Two Way Communication between Deaf & Dumb

Akshata Dabade¹, Anish Apte², Aishwarya Kanetkar³, Sayali Pisal⁴

Department of Computer Science & Information Technology, PVG’s College of Engineering and Technology, Savitribai Phule Pune University, 44, Vidyannagari, Parvati, Pune-411009, Maharashtra, India.

Abstract: This paper aims to introduce an 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, two algorithms are used, namely Connected Component and PCA; 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: Connected Component Algorithm, PCA

I. 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 (sign language) 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 gestures into text
2. Translation of input text into gestures

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

II. Module I (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. Image Generation
b. Image Pre-processing
c. Finding Region of Interest using Connected Component Algorithm
d. Actual translation of image into text using PCA algorithm

![Fig. 1 Module I flow]
A. Image Generation :

Here the video which has been taken as an input to the system will be converted into images. 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.

B. 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.

C. Finding ROI :

The images which are obtained as a result of converting the videos into frames will include things which are not necessary and are not used to communicate the ISL gestures. Here we only need to focus on the hand movement of the deaf/dumb person as ISL consists of only hand gestures. To do this we will need to find the region of interest in the image. This can be done by implementing the connected component algorithm which is explained below.

Connected Component:

Connected components algorithm scans an image and groups its pixels into components based on pixel connectivity, i.e. all pixels in connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labelled with a grey level or a colour (colour labelling) according to the component it was assigned to.

Extracting and labelling of various disjoint and connected components in an image is central to many automated image analysis applications.

Algorithm:

Connected Component Matrix is initialized to size of Image Matrix.

1. A Marker is initialized and incremented for every detected object in the image.

2. A counter is initialized to count the number of objects.

3. A row-major scan is started for the entire image.

4. If an object pixel is detected, then following steps are repeated till (Index! =0)

a. Set the corresponding pixel to 0 in Image.

b. A vector (Index) is updated with all then neighbouring pixels of the currently set pixels.

c. Unique pixels are retained and already marked pixels are removed.

d. Set the pixels indicated by Index to 1 in the Connected Component Matrix.

5. Increment the Marker for another object in the Image. In this way this algorithm separate foreground from background and makes a pattern of hand gestures. According to stored pattern corresponding action (Left, Right, Forward an and Backward) is performed by hardware.

D. Image to text conversion :

Following the first three steps we have successfully converted the input video to good quality images and have identified the region of interest in those images.

Now it is time to actually convert the images into equivalent text. For this it is necessary to maintain a database of images and their corresponding text translation. This database will be used to map the input images and thus will facilitate the actual conversion.

The mapping of images can be done by implementing the PCA algorithm which is explained below [4].

PCA:

Principal component analysis (PCA) has been called one of the most valuable results from applied linear algebra. PCA is used abundantly in all forms of analysis - from neuroscience to computer graphics - because it is a simple, non-parametric method of extracting relevant information from confusing data sets. With minimal additional effort PCA provides a roadmap
for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified dynamics that often underlie it.

Principal component analysis (PCA) is a technique used to emphasize variation and bring out strong patterns in a dataset. It's often used to make data easy to explore and visualize. PCA’s mathematical module consists of two phases; namely training phase and testing phase.

**Training Phase:**

1. Each gesture in the database is represented as a column in a matrix A. The values in each of these columns represent the pixels of the gesture image and range from 0 to 255 for an 8-bit grayscale gesture image ∑:

   \[
   A = \begin{bmatrix}
   a_{11} & \ldots & a_{1n} \\
   a_{m1} & \ldots & a_{mn}
   \end{bmatrix}
   \]

   Where, m = Size of image
   n = Number of Gesture Images.

2. Average of the matrix A is calculated to normalize the matrix A. The average of matrix A is a column vector in which every element is the average of every gesture pixel values respectively.

   \[
   \text{Avg} = (x_1, x_2, \ldots, x_m)
   \]

   Where,

   \[
   i = 1, 2, \ldots, m
   \]

3. Next, the matrix is normalized by subtracting each column of Avg matrix from each column of matrix A

   \[
   A = A - \text{Avg}
   \]

4. Then compute the covariance matrix of A, which is \( A * A^T \). It reduces the size of the covariance matrix and calculated as:

   \[
   L = A * A^T
   \]

5. Next step is to obtain the eigenvectors of original matrix thus we need to calculate the eigenvectors of the covariance matrix L, let us say eigenvectors of the covariance matrix are V, with size of V is same as L.

6. Now calculate the eigenvectors of the original matrix after the calculation of V as follows:

   \[
   U = A^T * V
   \]

**Recognition Phase:**

1. We represent the test gesture as a column vector:

   \[
   r = \begin{bmatrix}
   r_1 \\
   r_m
   \end{bmatrix}
   \]

2. The target gesture is then normalized:

   \[
   \text{r} = \begin{bmatrix}
   r_1 & \ldots & m_1 \\
   r_m & \ldots & r_m
   \end{bmatrix}
   \]

3. Next, calculate the projection of test gesture to project the gesture on gesture space by the equation given below [9]:

   \[
   \text{Alpha} = U^T * r
   \]

4. We then find the Euclidean distance between the targets projection and each of the projections in the database by the equation given below:

   \[
   \text{ED} = \sqrt{(\text{Alpha}(1,j))(\text{Alpha}(i,j))^2}
   \]

   \[
   j=1
   \]

   \[
   i=1, 2, \ldots, n
   \]

   \[
   m = \text{total number of pixels in a gesture}
   \]

   \[
   n = \text{number of gestures in the database}
   \]

5. Next, we decide which gesture is recognized by selecting minimum Euclidean distance from the Euclidean distance vector “ED”.
III. Module II (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)

B. Convert image to video :

[5] 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.)

IV. 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 (since it’s basically a mobile application).

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.
References


