Sign Language To Text Conversion

NUS

Group 8

Aniket Nitish

Sagnik Siddhant

Umang

Content

Abstract

Major development is being done in research for Sign Language Recognition. A lot of techniques have been developed in this field.

The Sign Language is used essentially for deaf and dumb people to communicate with each other and sign language is one of the oldest and most natural forms of language for communication, but since most people do not know sign language and interpreters are very difficult to come by, we have come up with a real time method using neural networks for fingerspelling based on American sign language.

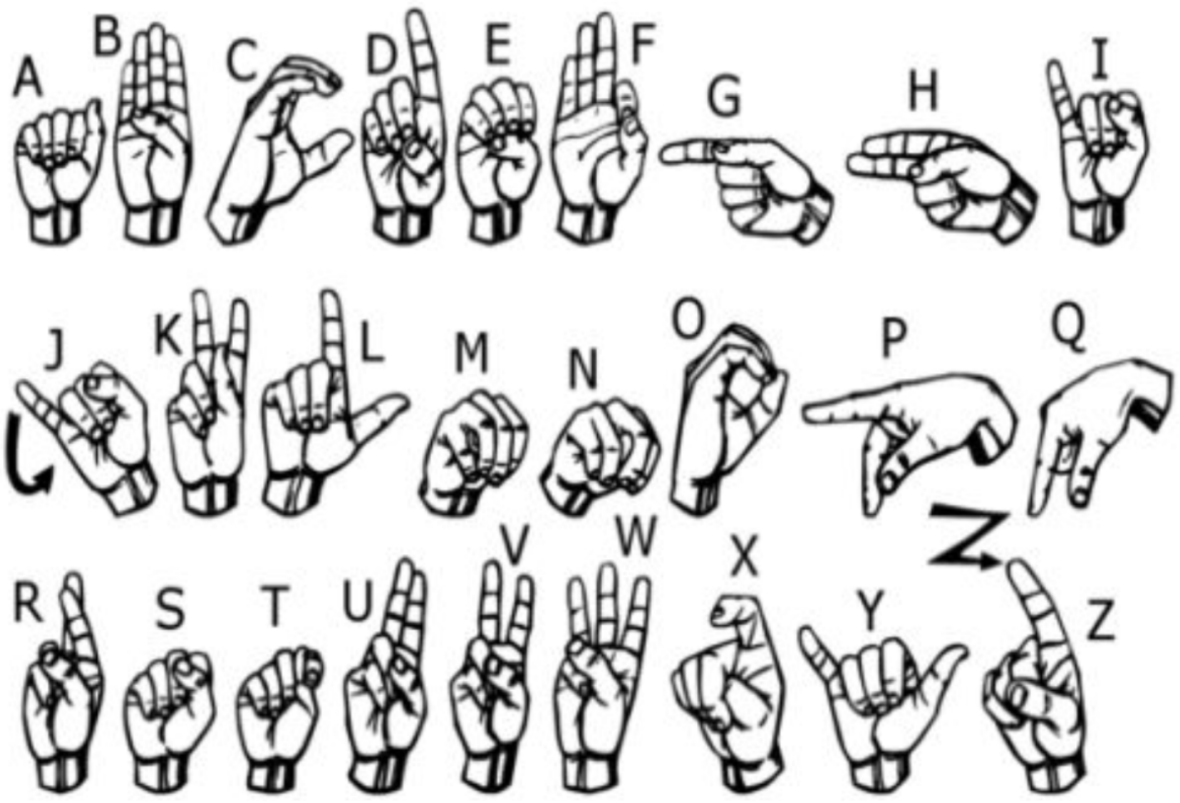
In our method, the hand is first passed through a filter and after the filter is applied the hand is passed through a classifier which predicts the class of the hand gestures.

Introduction

American sign language is a predominant sign language Since the only disability D&M people have is communication related and they cannot use spoken languages hence the only way for them to communicate is through sign language.

Deaf and dumb(D&M) people make use of their hands to express different gestures to express their ideas with other people. Gestures are the nonverbally exchanged messages and these gestures are understood with vision. This nonverbal communication of deaf and dumb people is called sign language.

In our project we basically focus on producing a model which can recognise Fingerspelling based hand gestures in order to form a complete word by combining each gesture. The gestures we aim to train are as given in the image below



Business Statement

Around 360 million human beings globally are afflicted via unable to hearing loss out of which 328000000 are adults and 32000000 children. A hearing impairment extra than 40 decibels in the better listening to Ear is referred as disabling listening to loss. Thus, with growing range of people with deafness, there is moreover a rise in demand for translators

Minimizing the verbal exchange gap among listening to impaired and regular humans turns into a want to make certain effective conversation among all.

For interaction between normal people and D&M people a language barrier is created as a sign language structure which is different from normal text. So they depend on vision based communication for interaction.

If there is a common interface that converts the sign language to text the gestures can be easily understood by the other people. So research has been made for a vision based interface system where D&M people can enjoy communication without really knowing each other's language.

The aim is to develop a user-friendly human computer interface (HCI) where the computer understands the human sign language. There are various sign languages all over the world, namely American Sign Language (ASL), French Sign Language, British Sign Language (BSL), Indian Sign language, Japanese Sign Language and work has been done on other languages all around the world.

Methodology

Dataset Generation

Two Types of Dataset was used for training and testing purpose

a)First Dataset Consisted of Gestures that were taken from four different angles.

It's a open source dataset the HGM-4 dataset is built for hand gesture recognition (the full dataset is available from: https://data.mendeley.com/datasets/jzy8zngkbg/4) which contains total 4,160 color images (1280 × 700 pixels) of 26 hand gestures captured by four cameras at different position.

b)The Second Dataset was created by our own

We used the Open computer vision(OpenCV) library in order to produce our dataset. Firstly we captured around 800 images of each of the symbols in ASL for training purposes and around 200 images per symbol for testing purposes.

Gesture classification

Our approach uses two layers to predict the final symbol of the user.

Layer1:

1. Apply gaussian blur filter and threshold to the frame taken with opencv to get the processed image after feature extraction.
2. This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.
3. Space between the words are considered using the blank symbol.

Layer 2:

1. We detect various sets of symbols which show similar results on getting detected.
2. We then classify between those sets using classifiers made for those sets only.

CNN Model

1.**1st Convolution Layer :**The input picture has a resolution of 128x128 pixels. It is first processed in the first convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 126X126 pixel image, one for each Filter-weights.

2. **1st Pooling Layer :** The pictures are downsampled using max pooling of 2x2 i.e we keep the highest value in the 2x2 square of array. Therefore, our picture is downsampled to 63x63 pixels.

3. **2nd Convolution Layer :**

Now, these 63 x 63 from the output of the first pooling layer is served as an input to the second convolutional layer.It is processed in the second convolutional layer using 32 filter weights (3x3 pixels each).This will result in a 60 x 60 pixel image.

4. **2nd Pooling Layer :**

The resulting images are downsampled again using a max pool of 2x2 and is reduced to 30 x 30 resolution of images.

5. **1st Densely Connected Layer :**

Now these images are used as an input to a fully connected layer with 128 neurons and the output from the second convolutional layer is reshaped to an array of 30x30x32 =28800 values. The input to this layer is an array of 28800 values. The output of this layer is fed to the 2nd Densely Connected Layer.We are using a dropout layer of value 0.5 to avoid overfitting.

6. **2nd Densely Connected Layer :**

Now the output from the 1st Densely Connected Layer is used as an input to a fully connected layer with 96 neurons.

7. **Final layer:**

The output of the 2nd Densely Connected Layer serves as an input for the final layer which will have the number of neurons as the number of classes we are classifying (alphabets + blank symbol).

Activation Function :

We have used ReLu (Rectified Linear Unit) in each of the layers(convolutional as well as fully connected neurons).ReLu calculates max(x,0) for each input pixel. This adds nonlinearity to the formula and helps to learn more complicated features.It helps in removing the vanishing gradient problem and speeding up the training by reducing the computation time.

Pooling Layer :

We apply **Max** pooling to the input image with a pool size of (2, 2) with relu activation function.This reduces the amount of parameters thus lessening the computation cost and reduces overfitting.

Dropout Layers:

The problem of overfitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn't perform well when given new examples.This layer "drops out" a random set of activations in that layer by setting them to zero.The network should be able to provide the right classification or output for a specific example even if some of the activations are dropped out[5].

Optimizer :

We have used Adam optimizer for updating the model in response to the output of the loss function. Adam combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm(ADA GRAD) and root mean square propagation(RMSProp)

Layer 2: Classifier

We are using two layers of algorithms to verify and predict symbols which are more similar to each other so that we can get us close as we can get to detect the symbol shown. In our testing we found that following symbols were not showing properly and were giving other symbols also :

1. For D : R and U
2. For U : D and R
3. For I : T, D, K and I
4. For S : M and N

So to handle above cases we made three different classifiers for classifying these sets:

1. {D,R,U}
2. {T,K,D,I}
3. {S,M,N}

Training and Testing

We convert our input images(RGB) into grayscale and apply gaussian blur to remove unnecessary noise.We apply adaptive threshold to extract our hand from the background and resize our images to 128 x 128.

We feed the input images after preprocessing to our model for training and testing after applying all the operations mentioned above.

The prediction layer estimates how likely the image will fall under one of the classes. So the output is normalized between 0 and 1 and such that the sum of each value in each class sums to 1. We have achieved this using the softmax function.
At first the output of the prediction layer will be somewhat far from the actual value. To make it better we have trained the networks using labeled data. The cross-entropy is a performance measurement used in the classification. It is a continuous function which is positive at values which are not the same as labeled value and is zero exactly when it is equal to the labeled value. Therefore we optimized the cross-entropy by minimizing it as close to zero. To do this in our network layer we adjust the weights of our neural networks. TensorFlow has an inbuilt function to calculate the cross entropy.

AutoCorrect Feature:

A python library **Enchant** is used to suggest correct alternatives for each (incorrect) input word and we display a set of words matching the current word in which the user can select a word to append it to the current sentence.This helps in reducing mistakes committed in spellings and assists in predicting complex words.

GUI with Tkinter

We Use Tkinter Module to develop the Graphical User Interface which is very easy to use and accessible. It Captures and Predicts the sign symbol in real time.

Conclusion

In this project, a functional real time vision based american sign language recognition for D&M people have been developed for asl alphabets.

Using various concepts of image processing and fundamental properties of image we tried to develop this system. By using recognition of gesture has done successfully

Any user can access the program and sit in front of the camera. He/ She can make symbols of American Sign Language (ASL). The system will simultaneously show the corresponding English text.

We realised that sign languages are spoken more in context rather than as finger spelling languages, thus, the project is able to solve a subset of the Sign Language translation problem

The main objective has been achieved, that is, the need for an interpreter has been eliminated.

We achieved final accuracy of 95% on our dataset. We are able to improve our prediction after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other.

This way we are able to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate.

Future Enhancement

We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms.

We are also thinking of improving the preprocessing to predict gestures in low light conditions with a higher accuracy.

To add more sign languages of different regions

To fully deploy as a Play Store Application that can easily be accessible by everyone.