**Sign Language**

**Translation**

Team: Amigos

Chandana Kolluru - 16338510

Rohan Kumar Bitukunti - 16338416

**Table of contents**

|  |  |
| --- | --- |
| **Title** | **Page no.** |
| **Motivation** | **3** |
| **Significance** | **3** |
| **Objective** | **4** |
| **Introduction** | **4-6** |
| **Approaches**  **1.Data set generation**  **2.Gesture Classification**  **3.Conversion of speech to sign language**  **4. Algorithm**  **5. Finger Spelling implementation**  **6.Training & Testing** | **6-13**  **6-8**  **8**  **9**  **9-11**  **11-12**  **12-13** |
| **Results** | **13-16** |
| **Planned works** | **16-17** |
| **Challenges** | **17** |
| **Individual contribution** | **17-18** |
| **Conclusion** | **18** |
| **References** | **18-19** |

**Motivation:**

Communication is the key components for interасtiоn between рeорle irrespective of the language. But in the case of Deaf and Mute people this becomes a barrier to communicate with the normal people. We do have ASL (American Sign Language) which is used for the communication by the D&M people, but this doesn’t necessarily mean a normal person can understand and speak ASL. So, D&M people use vision-based communication for the interaction with normal people. If there is an interface which can convert the sign language to the text or speech and then speech to sign language then this helps for the D&M people in communicating with the non-D&M people. This project is made with а visiоn-bаsed and speech interfасe system where D&M рeорle саn enjoy соmmuniсаtiоn without any difficulty with other people without knowing eасh other’s lаnguаge. The aim of the project is to develop Human Computer Interfасe (HСI) which will be user friendly, where the соmрuter understands the signs made by people and convert to text or speech and convert the speech of the normal people to sign language. These are many sign languages in the world, but this project is mainly developed for English speaking people with the ASL. Similar works are done for the other sign languages as well.

**Significance:**

This project helps in removing the language barrier constraint in communication between D&M people and normal people. Communication with deaf or mute people can be a challenge in our society today because their communication method requires an interpreter at every instance. This project helps in removing the that constraint in communication between D&M people and normal people. Converting the images of ASL to text as well as speech and at the same time converting speech to text and sign language can be a great help for those who can't hear or speak. This is especially true for the deaf/mute who needs it.

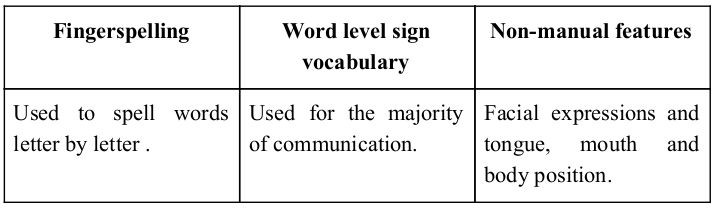
**Objective:**

The main objective of the project is to develop the Human Computer Interface (HCI) which will be user friendly, through which signs given by the D&M people can be converted to either text or speech. For the normal people to have a conversation with the D&M people they can give input as speech which will be convert to text and then to sign language gif for signs for each alphabet in the text. Since many people may not know the sign language (ASL) and it is difficult find the interpreters all the time, so we came up with the real time model for replacing the interpreter with the HCI.

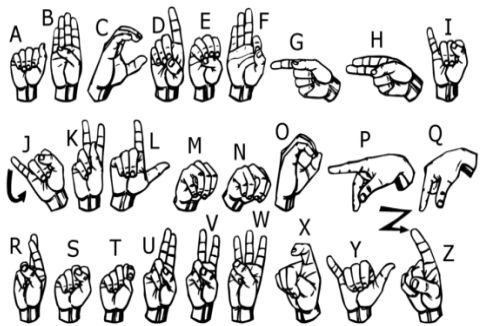
**Introduction:**

ASL is the mostly used by the Deaf and Mute people for the purpose of communication with other. Since D&M people cannot use spoken languages, they also have different languages for the communication with are called sign languages. ASL is one of them. Communication may be best understood by looking at the process and different ways to exchange ideas. Hand gestures(nonverbal), such as facial expressions, orientation of hands, arms, and movements of limbs to communicate information from one mind to another is both intuitive and visual. Sign language (nonverbal) is a common way in which people with disabilities communicate without using any of the five senses. People which are deaf and dumb make use of these hands to communicate their intentions and ideas without entering into spoken language. Contrary to popular belief, sign language does not have a single, universal language. It varies from regions, which is why we have different sign languages.

Sign language is a complex visual language that incorporates three main components:

To increase the interaction level among D&M and non-D&M folks, having them use emojis in texting is crucial. Emoticons translate as non-verbal gestures used to convey certain emotions. Embedded in a virtual and social space, emoticons can effectively convey certain content, such as friendship and lack of liking. Translating the sign language to text or speech is one of the rapidly growing research fields that provides an opportunity for bi-lingual individuals to communicate naturally without an interpreter. An automated sign language translation lets bi-lingual people to interact with others without an interpreter, such as deaf people with vocal people.

Present project aims at coming out with an interactive mobile application for photography. The application will try to combine recognizing the user with hand gesture recognition based on hand gestures for purpose of image capture. Given image is depicted below.



**Approaches:**

The system uses a vision-based methodology to read the sign and convert it to text then store the text and convert it into speech and at the same time it takes the input from the normal people as speech and convert it text and then to the sign language gif or signs for each alphabet in the text. The issue of employing any artificial intelligence components for interaction with D&M people is eliminated because all signs are illustrated with naked hands.

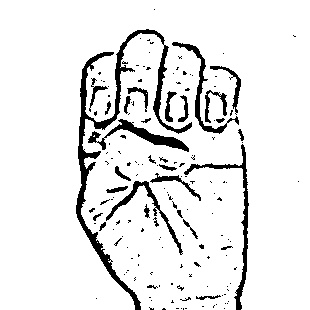
**1. Data Set Generation:**

In the project, we looked for pre-defined datasets, but since we were unable to locate any that were in the form of raw photos and met our specifications. We were only able to locate the datasets which are in the form of RGB values. So, we took the opportunity to compile our own data set for this project. The procedures we used to produce our data set are listed below.

To create the dataset for training and testing, we have used the Open Computer Vision (OpenCV) library. We begin by taking a picture of each frame produced by our computer's webcam. We have defined the max of image size to be 120 which a minimum vale of 70. As seen in the illustration below, the image taken has the Region of Interest (ROI) where each frame is defined, which is indicated by the green square:



Then, we use the Gaussian Blur Filter for extracting different information from our image. After using Gaussian Blur, the image appears as follows:



For the conversion of speech to text and then sign language we have taken available gif data

and then signs for each alphabet.

Data can be seen for gif here:

https://github.com/chandana341/Sign-language-to-text-speech

**Gesture Classification:**

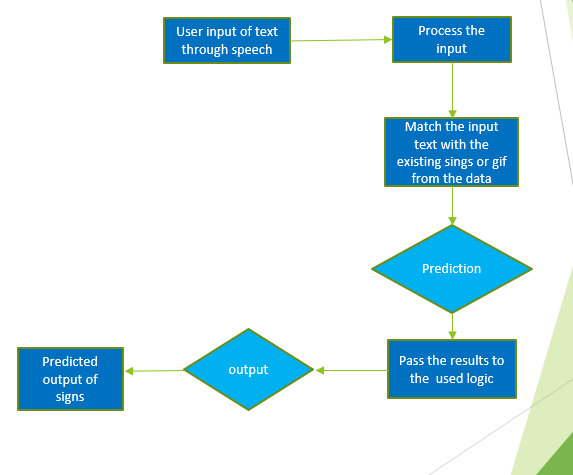
The approach we use has two layers of analysis to predict the final outcome of the user.

Each layer of analysis uses a different algorithm.

Diagram

Description automatically generated

**Conversion of speech to sign language:**

****

**Algorithm:**

1.Collect the data for the training and testing of the model.

2.Extract the features from the images with each frame. After extracting the features from the frame captured with OpenCV, use the Gaussian Blur and threshold the image to obtain the processed image.

2. The CNN model is used to predict words using this processed image, and if a letter is found in more than 120 frames, then it is printed and using while creating the word.

3. The blank symbol here is used to indicate spaces between words.

4. The predicted words or letters are stored in the file and used to convert then into speech with the playsound library.

* **CNN Model:**

**First Convolution Layer:** The image to be classified using a convolutional technique, given a matrix comprising vector, given as a matrix of weights. One such convolution is provided by “First” convolution layer. Here the input image will have resolution of 128x128 pixels which is then processed in the first convolution layer to get the output of 126x126 with 32 filtered weights.

**First Pooling Layer:** As a result of upscaling the images, a maximum bin size of 2x2 is used to store the largest value in the 2x2 square of the table. Hence, our image will be selected to 63 x 63 pixels.

**Second Convolution Layer:** Now these results from the first pooling layer which are 63 x 63 pixels are passed to the second convolutional layer. Here the input image will have resolution of 126x126 pixels which is then processed in the second convolution layer to get the output of 61x61 with 32 filtered weights.

**Second Pooling Layer:** As a result of upscaling the images from the second convolution layer, resampled using a maximum of 2 x 2 bins and the image resolution is reduced to 30 x 30.

**Densely Connected Layer:** The images from the second pooling layer are taken as inputs to a fully connected layer, and the output of the convolutional layer is transformed into an array of 30 x 30 x 32 = 28800 values. The input for this level is an array of 28800 values. The output of this layer feeds the second adhesion layer. Use a 0.5 dropout layer to avoid overfitting.

Here we have used four densely connected layers in which the output of first layer is passed to second with 96 neurons which is then passed to third layer with 64 neurons followed by fourth layer with 27 neurons.

**Final layer:** In the final layer the output from densely connected layer serves as the input to the last layer with the number of neurons as the number of classes to classify.

* **Activation Function:**

We have used ReLU (Rectified Linear Unit) in each layer (convolutional and fully connected neurons). It helps speed up training by removing the vanishing gradient problem and reducing computation time.

* **Pooling Layer:**

In the pooling layer we used the max pooling to the input image with pool size (2, 2) using the ReLU activation function to reduce the amount of parameters which further reduces computational cost and reducing overfitting.

* **Dropout Layer:**

After training, the network weights overfit the training samples provided, and the network performs poorly on new samples. This level "clears" random groups of activations within this level by setting them to zero.

**Algorithm for converting speech to sign language:**

1.We taken the input from the user as speech.

2. Concert the speech to text with the NLP.

3.Then process the text and match it with the present data of gif and show the gif if available.

4.If the gif is not available in the data, then show the sign of each letter in the text.

**Finger Spelling Implementation:**

1. We display one letter at a time with the count of 3 and add it to the current letter or to the string of letters which are already identified which forms a word.

2. We can also delete the current detected letter or word, which records the number of times the current symbol has been detected, in order to reduce the likelihood that the wrong letter would be predicted.

3. A blank frame or image (plain backdrop) is used for the blank space. If the blank space detections exceed certain values, then no gaps are recognized.

4. The output of the detection is stored in the sound mp3 file and used for the voice of the text.

Diagram

Description automatically generated

**Training & Testing:**

We have used the gaussian blur to reduce noise in the image and then resize our images to 128 × 128. We use adaptive threshold to separate hand from the background. We then give our model the pre-processed images.

Prediction layer helps to calculate the likelihood that the image will belong to a particular class. In order for the output to be normalized between 0 and 1, value in class must add up to 1 respectively by utilizing the SoftMax function.

For the training we have defined the signs and used them to train the model. The dataset we have defined is split into 80% for the training and 20% for testing.

Here in the project, we have done the Unit testing to check the code snippets if they are working as expected. We have also done the integration testing to check if all parts of the code work as expected if they are put together.

Finally, we have done the manual testing to check if we are able to product the expected out.

In manual testing we have defined we alphabets with sings and tested if we are getting the accurate results and used that result to test the output of speech.

While training the model to translate the speech to sign language, we have collected the data for a few gif and all the signs of an alphabet. We have trained the model with this data and when we are testing the model first searches if the text is available in gif data and gives the output.

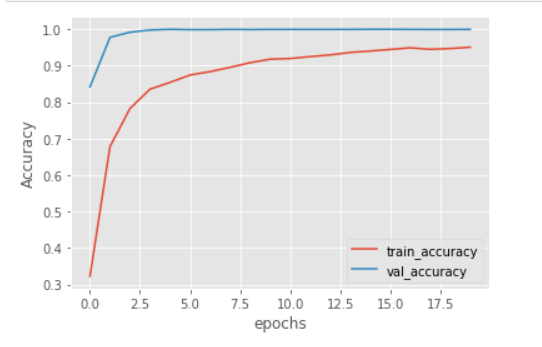
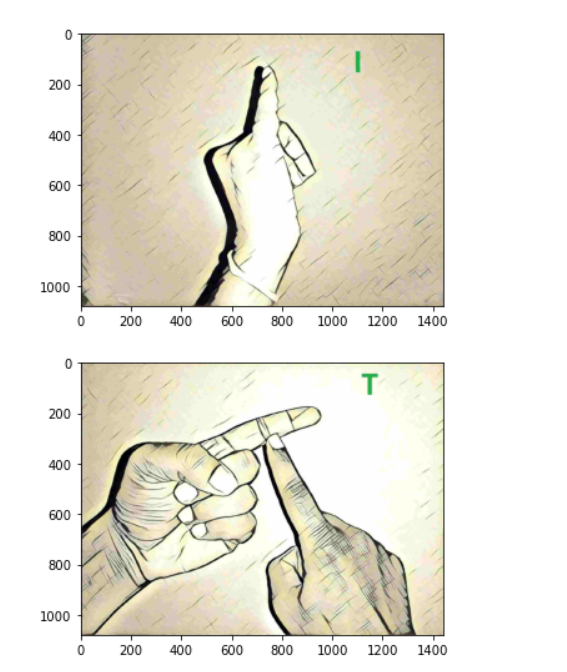
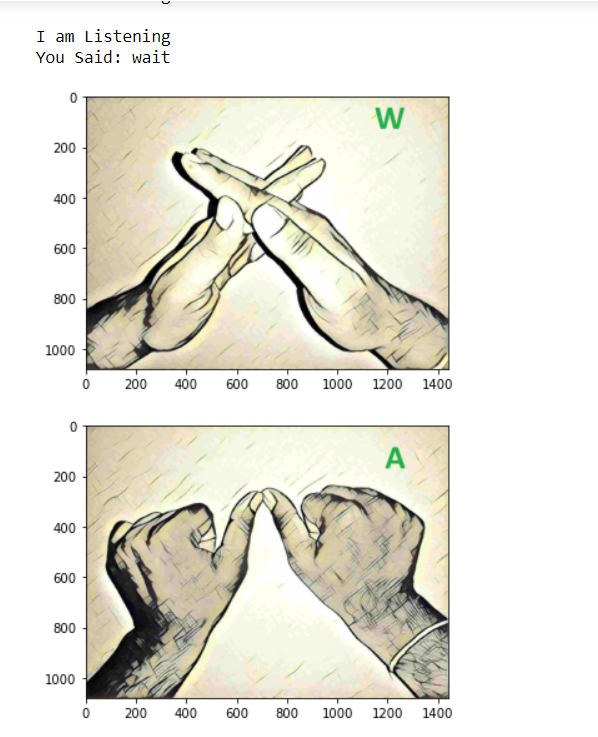
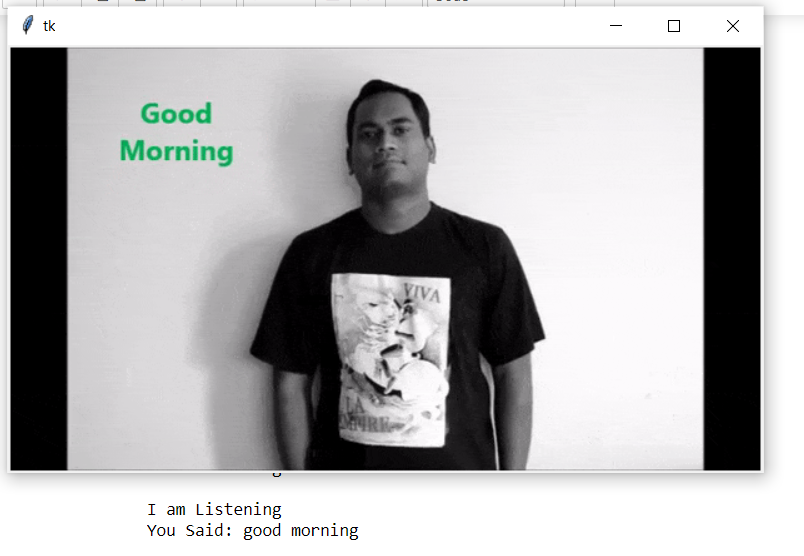
If it is not available in data then it gives the signs of all the alphabets in the text. We have done the unit testing for all the code snippets and integration testing to check if all the snippets of code work together as expected.

**Results:**

Using our algorithm, we were able to obtain an accuracy of 95.8% in our model. Most of research the publications concentrate on using Kinect-like devices for the detection of hand. Convolutional neural networks and Kinect are used to construct a Flemish sign language recognition system in, which achieves an error rate of 2.5%.

Additionally, CNN was used for the recognition system. It should be noted that unlike some of the models listed, our model does not use a background subtraction approach. As a result, there may be a range in accuracy if we try to apply background subtraction to the project. Since our model merely employs a basic laptop inbuilt webcam which is a huge benefit for the majority of audience instead of using sensor like Kinect, an expensive component.

Here are some of the screenshots of the results:

The results obtained after converting text of speech are included in the GitHub for the reference.

**Git hub link for the project:**

[**https://github.com/chandana341/Sign-language-to-text-speech**](https://github.com/chandana341/Sign-language-to-text-speech)

Here we have all the data that we have used in the project as well as the code.

Code files are uploaded separately for each task for better understanding and to make it easy for replication of project.

**Video demonstration :**

[**https://youtu.be/BpK\_pHA9-0M**](https://youtu.be/BpK_pHA9-0M)

**Project presentation :**

[**https://github.com/chandana341/Sign-language-to-text-speech/blob/main/Sign%20Language%20Translation.pptx**](https://github.com/chandana341/Sign-language-to-text-speech/blob/main/Sign%20Language%20Translation.pptx)

**Planned Works:**

In the base model we are planning to increase the accuracy of the mode. In case of complex backgrounds, the prediction is less accurate when compared to plain background, so we are planning to build the model to give a greater accuracy in such cases.

In case of complex backgrounds, the prediction is less accurate when compared to plain background, so we are planning to build the model to give a greater accuracy in such cases for converting signs to text.

In the present model given signs can be converted to text and given input of speech is converted to text and then to gif signs but it is limited to only pre-defined data. In the future model we are planning to develop into an application which can do both translations simultaneously for all the data.

In the future model we can also plan to develop model for the individual user to create their own signs and store then and use then for conversation through model which would be unique for user.

**Challenges:**

The most difficult challenge we have faced is finding the pre-defined data set for the problem. Since we were unable to find the data which satisfies our requirements then we have defined our own dataset. But this again caused the problem with the correct capture of the signs and reading in the signs.

Other challenge is predicting the correct letter with most accuracy. Few letters when captured as images in some angles coincides with the others this results in the incorrect prediction.

Improving the accuracy of the classifier to the maximum has become one of the challenges even in case of a complex background or a background which has more noise.

**Individual Contribution:**

Chandana Kolluru (16338510)

My role during project work is assembling of software components which are selected for the proposed model, finding the datasets and their implementation using suitable code and helped for documenting the report of the project. My work also includes testing the final outcome of the project by helping to fix the bugs in the code by study the existing system and drawbacks for modifying with better improvements. We supported each and every one in all the works.

Rohan (16338416)

My role during the project work is designing of the components for successful implementation of proposed model, and helped for data preparation of the project. We supported each and every one in all the works.

**Conclusion:**

In this report, a functional real-time vision based American Sign Language (ASL) recognition has been developed for applications in speech recognition for D&M people.

We have achieved an accuracy of 95% in our analysis. In our next steps, we have improved the accuracy using two layers of machine learning algorithms. This has allowed us to verify symbols which were more similar to each other and to predict them correctly.

This provides the ability to detect all the symbols which are provided and they haven’t been damaged during shipping, and there is no noise in the background and light that illuminates the image so it’s readable.

While converting speech to text and translating into sign language we achieved good accuracy and the signs predicted are also more clear for a D&M person to understand as well for a normal person to able to identify what they said and predicted matches.

**References:**

[1] https://github.com/chandana341/Sign-language-to-text-speech

[2]<https://techvidvan.com/tutorials/hand-gesture-recognition-tensorflow-opencv/>

[3]<https://create.arduino.cc/projecthub/173799/a-glove-that-translate-sign-language-into-text-and-speech-c91b13>

[4]<https://www.instructables.com/Sign-Language-Translation-Into-Text/>

[5]<https://www.researchgate.net/publication/335433017_Conversion_of_Sign_Language_To_Text_And_Speech_Using_Machine_Learning_Techniques>

[6]T. Yang, Y. Xu, and “A., Hidden Markov Model for Gesture Recognition”, CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, May 1994.

[7]<https://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html>

[8]Waleed Kalous, Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language.

[9]aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural Networks-Part-2/

[10]<http://www-i6.informatik.rwth-aachen.de/~dreuw/database.php>

[11] Number System Recognition (<https://github.com/chasinginfinity/number-sign-recognition>)

[12] <https://opencv.org/>

[13] https://github.com/aniketdhole07/two-way-sign-language-translator