International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Two Way Communication between Deaf and Dumb

Shubhangi Dixit[1], Akshata Dabade[2], Aishwarya Kanetkar[3], Anish Apte[4], Sayali Pisal[5]

[1]Professor(HOD), PVG's College of Engineering and Technology, Pune, Maharashtra, India

[2, 3, 4, 5] Student, PVG's College of Engineering and Technology, Pune, Maharashtra, India

Abstract: ***This paper aims to introduce a mobile application concept for facilitating communication between deaf & dumb and normal people. We propose a system which translates Indian Sign Language into equivalent text and also converts input text into equivalent ISL gestures. For the former, we use a SVM classifier to get the required output. While the latter can be done by image processing and/or animation techniques. With this application we hope to reduce the communication gap between 'deaf & dumb' and normal people by eliminating the need of a human translator.***

Keywords: SVM

1. Introduction

Verbal communication plays an important role in expressing our thoughts and emotions. The inability to hear or speak can cause a great deal of mental anguish.

Although the deaf and dumb are able to communicate using a mixture of hand signals and lip reading, not everyone can understand them. Learning a sign language is as difficult as learning any new language. Also lip reading is a difficult art to master and is not completely reliable even after years of practice.

Furthermore, the sign languages differ not only from country to country but also from region to region. Therefore it becomes a cumbersome task for them to express themselves. A sign language interpreter is required to act as a mediator. However it may happen that the interpreter fails to correctly understand what the person is trying to say or may make some mistakes while trying to convey something.

In this paper we propose a system which will get rid of the requirement of an interpreter and thus eliminate the possibility of error during translation. Also by maintaining an extensive database on the varying sign languages the system will be able to bridge the communication gap with ease. Although we have taken into consideration only the Indian Sign Language (ISL) for now, the system can be suitably scaled up to cover other sign languages from different countries or scaled down to cover only a regional sign language.

What we are proposing is twofold.
1) Translation of input text into gestures
2) Translation of input gestures into text

We will now go ahead and explain the working of the two aforementioned modules, followed by conclusion and references.

2. Module 1(Text to Gestures)

In this module the input will be in the form of text given by the normal person. This module will convert the given input text into its corresponding ISL gestures.

The conversion from text to ISL gestures will be done by following the following steps.
a) Mapping text to image database
b) Converting images to video
c) Display video as ISL gesture output

Note: (*The input can be accepted as speech which will be translated to corresponding text by using Google Speech To Text (STT) API)

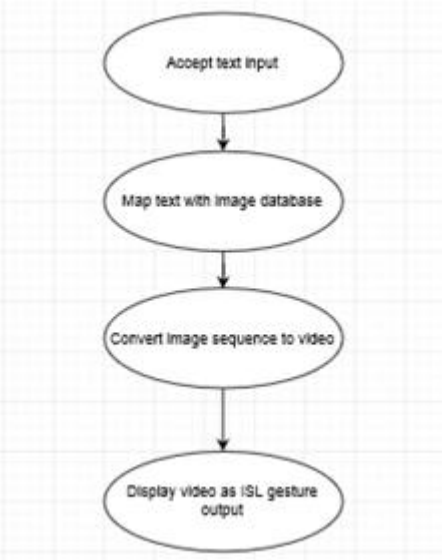

Figure 1: Module I flow

2.1 Map text with image database

In this step the input text has been obtained from normal user. Similar to the first module, we will have with us a database of text and corresponding gestures. This database will be used to translate the input text into required gesture output. The mapping can be done in two ways:
1) Maintain a database of letters and their gesture image. Map each letter of the input text and get gestures for each one.
2) Maintain database for words and phrases too whenever possible. Map a word or phrase completely and directly as and when applicable.

The second method will be less tedious and will reduce time complexity of the actual conversion. However more memory will be required to store a large number of words and

phrases, along with the whole alphabet. But the advantage of a highly optimized time complexity will surely outweigh the disadvantage of some extra memory requirement. Thus the text will directly be converted into its corresponding image by mapping it with the database.

2.2 Convert image to video

The obtained images as a result of the first step will actually be in sequence of the input text. These images can be stored sequentially and then be displayed in the form of a video with appropriate frame speed. This video will be nothing other than the required ISL gesture output.

NOTE: (*The output can also be displayed by an animation avatar. The avatar can be created by a variety of 3-D animation tools available.)

3. Module 2(Gestures to Text)

In this module the input will be in the form of a video of a 'Deaf/Dumb' person communicating using the Indian Sign Language. This module will perform the following steps to successfully convert the input gestures into text.

a) SVM training
b) Image Generation
c) Image Pre-processing
d) Testing given image in SVM
e) Display resultant output text

Image processing in Android Studio is somewhat complicated and timeconsuming. So we are proposing a client server architecture where the android application acts as client which is responsible for providing input video. The video will be stored on server side and python will be used for further processing of the video to get the translated text output. A SVM classifier will be pre-trained in the python code and will be used for image translation.

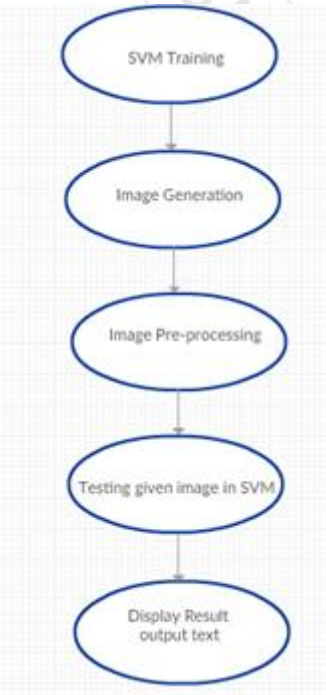

Figure 2: Module II flow

3.1 SVM training

A SVM classifier will be trained with images and their translated text. The images (which will be in an array form) will act as features while the text will be the labels. We propose that the SVM should be trained as thoroughly as possible to minimize the scope for error. This means that a minimum of 10 images should be trained for each label. Here the labels will be a-z characters and common words from ISL.

3.2 Image Generation

Here the video which has been taken as an input by the android application will be sent to the PHP server; where it will be converted into images using Python. It is difficult to do the actual mapping while working with videos and so the conversion needs to be done. Videos are nothing but a series of images displayed in a sequence at some constant speed. So it is relatively easy to convert the video frames into a series of sequential images.

3.3 Image Pre-Processing

Before mapping the images with the database it is necessary to process the images in order to improve their quality and thus better the chances of a correct mapping. Pre-Processing will include noise removal and image enhancement. We also recommend converting the images to greyscale to reduce complexity. The SVM should also be trained with greyscales images.

3.4 Testing given image in SVM

Now on reaching this step the input video has been successfully converted into frames. Also the images have been greyscaled. Now the images have to be fed as input to the pre-trained SVM classifier. The classifier will compare the input image with its library of trained images. When it hits a match it will return the corresponding label of the matched image. This process will continue till all the input images have been mapped. The resultant text obtained by matching all images will be the required output text.

3.5 Display resultant text as output

Store the resultant text obtained by translating the input images using SVM into a file. The client android application will get the file from server and then extract the text. This text will be displayed to the user. This will be the translated text of the input ISL gestures provided by user.

NOTE: (*The resultant text output can be converted into speech by using Google Text To Speech (TTS) API)

4. Conclusion

Some innovations have already been made to convert gestures into text, the most prominent being the data glove. But the gloves only facilitate one way communication and are not easily portable in the logical sense. But the system we are proposing will not only act as a two way

communicator but it will also be portable and easy to use.

We hope that the proposed system will help in the long run to not only reduce but completely eliminate the communication gap between 'deaf & dumb' and normal people. After all, a communication barrier must not hinder a person's chances of a good education or a job.

We recommend performing image processing and SVM training & testing using Python language because it is much easier than doing the same in Android Studio. Moreover using a client-server scenario will improve the android application's throughput.

References

[1] Prof. Prashant G. Ahire, Kshitija B. Tilekar, Tejaswini A. Jawake, Pramod B. Warale- "Two Way Communicator Between Deaf And Dumb People And Normal People" ,Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on Year: 2015
[2] Introduction to Video And Image Processing Building Real Systems And Applications.'http://www.springer.com/978-1-4471-2502-0'
[3] "Two Way Communicator Between Deaf and Dumb People And Normal People" Prof. Prashant G. Ahire, Kshitija B. Tilekar, Tejaswini A.Jawake, Pramod B. Warale
[4] Ibrahim Patel, Dr. Y. SrinivasaRao-"AUTOMATED SPEECH RECOGNITION APPROACH TO CONTINUOUS CUE-SYMBOLS GENERATION"Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference onYear: 2010
[5] SachinBhat, Amruthesh M, Ashik, Chidanand Das, Sujith-"Translating Indian Sign Language to text and voice messages" Machine Vision and Image Processing (MVIP), 2012 International Conference onYear: 2012