Dynamic sign language translating system using deep learning and natural language processing

Aishwarya Kulkarni [a], Pranav Halgekar[b], Girish.R.Deshpande[c], Anagha Rao[d], AishwaryaDinni[e]

[a]Student, Dept. of CSE, GIT,Belgaum-590008
[b]Student, Dept. of Mechanical Engineering, GIT, Belgaum-590008
[c]Assistant professor, Dept. of CSE, GIT, Belgaum-590008
[d]Student, Dept. of CSE, GIT, Belgaum-590008
[e]Student, Dept. of CSE, GIT, Belgaum-590008

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: People around the world with speech and hearing impairment use a media of communication known as 'Sign language'.

In recent times, Sign language is omnipresent. However, there exists a challenge for people who do not know sign language, to communicate with people who can communicate exclusively using sign language. This gap can be bridged by using technologies of recent times to recognize gestures and design intuitive systems with deep learning. The aim of this paper is to recognise American Sign Language gestures dynamically and create an intuitive system which provides sign language translation to text and speech of various languages. The system uses Convolutional neural network, natural language processing, language translation and text-to-speech algorithms. It is capable of recognizing hand gestures dynamically and predicting the corresponding letters to form a desired sentence accurately.

Keywords: Sign language recognition, convolution neural network, text to speech conversion, translation, Vader sentiment analysis.

1. Introduction

Recognition of gesture can be implemented in a wide variety of applications. With the increasing capability of computers to detect and apprehend human actions, greater

convenience can be achieved by delegating

such work to machines. One such area is

Sign language recognition.

Keywords: Sign language, Convolutional Neural Network, translation, text-to-speech, sentiment analysis, canny edge detection.

2. Literature Review

The recent development in technology has led to new and advanced equipments and methods used by people affected with hearing loss and speech loss to converse easily and frequently with ordinary people and vice versa.The work so far gone for the development of such systems includes various technologies such as portable smart gloves which use LED, LDR sensors and micro controllers for recognition[1], Convolutional Neural Networks which automate feature construction and recognise gestures with high accuracy[2], dynamic loading and processing of images[6],finger detection achieved based on boundary tracing and fingertip detection [3], baseline system converts sentence level gestures into synthesis speech[4], feature extraction on recorded video frames to create sign language feature space[5], recognising static images using deep neural networks,application of natural language processing by extracting sentiments based on subjectivity classification, semantic association, and polarity classification [8], VADER- a sentiment analysis tool to analyse reviews at a faster rate[9] and others.

3. Existing Systems

Drawbacks of the current systems:

1) Recognition with electronic gloves :

Even though they provide dynamic recognition, they require initial investment cost for the devices and atleast basic knowledge pertaining to electronics

2) Recognition without gloves:

i) Static: They take static images as input whose sign is predicted and given as output. This delay in prediction each time an input is given is very inconvenient.

ii) Dynamic: They need to have an accurate model with a great accuracy which is not practically possible to achieve always, as even if the recognition is dynamic, due to less accuracy, a letter categorized into a particular class label as an incorrect letter, remains same throughout. It has partial staticness.

3) There is an availability of conversion to speech. However, it is restricted only to global languages. This makes it less user-friendly to local subjects in a diverse country like India.

4. Proposed System

The objective of this paper is to build a more hybrid neural network and natural language processing system which is capable of recognising, interpreting and expressing dynamic hand gestures as per the ASL standard to 10 different spoken languages. The system provides an ability to adjust the gestures dynamically for proper recognition of letters which prevents the chances of a letter getting recognized as a different one and remaining the same throughout. It is capable of recognising dynamic hand movements and associating them to static hand gestures accurately to form a sentence, convert into speech of different languages and detect the sentiment behind it.

4.1 ER Diagram

The conceptual model of the proposed system in the form of ER diagram is as shown in fig (4.1).

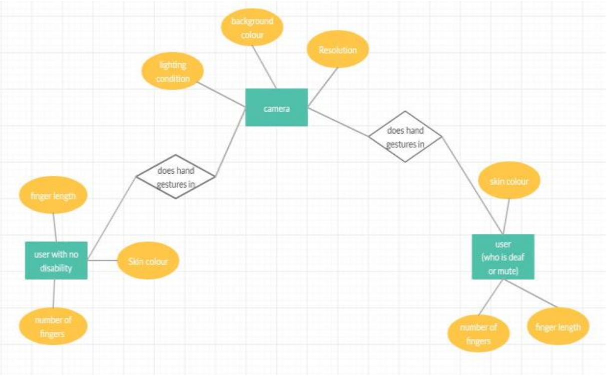


Fig 4.1: ER Diagram for the system

This entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates the complete system's entities. The entities of this system are the users with disability and no disability and a laptop with a webcam.

4.2 Sequence Diagram

The sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

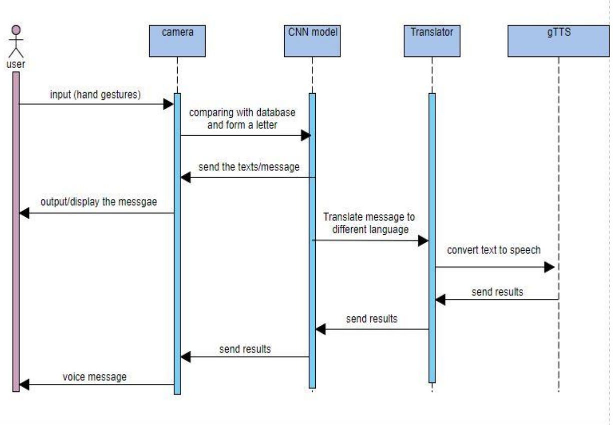


Fig 4.2: Sequence Diagram for User Module

5. Methodology And Implementation Details

The hybrid model is built in the following stages

1. Developing a Convolutional Neural Network model to recognise ASL hand gesture using pre-existing dataset of American sign language.

2. Capturing image frame and image processing of dynamic ASL hand gesture for prediction.

3. Application of an algorithm to translate predicted data into speech.

4. Application of Vader sentiment analysis on the output text to find the percentage of every sentiment conveyed in the statement.

All these phases can manage complexity without compromising on user experience.

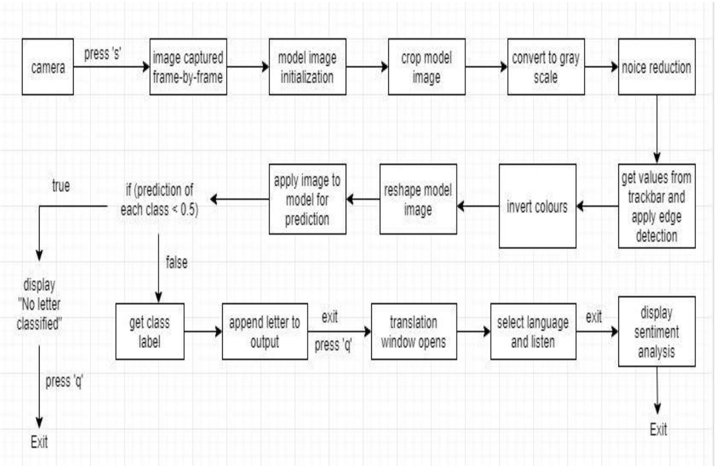

camera
press 's'
image captured frame-by-frame
model image initialization
crop model image
convert to gray scale
noice reduction
get values from trackbar and apply edge detection
invert colours
reshape model image
apply image to model for prediction
if (prediction of each class < 0.5)
true
display "No letter classified"
press 'q'
Exit
false
get class label
append letter to output
exit
press 'q'
translation window opens
select language and listen
exit
display sentiment analysis
Exit

Fig 5.1: Hybrid system architecture

Fig (5.1) gives the overview of the system components and the flow of the system. Primarily, the camera captures the gestures portrayed by the signers. When's' is pressed on the keyboard, the image is captured frame-by-frame. The image then undergoes a series of image pre-processing steps to enhance the accuracy. The pre-processed image is fed as input to the trained CNN model and the class of the image is predicted. After recognition, 'q' is pressed to exit. The translation window opens where the desired language can be chosen. 'Listen' is clicked to hear the speech of the formed text in the translated language. After exiting this window, the sentiment analysis window containing the percentages of the nature/sentiment of the text formed with the final sentiment recognized is displayed.

5.1Convolutional Neural Network

A Convolutional Neural Network or CNN model is used to classify the images in the dataset. The dataset used is downloaded from Kaggle's open source database. While building the neural network we first define the input layer. Each image is converted to a series of numbers. The input layer is processed by the neural network's hidden layers. The network uses 6 hidden layers. The architecture of our neural network can be seen in fig (5.2).

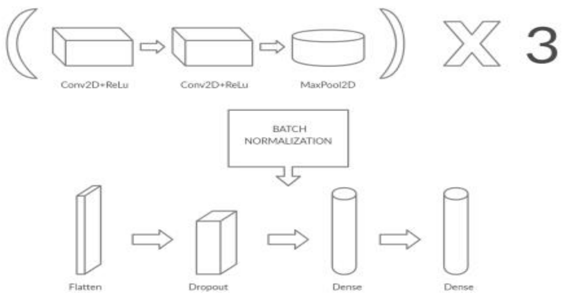


Fig 5.2: CNN model building architecture

A rectified linear unit or ReLU function is used for the neural network activation. The outputs from the ReLU serve as the inputs to the next hidden layer in the network. Max pooling is done after every 2 hidden layers. Once the data has passed through the Convolution and MaxPool layers of the neural network, it undergoes Batch Normalization and finally enters the Flatten and Dense layers. These layers help reduce the data to one dimension and identify an image’s class. Multiclass classification of the image is done by using the softmax function. The model has:

Evaluation Accuracy = 99.63%

Evaluation loss = 0.274437

5.2 Image Pre-processing and prediction

The dynamic image frame captured by the camera undergoes a series of image pre-processing steps by using the opencv image processing library in python. The image is cropped using the frame.copy() class. It is then converted to grayscale using the cvtColor() function. Noise in the image is reduced using GausianBlur() with (5,5) size of kernel and 0 standard deviation in X direction, erode() and dilate() functions are used with 2 iterations. Threshold values are adjusted using the trackbar position with the getTrackbarPos() function. Canny edge detection algorithm is used to detect the edges of the image. The image is then resized and reshaped using the reshape() and resize() functions respectively. The canny image’s colours are inverted and fed as the model image which undergoes resizing to 100 and interpolation. It is then converted to a float numpy array . The pre-processed image is then given as input to the CNN model and the class of the image is predicted by the loaded CNN model. If the prediction of each class is less than 0.5 then a message “None. Try again” is displayed or else, the class label (letter predicted) is displayed. This letter is then appended to the output variable if the same gesture is held on for 15 frames.And each of these images is fed to the CNN model and the class is predicted. All the predictions are then appended to form a sentence.

5.3 Translation and text to speech

After the dynamic image recognition of the gestures is completed, a sentence is formed as the output. This sentence is translated into desired language and then converted to speech. The sentence is translated using the ‘Translator’ class from the googletrans library .The translation window opens where the desired language can be chosen. The Translate() function from the ‘Translator’ class translates the output text into various languages . ‘Listen’ is clicked to hear the speech of the formed text in the translated language. Internet connectivity is required for the conversion. The function gTTS imported from gtts(google text to speech) library helps in converting the text to speech in the desired language. This is stored as an audio(.mp3) file on the disk .

5.4 Sentiment Analysis

The output text undergoes sentiment analysis using the Vader Sentiment Analysis.The SentimentIntensityAnalyser class is imported from vaderSentimentlibrary. The function polarity_scores() is used to find the percentage of every sentiment (positive, negative and neutral). It also considers the factors of capitalization and punctuation. This is displayed in a new window.

6.Design Details

6.1 Model

The CNN model is trained using around 920 images of each letter adding upto 26,680 images in total. The model is tested using 230 images of each letter adding upto 6,670 images. The libraries used are Keras, Tensorflow, cv2 etc. The function Conv2D with filters of 16,32,64,128 and 256 are used to form layers that create convolutional kernel which winds with input layers to give output tensors. The layers have a kernel size of [3,3] and the given input size is (100,100,1). MaxPooling is done after every 2 layers with a pool size of [3,3]. The Dropout layer has a value of 0.5 and the Dense layers have 512 and 29 number of classes determined empirically. This model is compiled using the ‘adam’ optimizer. Keras is used to load the saved Convolutional model.

Fig (6.1) shows the unique labels for each unique gesture .

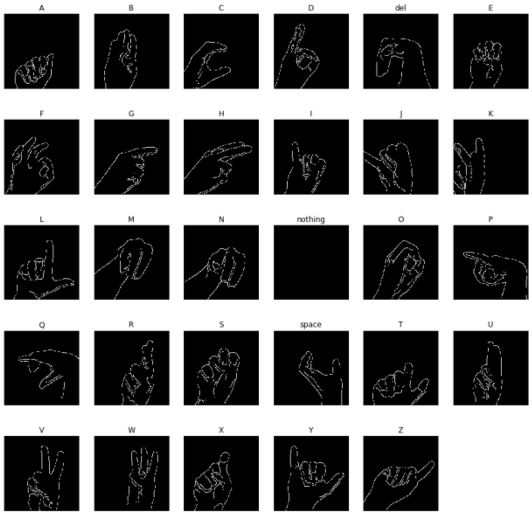


Fig 6.1: Unique labels

Fig (6.2) and Fig (6.3) show the accuracy plot and the loss plot of the train v/s test data of the loaded model when fit to the data for 10 epochs with a batch size of 64.

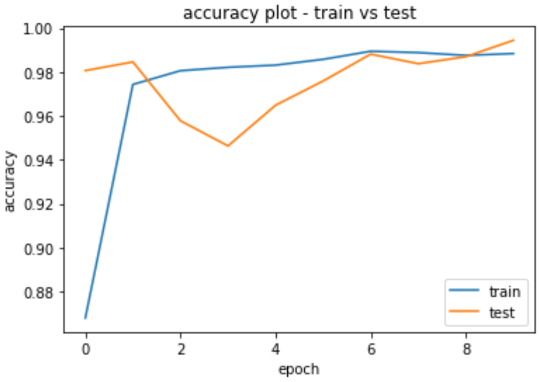


Fig 6.2: Accuracy plot

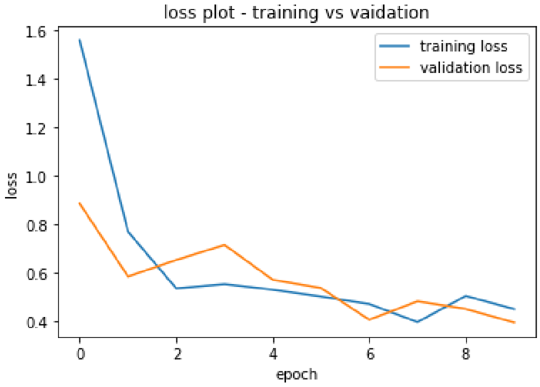


Fig 6.3: Loss plot

Fig (6.4) is a plot of the confusion matrix depicting the performance of the classification model for 29 labels.

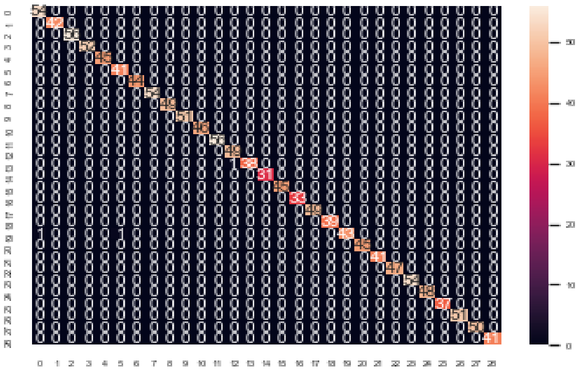


Fig 6.4: Confusion matrix

6.2 Recognition

The frame dimensions for capturing the video is 700*500. The frame inside the main video frame to perform the gesture and send to the model for recognition is of dimensions 200*200 (starting at x0=70 and y0=125) and the borders of this box are coloured blue. The class into which the letter is classified is displayed in the model picture frame. The canny window is of default size which displays the edges detected in the model picture frame dynamically simultaneous to recognition.

The threshold value for canny edge detection is set to 100 (lower threshold) using the track bar created to adjust detection. This canny image captured is sent for image processing as mentioned in topic (5.2) and a gesture with threshold value greater than or equal to 0.85 i.e, if the ratio of the same letter in the last 15 predicted letters is greater than or equal to 0.85 then it is appended to the output.

6.4 User Interface

The User Interface for the system is very intuitive. Pressing 's' starts and stops the recognition of gestures, pressing 'f' is equivalent to a backspace , pressing 'd' deletes the whole output sentence to start fresh and pressing 'q' quits the application window. Recognition for 'space', a 'delimiter' i.e, a comma and also 'nothing' is also included along with the 26 English alphabets. After recognition, the 500*300 window for translation opens with a dropdown of languages. Then the audio for the selected language plays. Finally the 500*300window for sentiment analysis pops up with detailed information of each sentiment.

7. Comparison

Existing systems implement their own unique methods of easing the lives of people to communicate with sign language. A 3D (CNN) takes out features which are spatial-temporal from a video stream that is raw, without features of designing [12] but a 3D CNN leads to a greater computational cost . Our model proposes the use of a 2D CNN which provides similar recognition and also adapts to large variations. The use of multi-sensors for real-time recognition system of sign language with multiple sensors data fusion (MSDF) hidden Markov models (HMM) prove to be better than single sensor framework [11] however our system provides similar consistent results without any investment on sensors or electronics. Usage of Leap Motion Sensor (LMC) for dynamic recognition [13] however this can have inaccurate results with changing positions of hand and finger movements. Our system provides a way to overcome this by providing the freedom for adjustment during dynamic recognition by displaying the letter/class that the gesture is getting characterized into so the user can adjust accordingly before appending it to the output. There are systems which provide similar recognition, computational costs and implementations, but they are limited to recognition and conversion to speech using various techniques like HMM (Hidden Markov Model)[14] with high computations. Our proposed system combines recognition with improved conveyance like translation to varied languages, to speech using gTTs and also sentiment analysis which understand the emotions behind words and also capitalization and punctuation combined all into one with an interactive interface. The current systems lag to provide a user friendly interface for recognition.

8. Results And Analysis

8.1 Recognition

Fig (7.1) shows the window for capturing the original gesture done by the Signer.

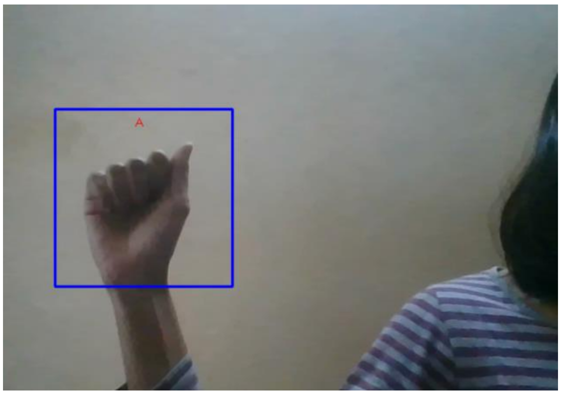


Fig 7.1: Original image window

1. Condition: Perform gestures without pressing ‘s’.

Output: Recognition starts however doesn’t append the recognized letter to the output sentence

2. Condition: Perform gestures in very bright lighting conditions

Output: System remains as it is

3. Condition: Perform gestures in very dark conditions

Output: System shows empty screen with no recognition

4. Condition: Perform gestures in proper lighting conditions

Output: System recognizes the letter

5. Condition: Perform gestures with varied finger sizes

Output: System recognizes the letter

6. Condition: Perform gestures with varied number of fingers

Output: Inconsistency in recognition

Fig (7.2) shows the window for displaying the canny image with all the edges recognized. It also has a threshold adjustment bar for better edge recognition using threshold intensity manipulation for varied physical conditions (background, lighting, hand/finger size, skin colour)

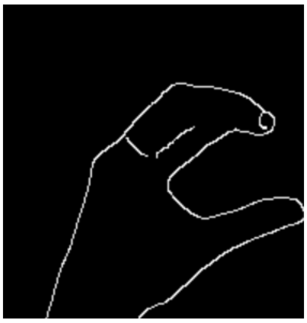


Fig 7.2: Canny image window

1. Condition: Perform gestures with discrepancies in the background (not plain)

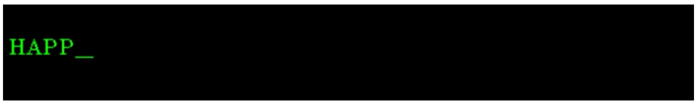
Output: Inconsistency in recognition due to edge detection of all entities present in the screen

2. Condition: Perform gestures with plain background

Output: Consistency in recognition

Fig (7.3) shows the window for displaying the prediction appended to the output after the gesture is held on for fixed number of

frames.



```
HAPP_
```

Fig 7.3: Output window

1. Condition: Not holding the gesture for 15 frames

Output: No letter appended to the output sentence.

2. Condition: Quit without pressing ‘q’

Output: System remains as it is.

8.2 Translation and speech

Once the recognition is complete, the translation window opens where the user chooses the desired language and clicks on ‘LISTEN’ to hear the audio of the translated text.

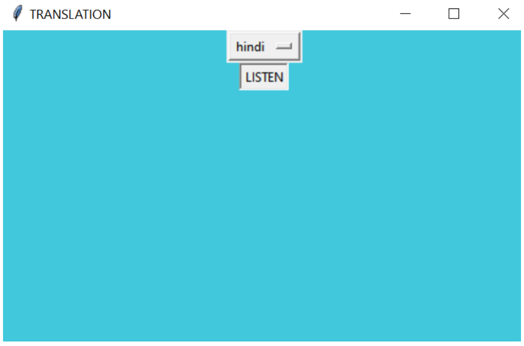
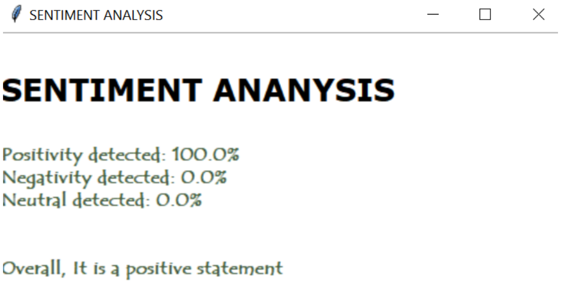


Fig 7.4: Translation and speech window

1. Condition: Click ‘Listen’ after choosing the language
Output: System plays the audio in the selected language
2. Condition: Click ‘Listen’ without choosing the language
Output: System remains as it is

8.3 Sentiment analysis

Fig (6.5) shows the sentiment analysis window which opens by default after exiting the translation window and shows the percentage of each sentiment expressed in the text.



SENTIMENT ANANYSIS

Positivity detected: 100.0%
Negativity detected: 0.0%
Neutral detected: 0.0%

Overall, It is a positive statement

Fig 6.5: Sentiment analysis

1. Condition: No recognition formed and system reaches sentiment analysis window

Output: No result is displayed in the window

9. Conclusion

A hybrid automated system which recognizes gestures and interprets them dynamically is built and implemented. It successfully demonstrates the use of primitive and complex deep learning concepts as the basic blocks of gesture recognition. It uses the concept of CNN model building efficiently with an accuracy of 99.63%. The system also manages to translate the output into a desired language and speech using the googletrans and gtts libraries to full extent. The users can now seamlessly find the sentiment and analyze the thought of the statement by the sentiment analysis provided by Vader which gives the degree of each sentiment expressed in the given statement as input. We found that, all in all a recognition system built on a deep learning framework should have an efficient model with the highest possible accuracy as the base along with other modules which provide an user-friendly way of conveying the recognized material to the end user. The future scope of this automated recognition system has great potential. The recognition can further be worked upon without wavers so that it can give better results on a portable device. The translation languages can have international options along with accent modification for a farther reach out of the people.

References

1. Nikitha Praveen, Naveen Karanth, MS Megha, Sign language interpreter using a smart glove (2014)
2. Lionel Pigou, Sander Dieleman, Peter-Jan Kidermans, Benjamin Schrauwen, Sign language recognition using Convolutional Neural Networks (2015)
3. Ravikiran J, Kavi Mahesh, SuhasMahishi, Dheeraj R, Sudheender S, Nitin V Pujari,Finger Detection for sign language recognition (2009)
4. Jiyong Ma, Wen Gao, Jiangqin Wu and Chunli Wang, "A continuous Chinese sign language recognition system" (2000)
5. Rao G, Kishore P.V.V., Sign Language Recognition System Simulated for Video Captured with Smart Phone Front Camera (2016)
6. P. S. Rajam and G. Balakrishnan, "Real time Indian Sign Language Recognition System to aid deaf-dumb people," (2011)
   A. Das, S. Gawde, K. Suratwala and D. Kalbande, "Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images," 2018
7. W. Y. Chong, B. Selvaretnam and L. Soon, "Natural Language Processing for Sentiment Analysis: An Exploratory Analysis on Tweets,"(2014)
8. Heather Newman, David Joyner,Sentiment Analysis of Student Evaluations of Teaching (2018)
9. N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In IEEE International Conference on Computer Vision (ICCV), (2017).
10. K. Fok, N. Ganganath, C. Cheng and . K. Tse, "A Real-Time ASL Recognition System Using Leap Motion Sensors," 2015
11. Jie Huang, Wengang Zhou, Houqiang Li and Weiping Li, "Sign Language Recognition using 3D convolutional neural networks," 2015
12. Chong TW, Lee BG. American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach. Sensors(Basel). 2018;18(10):3554. Published 2018 Oct 19. doi:10.3390/s18103554
13. P. Vijayalakshmi and M. Aarthi, "Sign language to speech conversion," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, 2016, pp. 1-6, doi: 10.1109/ICRTIT.2016.7569545.
14. **Article**
15. https://core.ac.uk/download/pdf/55693048.pdf
16. http://reports.ias.ac.in/report/19049/real-time-indian-sign-language-recognition
17. Initial dataset obtained from:
18. https://www.kaggle.com/grassknoted/asl-alphabet