be used for communication. The project will mainly consist of two parts, i.e., sign-to-speech conversion and mobile application development. In sign-to-speech conversion, the sign will be recognized using a mobile camera with the help of computer vision, image processing and machine learning algorithms. The recognized sign will be converted into text, and text will be used to generate speech. With the incorporation of the mobile app, this system will ease people with disabilities to communicate with non-disabled people and reduce the communication barrier between them.

Keywords— ***Machine learning, computer vision, mobile application development, image processing***

I. Introduction

Natural Language Processing (NLP) is an emerging field that helps in conveying information and meaning with semantic cues such as words, signs, or images. Sign Language is a very convenient way for hearing impaired and inarticulate people. But for other world, communication with hearing impaired and inarticulate people is difficult because a normal person cannot understand their sign language in normal circumstances. This difference in society can be minimized if we can program a machine in such a way that it translates sign language into text or speech. This project utilized computer vision and image processing techniques to develop a system that can convert sign language into textual and audio form with a mobile application interface, thereby, reducing the communication barrier between specially-abled and non-disabled people. It is a reliable alternate mode of communication that does not need a voice and is critical for individuals suffering from hearing and speech disabilities to improve their quality of life. This project aims to meet UN's SDG 8.5, 10.2 and 10.3. It aims to increase the productive employment of all persons including persons with disabilities. The second purpose of this project is to reduce inequalities and communication barriers in society.

The sign language of hearing-impaired and inarticulate people will be recorded in the data collection phase in the form of videos consisting of words and sentences that are used in daily life and emergencies. Since video is dynamic in nature and it is a collection of images that are referred to as frames, therefore it is not much different from an image classification problem. These frames will be processed in sequence and will be used to extract features with the help of deep learning algorithms, and then classify those frames based on these extracted features.

A video consists of an ordered sequence of frames that contain special information and a whole sequence of the video contains main information. So a video classification model with a Convolutional Neural Network – Recurrent Neural Network (CNN-RNN) architecture will be trained on the existing dataset and deployed on the mobile application. This machine learning model will be used to translate the sign language of hearing impaired and inarticulate people into different words and sentences which belong to 29 different classes. This report is structured in different chapters which will describe the complete methodology and design process of the project including a literature review.

II. Literature Survey

Sign language recognition is an important field of research and a lot of related work has been done in this area for different fields. Sign language recognition was done by vision-based approaches, data glove-based approaches, machine learning methods like Convolutional Neural Network, Artificial Neural Network, Fuzzy Logic, PCA, LDA, Genetic Algorithm and SVM, etc. These techniques utilized different approaches consisting of Hand segmentation, Facial expression, Feature extraction, or Gesture recognition. Many researchers have utilized these techniques for sign language recognition.

P. D. Rosero-Montalvo proposed a method of sign language recognition based on intelligent gloves using machine learning techniques in which he used hardware-based gloves [1]. The accuracy of the system was 85% and it utilized hardware-based flex sensors to record the sign gestures which resulted in extra cost and the user have to carry additional equipment with him.

Mahesh Kumar NB designed a web application-based system to convert sign language into text using Linear Discriminant Analysis (LDA) approach [2]. In this system, the researcher used images of hand sign and extracted features from them to compare with the existing database to classify the sign and generate respective text. Only hand gestures were

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

classified in this system and signs related to other body motions were ignored.

M. Mirzaei proposed a vision-based method using Augmented Reality (AR) and Automatic Speech Recognition (ASR) technologies that combine audio, video and facial expression of the signer to translate their sign language [3]. In this system, facial expressions of the subject were recorded and the use of AR technology helped the narrator and signer communicate with ease. But the system was expensive in terms of hardware resources and computational cost. A maker-free, visual Indian Sign Language (ISL) recognition system was developed by Kanchan Dabre in which a machine learning model for sign language interpretation based on webcam images was used to extract features of hand images and classified with Haar Cascade Classifier [4]. This system also used hand gesture images only and not the facial expression or full-body motions. Also, the web application based system made it hard and non-versatile to use in daily life activities.

Ankit Ojha presented Convolutional Neural Network (CNN) architecture to recognize the American Sign Language (ASL) for numbers and alphabet [5]. The proposed method used OpenCV for real-time hand pose detection and augmentation and fed these images to the CNN model and trained it to recognize the gesture and convert them to textual content and speech. This system was developed to classify the American sign alphabets only and therefore the utility of the system was limited. A large-scale video classification model based on a spatial-temporal network and a connected multiresolution architecture was proposed by Andrej Karpathy in which video-based approach was used to classify 1M YouTube videos [6]. Specifically, this proposed architecture was used to recognize human activities and was expensive in terms of computation and performance.

In this project, we will also use vision-based approaches like image processing and machine learning to develop a sign language recognition system with easy to use mobile application interface. We will be using video classification to extract frames of video and process them to extract meaningful features. Unlike image classification, videos cannot be easily processed due to their temporal and time-series property. Videos have a fixed-sized architecture that has both spatial and temporal information. Specifically, we will be using a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) architecture for spatial features and temporal features respectively. This architecture is consisting of CNN, GRU and LSTM layers which is a hybrid architecture also known as CNN-RNN. We will also experiment other large-scale video classification models like ConvLSTM which are also hybrid architecture of the same kind. At the end, we will be comparing the results of our developed system with other state of the art models to evaluate the performance and efficiency.

III. Methodology

The Sign Language Recognition System works in multiple stages as shown in Figure1. The methodology includes data collection, mathematical models, algorithm, server connection and android application development.

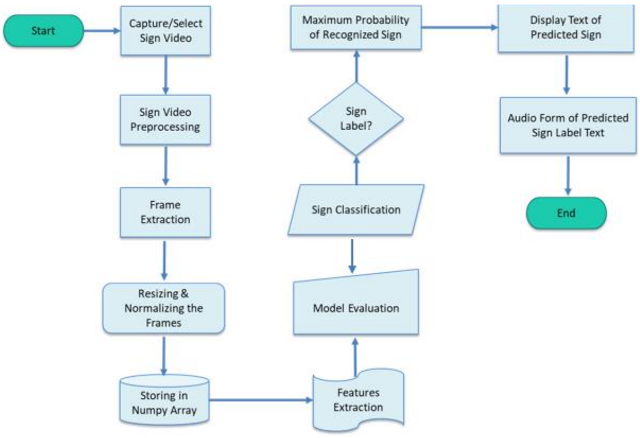

Start
Capture/Select Sign Video
Sign Video Preprocessing
Frame Extraction
Resizing & Normalizing the Frames
Storing in Numpy Array
Features Extraction
Model Evaluation
Sign Classification
Sign Label?
Maximum Probability of Recognized Sign
Display Text of Predicted Sign
Audio Form of Predicted Sign Label Text
End

Fig. 1. Methodology

A. Data Collection

The first step is data collection. We defined the area of interest consisting of categories that we want to address in our project and train the classification model on it. These include words and sentences that are normally used in daily life activities and emergencies. For data collection, the local government-owned school named *Government School for Deaf and Dumb Mianwali* was approached to allow the video recording of deaf-mute students for project purposes. We made the setup and kept normal background and lighting conditions to record videos. The subjects were asked to perform signs and gestures for the tasks listed in table 1. We recorded video samples for each sign and processed them for better use. The meta data of recorded dataset is given below:

TABLE 1. Metadata Of Recorded Dataset

No. of Subjects	**30**
No. of Categories/Classes	29
No. of Samples per Category/Class	70-80
Total no. of Samples	**1124**

B. Mathematical Model

We have used CNN-RNN architecture model in this paper. These are deep learning algorithms that are very powerful and mostly used in solving computer vision and image or video classification problems. These models are sequential and help to execute functional mathematical operations on dataset and also maintain the long-range sequences in time-series input data. Using the said architecture, we have utilized different models for video processing and recognition. We have utilized following models for processing the spatial information of video frames and maintaining the sequence of the whole sign video with temporal information:

- Convolutional Neural Network (CNN) Detector.
- Recurrent Neural Network (RNN).
- Gated Recurrent Unit (GRU).
- Long Short Term Memory (LSTM).
- Convolutional Long Short Term Memory (LSTM).

C. *Algorithm*

1) *Video Processing and Frames Extraction:*

The main challenge of building a video classifier is finding out a way to feed the videos to a neural network. There are several methods available for this and we followed the traditional approach. Since video is an ordered sequence of frames, we could just extract the frames in order and put them in a 3D sensor so that they can be fed to the neural network. We had different numbers of frames in each video since the number of frames differ from video to video. So we saved them at a fixed interval until the maximum frame was reached. We have performed the following steps to process the videos:

a) Capture the frames from the video using the OpenCV library

b) Extract frames from the whole video until the maximum frame is reached.

c) If the video's frame count is less than the maximum frame count, then we need to copy previous frames or use zero padding to match the count length.

2) *Feature Extraction*

- **InceptionV3:** InceptionV3 which is also a very popular and useful model for extracting features from the image dataset. We passed the frames dataset to this pre-trained InceptionV3 model and extracted features from all frames of all training and testing videos and stored them in a feature vector. This vector is then passed to the sequential model which is based on CNN-RNN architecture. We trained this model and checked performance and accuracy on training and testing datasets. The performance of this model was satisfactory at this level where we do not have an adequate amount of data. We assigned labels on testing videos and calculated probabilities. The label with the highest probability is then predicted along with other classes after that. The results are recorded in the Results part.

D. *Server Configuration*

Servers are used for different purposes like model deployment, processing and computing power etc. This is the domain of network infrastructure in which we deploy machine learning models and rely on CPU/GPU/TPU-enabled servers, flash storage, and other compute infrastructure. This process is done as machine learning models are very processor and storage intensive. However, the storage systems and servers must be fed data that traverses a network. For the connection and integration of frontend and backend we can use the following approaches:

1. WEB SERVER (PUBLIC NETWORK)
2. LOCAL SERVER (PRIVATE NETWORK)

In our project, we have utilized Local server networking. For this purpose, there are different Python web frameworks that provide local server networking like Flask, Django etc. We have used Flask in this project to establish a backend server connection. Machine Learning model is deployed on the Flask server, that processes the video being uploaded from the Android Application. In order to put Flask localhost network online, we used Negrok, so that we can connect our mobile device with the online network service. Ngrok is an online service that helps to connect the apps with Ngrok's network edge. This is a great platform to put localhost on the internet securely. It can be connected to any system regardless of network or location.

E. *Android Application*

In order to provide an easy user interface to the user, Android application is developed. For this we used the Android studio to design front end and backend is connected with the flask server which in turn gives the response of machine learning model. Android application provides an easy user interface. By this the user can upload video from the memory or can record video on the spot. The selected/uploaded video is then sent to the flask server and then the flask server gives the result of ML model. The result is actually the model prediction that is sent back to the android application screen. Android application has another feature as well i.e. text to speech conversion. The predicted result is displayed on the app screen and then by clicking on the "Text to Speech" button, you can get the audio form of predicted text.

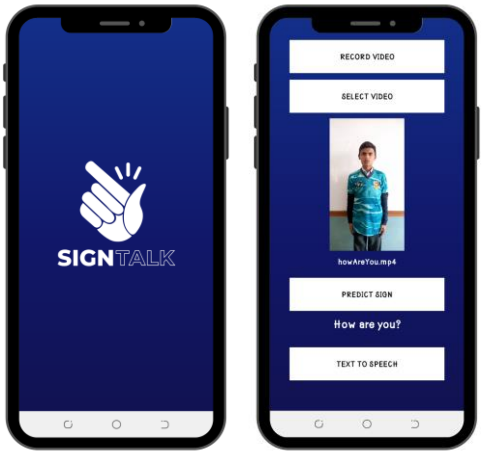


Fig. 2. Android App Interface

IV. BLOCK DIAGRAM

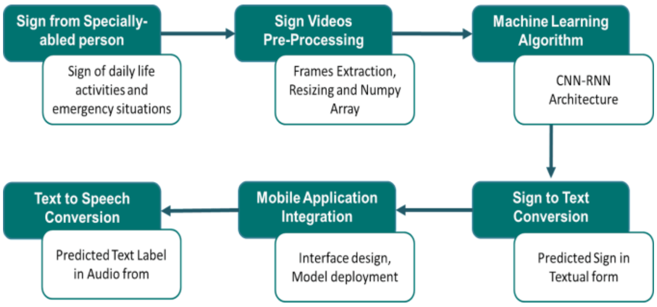

Sign from Specially-abled person
Sign of daily life activities and emergency situations
Sign Videos Pre-Processing
Frames Extraction, Resizing and Numpy Array
Machine Learning Algorithm
CNN-RNN Architecture
Sign to Text Conversion
Predicted Sign in Textual form
Mobile Application Integration
Interface design, Model deployment
Text to Speech Conversion
Predicted Text Label in Audio from

Fig. 3. Block Diagram

V. EXPERIMENTAL RESULTS

We have used five different classes for training and testing. These classes consist of videos dataset from daily life activities and emergency situations. Five classes are shown below in the TABLE 2.

A. *CNN-RNN Model Results*

TABLE 2. SENTENCES USED FOR TRAINING SET

Sr. No	Word/Sentences used for Training & Testing
1	How are you?
2	I cannot speak
3	I don't understand
4	I am a student
5	Call the ambulance
6	Invalid Class

On average each class consists of 80 videos. These videos are passed to machine learning model to perform the training. Model training results are shown below.

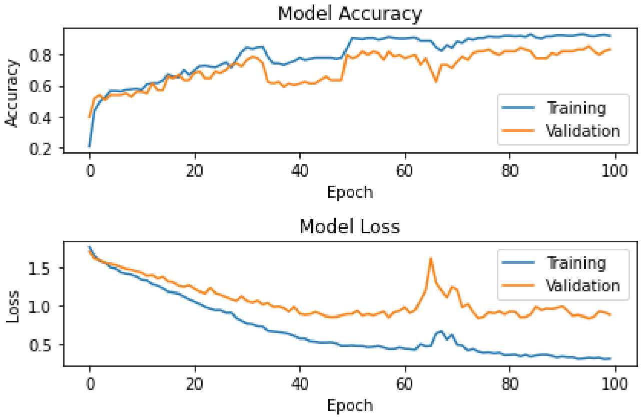
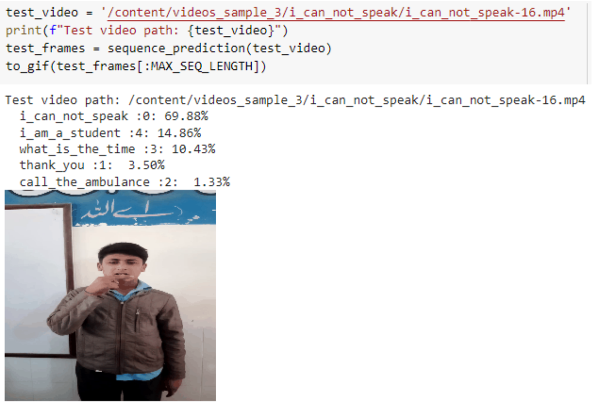


Fig. 4. Model loss & Accuracy

B. *Test Cases*

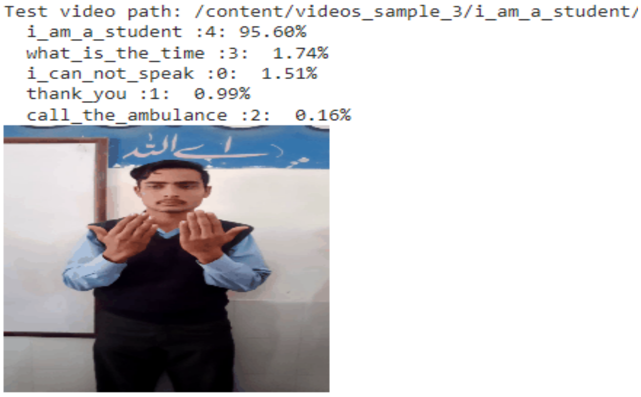
1) *Test Run (Backend)*



```
test_video = '/content/videos_sample_3/i_can_not_speak/i_can_not_speak-16.mp4'
print(f"Test video path: {test_video}")
test_frames = sequence_prediction(test_video)
to_gif(test_frames[:MAX_SEQ_LENGTH])
```

```
Test video path: /content/videos_sample_3/i_can_not_speak/i_can_not_speak-16.mp4
  i_can_not_speak :0: 69.88%
  i_am_a_student :4: 14.86%
  what_is_the_time :3: 10.43%
  thank_you :1:  3.50%
  call_the_ambulance :2:  1.33%
```

Fig. 5. Test video prediction for label "i_can_not_speak"

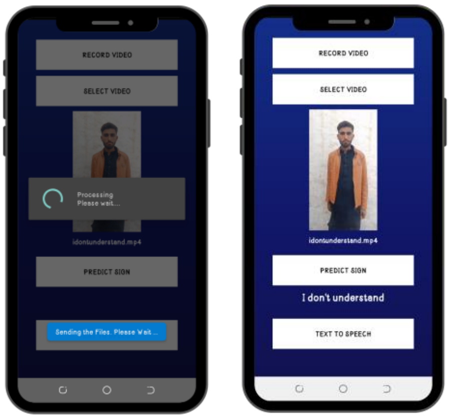


```
Test video path: /content/videos_sample_3/i_am_a_student/
  i_am_a_student :4: 95.60%
  what_is_the_time :3:  1.74%
  i_can_not_speak :0:  1.51%
  thank_you :1:  0.99%
  call_the_ambulance :2:  0.16%
```

Fig. 6. Test video predictions for label "i_am_a_student"

After performing the model training, the model is then integrated with mobile application. Android app is connected to the server on which ML is deployed. Video is passed from mobile app to the server to give the prediction from model and then the predicted result is sent back to the app that shows the result on app screen. Audio form of predicted text can also be taken using Text to Speech button click. Android app results are shown in Figure:

2) *Test Run (Frontend)*



Processing
Please wait....

Sending the Files. Please Wait ...

I don't understand

Fig. 7. Test video prediction for label "I don't understand

VI. CONCLUSION

The prediction using InceptionV3 and GRU based CNN-RNN architecture for Pakistan Sign Language recognition is presented in this paper which got 91% accuracy. We have introduced an automatic and intelligent system based on mobile application and deep learning that is capable of converting sign language into text and speech with the help of a recorded sign video. In comparison to other methods like support vector machines, hidden Markov models, backpropagation algorithms and k-nearest neighbor methods, this method is much faster. In addition, the convergence rate is also faster, increasing the speed and accuracy of sign language interpretation within mobile application for real-time use. The most accurate results and better performance weighted model was used for deployment on Flask server and then integrated with the mobile application. This android application is used to record or select a video and send it to the server for prediction. The server used here is a local network developed with the help of Flask framework. This server establishes a successful connection of mobile applications with the ML model.

The training and testing of the algorithm is a continuous process and we can increase the dataset for better accuracy of the model so that it has minimum loss when deployed on the mobile application. We have trained the model for a smaller number of classes with more datasets. This means if we increase the dataset per class, we would get much better accuracy of the model and predictions will be done correctly. To this end, we are able to embed this whole system with a mobile application that can be easily used by the hearing impaired and inarticulate people to convey their message to the ones who don't have any disability. Moreover, people with no hearing and speaking disabilities are also able to use this mobile application to understand the sign language of disabled people.

VII. Future scope

The introduction of facial expressions and whole body gestures in the identification of signs made by deaf-mute people will significantly improve this Sign Language Recognition System.

For a local sign language, a big data collection for sign language recognition system is a challenging task since it requires learning every sign under every aspect of the moment, hand shape, size, color, and background etc. Having a good number of dataset can help the model to perform even better to give us the correct results. After pre-processing the dataset, the next step would be to implement the pre-trained CNN-RNN model. The model's hyper parameters can be changed to get good accuracy on training, validation, and testing. As a result, we would be able to correctly predict the unseen data from the trained model. Hence by this, we would generalize our model for more classes as well

A versatile and more fast system can be implemented on different mobile platforms for both Android and IOS users.

Acknowledgment

I acknowledge ALLAH Almighty as the most merciful and most beneficent. Without acknowledging the people who contributed to the accomplishment of this paper, we would not be able to fully express our joy and satisfaction. Their constant guidance, support, and encouragement resulted in the success of our project. We were able to finish the paper with Allah's grace and the prayers of our family and teachers. We would like to thank our project supervisor, Dr. Sajjad ur Rehman, and our co-supervisors, Dr. Majid Ali and Dr. Ihsan Ullah Afridi. The experience of working with them was truly extraordinary. As a result of this project, we have acquired a great deal of knowledge that will inspire our future professional endeavors. Throughout this project, their guidance and expertise kept us on track toward success. Finally, we cannot forget the specially-abled students and teachers of *Government School for Special Education Mianwali*, who have been tremendously helpful in data collection and have supported us throughout the project.

References

[1] P. D. Rosero-Montalvo et al., "Sign Language Recognition Based on Intelligent Glove using Machine Learning Techniques," in IEEE Third Ecuador Technical Chapters Meeting (ETCM), 2018, pp. 1-5, doi: 10.1109/ETCM.2018.8580268.

[2] Mahesh Kumar, "Conversion of Sign Language into Text" in International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 7154-7161

[3] Mirzaei, M.R., Ghorshi, S. & Mortazavi, M. "Audio-visual speech recognition techniques in augmented reality environments." Vis Comput 30, 245–257 (2014). doi: 10.1007/s00371-013-0841-1.

[4] Mirzaei, M.R., Ghorshi, S. & Mortazavi, M. "Audio-visual speech recognition techniques in augmented reality environments." Vis Comput 30, 245–257 (2014). doi: 10.1007/s00371-013-0841-1.

[5] K. Dabre and S. Dholay, "Machine learning model for sign` language interpretation using webcam images," in International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA), 2014, pp. 317-321, doi: 10.1109/CSCITA.2014.6839279.

[6] A. P. S. M. A. T. D. D. P. Ankit Ojha, " Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network," in INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT, vol. 8, no. 15, 2020.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725-1732, doi: 10.1109/CVPR.2014