Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

Smart Inquisitive DL Model for Assistive Communication

Samved Mani Satisha[1], Satvik Singh S[2], Pratik R Jain[3], Rakshitha V[4], Dheeraj D[5]
Dept. of Information Science and Engineering
Global Academy of Technology
Bangalore, India

***Abstract*—Language of signs is an acceptable language for communication among deaf and dumb people in our society. The society of the deaf and dumb admits a lot of obstacles in day to day life in communicating with their acquaintances. The most recent study done by the WHO reports that 300 million people present in the world have hearing loss. This gives us a need for the invention of an automated system which converts hand gestures into meaningful sentences and vice versa. To draw a stage nearer to these objective applications for conversion of hand gestures to speech, we use CNN with Deep Learning and TensorFlow. Similarly, we use Google SpeechRecognition API along with Python pyttsx3 library and OpenCV to achieve the objective of conversion of speech to hand gestures. The proposed work achieves a training accuracy of 95.03%.**

Keywords—CNN; Deep Learning; TensorFlow; OpenCV; Computer Vision; Image Processing; Hand Gestures; Speech

I. INTRODUCTION

Computers will increasingly influence our everyday life because of the constant decrease in the price of personal computers. The efficient use of computer applications requires more interaction. Human–computer interaction is assuming utmost importance in our daily lives. Thus, HCI has become an active and interesting field of research in the past few years.

Sign language has always been the primary way of verbal communication among people who are both deaf and dumb. While communicating these people become very helpless and thus are only dependent on hand gestures. Visual gestures and signs which are a vital part of ASL that provide deaf and mute people an easy and reliable way of communication. It consists of the well-defined code gesture where each sign conveys the particular meaning in terms of communication.

Hand Gestures Recognition to Speech subsystem involves converting hand gestures to speech. The hand gestures are captured through a webcam/camera and the letters of the alphabet are recognized. These recognized letters are then concatenated to form words and sentences. Lastly, the formed words and sentences are converted into speech. The Convolutional Neural Network (CNN) is used on 25 hand signals of American Sign Language in order to enhance the ease of communication.

Speech Recognition to Hand Gesture subsystem involves conversion of voice that is recorded and splitting it into individual letters. The specific hand gesture of each letter is combined together to return a stream of images corresponding to the speech.

II. METHODOLOGY

Most of the researchers/developers classified the gesture recognition system into getting the input image/data from the camera. This section introduces the way for operating the data provided from the user from cameras.

An essential objective is to create a framework which can distinguish between different human hand gestures by completing explicit motion before a webcam. The captured image is recognized by the system and the result is displayed to the user [1].

A. Hand Gesture Recognition to Speech Subsystem

OpenCV Library is used to recognize gestures through the webcam [2]. The captured image is adjusted for various properties. The letters are then predicted using our model. The CNN model generates h5 file and weights. The letters are then detected, and formed into words using space as a delimiter and displayed on the GUI. These detected words are then converted into speech using pyttsx3 library. The process flow of this subsystem is provided in Fig. 1.

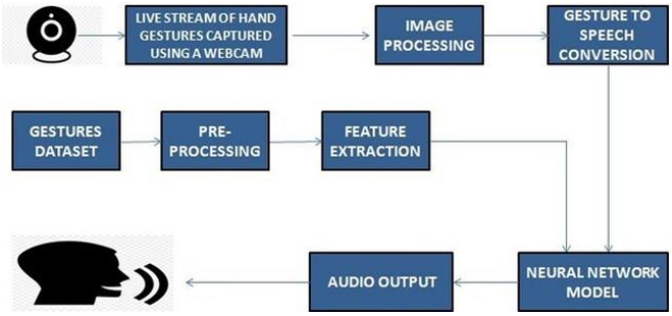

Fig. 1 Process Flow of Hand Gesture Recognition to Speech Subsystem

The CNN model comprises of three sets of Conv2D layers that are used for feature map creation and MaxPooling2D layers are used to reduce the size of these feature maps. The feature map at the end of the 3rd set of Conv2D and MaxPooling2D layers is converted into a single column vector using the Flatten function. This single column vector is then passed to the Fully Connected Layers.

The dense function is used to add the fully connected layer to the neural network, whereas the dropout function is used to reduce over fitting of the data by dropping 40% of the nodes in random from the neural network.

The final dense function uses SoftMax activation function to categorize the image into one of the 26 discrete class labels. The CNN Model Representation is shown in Fig. 2.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

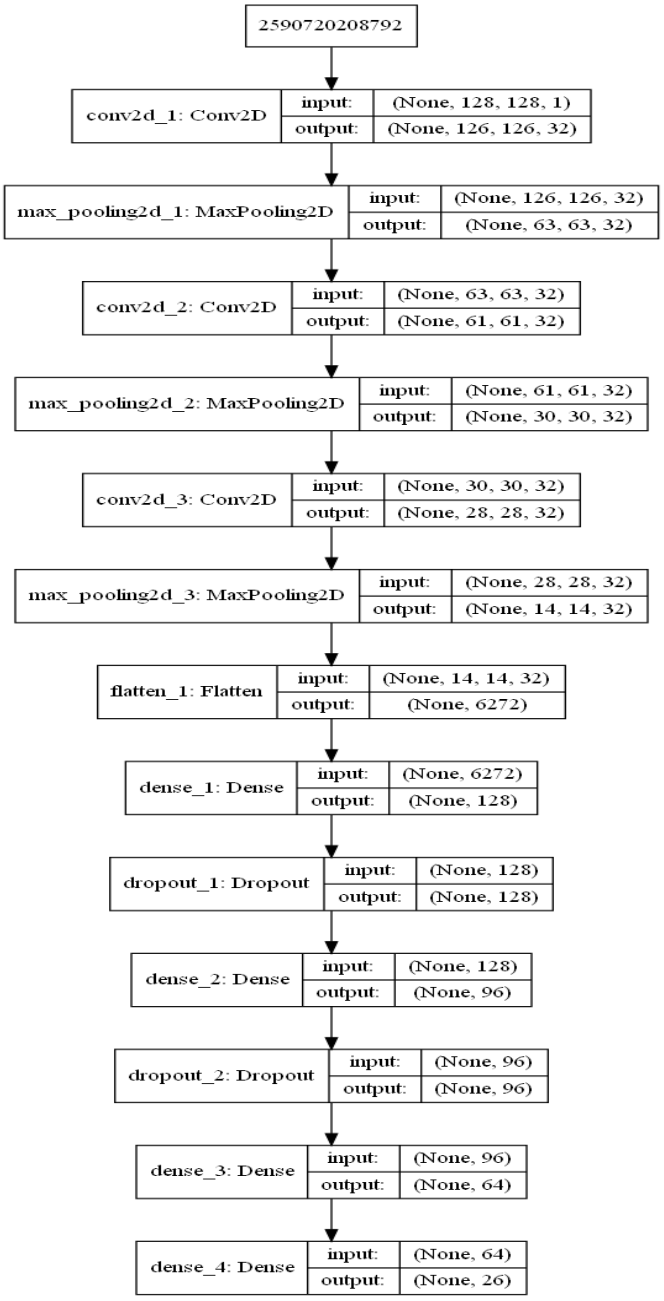

Fig. 2 CNN Model Representation

B. *Speech Recognition to Hand Gestures Subsystem*

. Google SpeechRecognition API is used to record the speech from the user. The recorded voice is converted into a string of text and then split into its individual letters. These letters are mapped to its respective hand gesture using a dictionary and then the hand gesture images are stitched together using OpenCV functionality to form a single image, which is then saved and displayed to the user. The process flow of this subsystem is provided in Fig. 3

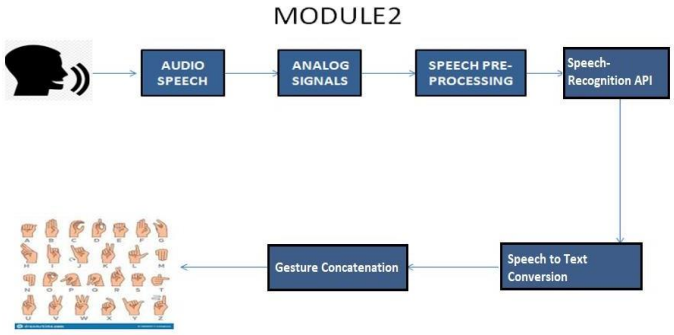

Fig. 3 Process Flow of Speech Recognition to Hand Gesture Subsystem

III. SYSTEM IMPLEMENTATION

The system working conditions and environment is based on Anaconda Environment Interface Design, OpenCV, TensorFlow, Tkinter, Numpy libraries and some of the sub packages of these libraries. Camera Resolution is 1920*1080 and with FPS of 30 (Default System Camera).

A. *Image Capture and Pre-processing*

We are creating our own dataset using OpenCV with 1200 images per alphabet and an additional 1200 images for the space gesture. The figure below represents the hand gesture image captured for alphabet A.

Fig. 4 ASL Hand Gesture for Alphabet A

The images captured are then processed to form grayscale images with boundaries highlighting the finger formation using OpenCV cvtColor function and then followed by OpenCV adaptiveThreshold function. This creates the image as seen in Fig. 5.

Fig. 5 Grayscale Image with finger boundaries for Alphabet A

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

B. User Interface

Flask is a micro web framework written in Python. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. Flask is used as the backend to host our webpage (Fig. 6) which redirects the user to the base interface from which the project is executed. The interface has been created using built in python library Tkinter.

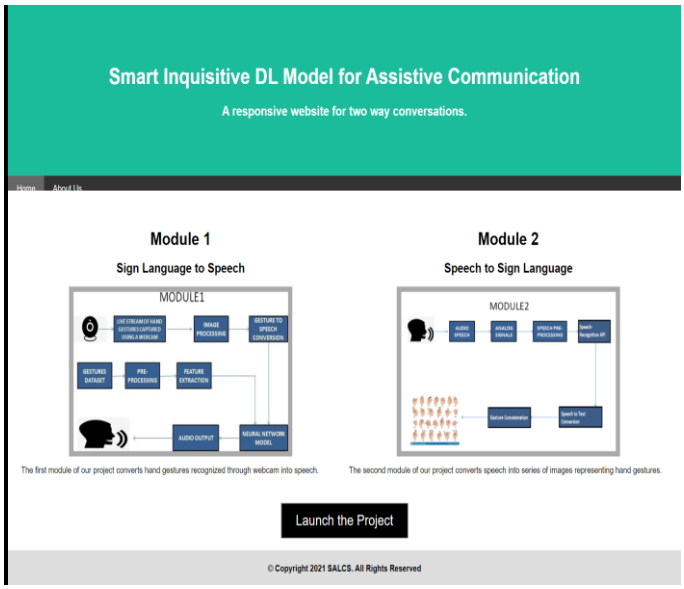

Fig. 6 Webpage running on Flask

The webpage leads to the base interface (Fig. 7) created using Tkinter. From the base interface, the user can redirect to either of the subsystems. The base interface is the medium that facilitates two-way communication.

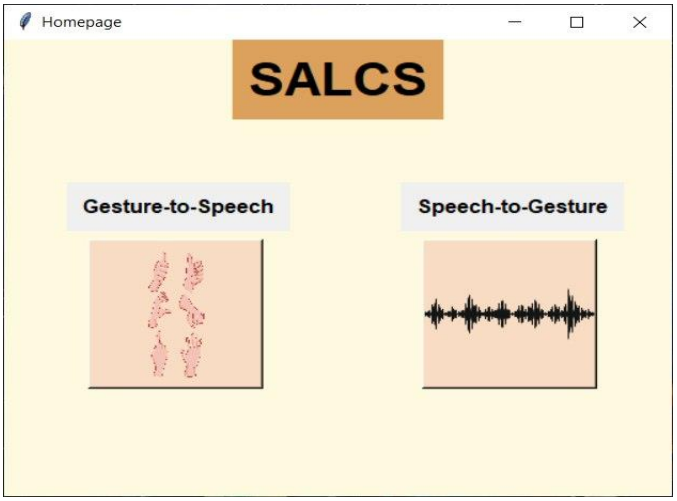

Fig. 7 Base Interface

Hand Gesture Recognition to Speech subsystem interface (Fig. 8) comprises of a real-time view of the hand gesture being captured. Once the specific hand gesture has been recognized, it is displayed in the letter exhibit. Corresponding gestures are recognized and concatenated together to form words. After the space gesture is recognized, the corresponding word is added to the sentence exhibit. On closing of this interface application, the sentence is passed to the pyttsx3 library, which saves it in an audio format and tells it out loud.

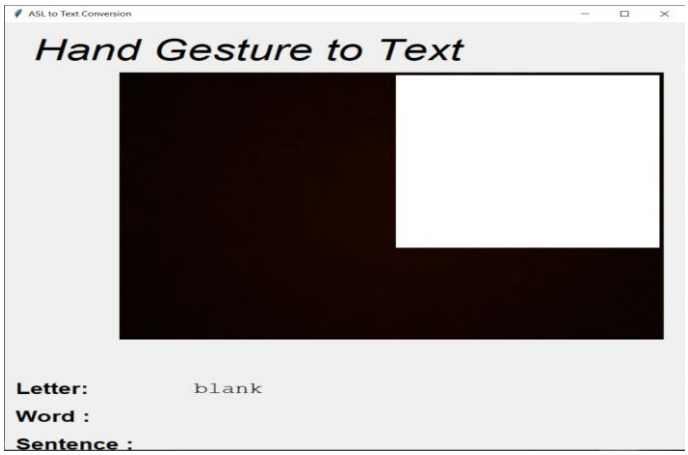

Fig. 8 Hand Gesture Recognition to Speech Interface

Speech Recognition to Hand Gesture subsystem interface (Fig. 9) allows the user to record their audio on the click of a button using the system microphone. This recorded audio is recognized using Google SpeechRecognition API and converted to its respective string of text. This string of text is displayed on the interface along with its corresponding stream of hand gesture images.

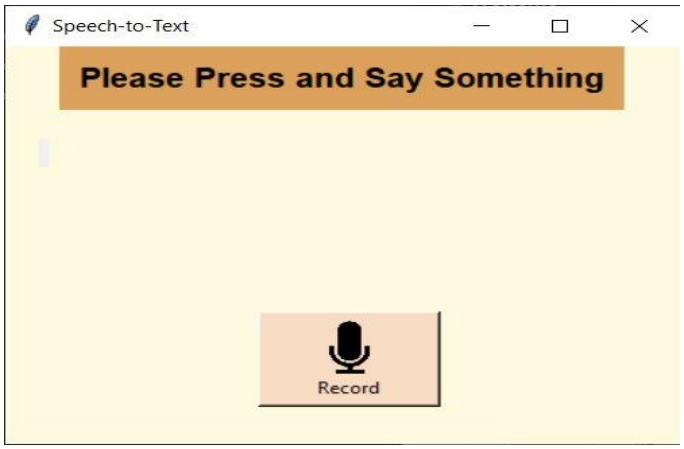

Fig. 9 Speech Recognition to Hand Gesture Interface

IV. PERFORMANCE ANALYSIS

To evaluate the performance of real-time hand gesture recognition, we are using 1200 images per alphabet along with another 1200 images for the hand space gesture. This amounts to a total of 31200 images that need to be classified into 26 categories. The training and test split ratio is 0.7:0.3. This results in training dataset with 21840 images and testing dataset with 9360 images. Both the datasets are shuffled before being sent to the CNN model for training and validation.

A. Dropout Variation

Usually, when all the features are connected to the Fully Connected layer in the CNN model, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data. To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 06, June-2021

To find the optimal dropout value for our CNN model, we tested with three different values for the dropout layer, namely 0.3, 0.4 and 0.5. For example, on passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

The CNN model is trained for 8 epochs with a batch size of 32. Keeping these values constant, the CNN model was trained for different dropout values and the results are tabulated below (Table 1).

Table 1 Dropout Variance Results

Dropout	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
0.3	0.0641	0.9798	0.5198	0.9248
0.4	0.1570	0.9503	0.3280	0.9353
0.5	0.1661	0.9481	0.3325	0.9357

From Table 1, we can conclude that, for dropout value of 0.3, the training loss and training accuracy are better due to the overfitting of the training data. This can be noticed in the marginally higher validation loss in the Neural Network. For dropout values of 0.4 and 0.5, there were marginal differences. However, dropout of 0.4 had a slightly better training accuracy and validation loss. Hence, dropout value of 0.4 was optimal for our CNN model.

B. Dataset Variance

To test and evaluate the performance of our CNN model, we trained the model on "RGB Image Dataset of American Sign Language Alphabets" published on Kaggle by Kapil Londhe [3]. The metrics used to train the model were kept constant, the only difference being, the datasets used to train the CNN model. The results are as follows.

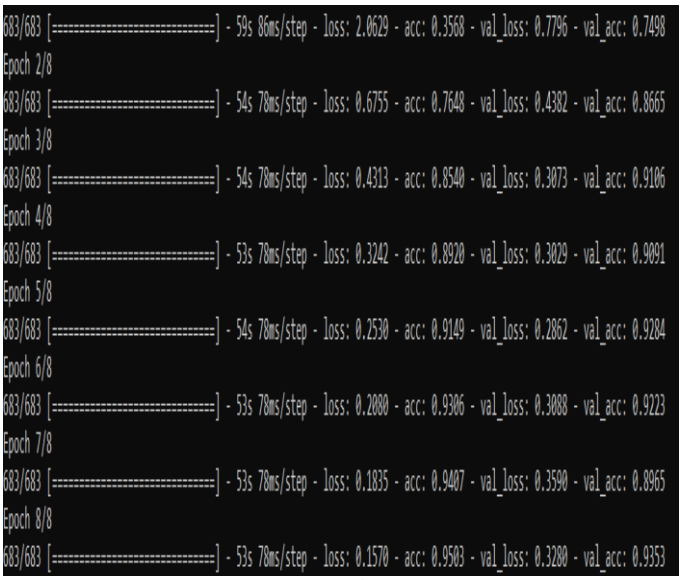

Fig. 10 Training Results for Original Dataset

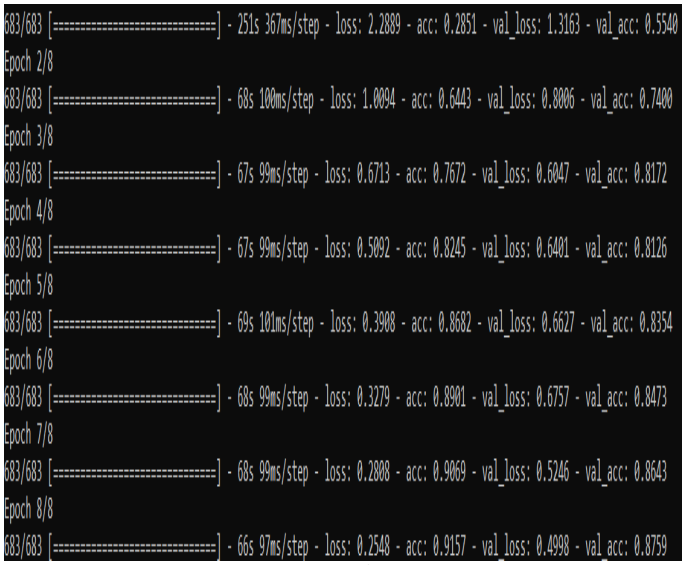

Fig. 11 Training Results for Kaggle Dataset

V. CONCLUSION

The proposed application would mean a new way of communication for about 300 and more million people with hearing and speech impairment to communicate and connect to people around them. This two-way conversation system would lower the communication gap between the deaf and mute community and the normal world.

The existing systems only work as individual components. There is a need for two-way communication system to enable efficient real-time communication between differently-abled users and normal users. The project aims to solve this issue by creating a single interface which enables two-way communication.

The work carried out has an accuracy of 95.03% and is able to recognize 25 American Sign Language Hand Gestures. This is the accuracy obtained on training the model with our dataset. An accuracy of 91.57% obtained on training the model on the Kaggle dataset.

Future enhancement would be to implement air motion tracking for the letter ‘Z’.

REFERENCES

[1] Mr. N. Sowri Raja Pillai, Miss. V. Padmavathy , S. Nasrin, “A Deep Learning Approach to Intelligent Gesture Recognition System for Deaf, Dumb and Blind Communication using KNN Algorithm on Tensor Flow Technique”, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 03, Mar 2019

[2] Kollipara Sai Varun, I. Puneeth and T. Prem Jacob, “Hand Gesture Recognition and Implementation for Disables using CNN’S”, International Conference on Communication and Signal Processing, April 4-6, 2019

[3] Kapil Londhe, “American Sign Language”, Kaggle, 2021, doi: 10.34740/KAGGLE/DSV/2184214