ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse651032021.pdf
https://doi.org/10.30534/ijatcse/2021/651032021

Sign Language and Common Gesture Using CNN

Piyush Kapoor[1], Hema N[2]
[1]Department of Computer Science and Engineering, Vellore Institute of technology, Chennai, India,
piyushkapoor3649@gmail.com
[2]hema.n@vit.ac.in

ABSTRACT

The Deaf, Dumb, and Blind Community and the general public have a true communication difficulty. The advancements made during the automated signing recognition try to break down the communication barrier. Our commitment considers a recognition method based on the Microsoft Kinect, convolutional neural networks (CNNs), and GPU acceleration. CNNs are prepared to automate the procedure of feature construction rather than developing intricate handcrafted features.We have a high level of accuracy in recognizing gestures and sign language. We also created more modules to make communication easier for persons with diverse abilities.This Project Mainly **t**o help the deaf, dumb and blind community by using this project approach this community can be able to communicate like a normal human being for this have used different modules for that.

Key words: CNN, Deciphering, Extraction, Text-Scrapping

1. INTRODUCTION

There are numerous differently-abled persons in every part of the earth, as we can see. It's true that no one is born without flaws. Over the last two decades, there has been a rising awareness that institutionalized care for the disabled isn't always appropriate for individual needs, dignity, and independence.

Normal people attempt to avoid assisting them in any way. They have a hard time surviving in the real world.
For someone who is deaf-mute and blind, communication can be a big challenge.

According to the Globe Health Organization, there are approximately 285 million visually impaired people in the world, 466 million people with hearing loss, and 1 million people who are deaf.

In this project, we propose a replacement system prototype that will assist people who are blind, deaf, or dumb, or who have a combination of those three disabilities.
To help these people's problems I have a three-module approach.

Sign Language Detection:
This module will detect sign language in real-time. It produces accurate words and can be used as an additional module in a video conferencing app so that dumb people can use it and communicate with others.

Text to Speech:
This module will convert text to human-understandable language, here English, in order that a deaf or a dumb person can type whatever he wants to talk to a traditional person and the rest of the work of chatting with the person are going to be done by the software UDSLD provided by us.

Text Scrapping:

This module is split into two parts i.e.,

- Taking the user's speech and turning it to text so that a deaf person may read what a hearing person is saying.
- The process of converting an audio file to text so that a deaf person may read what is being spoken in a recorded audio file.

1.1 Objectives

Following are the objectives of the paper:

- To construct a UDSLD to enable differently-abled people to function effectively.
- To create software that protects the user from falling into misunderstanding due to poor communication medium.
- To develop a personalized communication software for the user.
- To make the software efficient for both the user and the normal person also.
- To collect data from the user and after processing using these modules and send it to a normal person and vice versa so, it can perform actions accordingly.

2. PROBLEM STATEMENT

Communication has always been tough between a hard of hearing, silent, and visually handicapped man. Science and invention have made human existence addictive to comfort, but there is also an oppressed group of people who are struggling to find a creative approach to make the communication cycle easier for them.

Individuals who are dazed can speak freely in everyday language, however the hard of hearing imbecilic have their own manual-visual language. Sign language is the most common method for deaf and dumb people to communicate.

If the distance between them is greater, communication with deaf persons becomes more difficult. For example, if two deaf/dumb people are in close proximity, they will communicate using sign language; nevertheless, this approach is inefficient because both parties must have a thorough comprehension of sign language.

The majority of individuals, including the blind, are unable to understand sign language. If a person has all three disabilities, for example, if a blind person is also deaf-mute, he or she has no way of communicating. Deaf and dumb people may not be able to read the Braille script because blind individuals only recall the Braille script. They have difficulty communicating with one another.

This problem prompted us to conduct research on communicators who are blind, deaf, or mute. The long-term goal is to facilitate communication between outwardly impaired (i.e., dazed), hearing and discourse weakened (i.e., not too sharp) people on the one hand, and externally impeded, hearing and discourse weakened people on the other.
There are now no ways of communication between such persons, who unfortunately number in the millions in countries like India. By establishing a true-time system, our model presents a solution to wasteful communication between normal and impaired people.

3. METHODOLOGY

3.1 Dataset Collection

In this paper, we use the Kaggle data collection. The ASL alphabet dataset contains 87,000 images and 29 classes. 26 of these classes are the letters A-Z and therefore the other 3 are the signs for nothing, space, and delete. These 87,000 images were divided into 78,300 images that might be fed into the model as training data and eight,700 that might be used as validation data.

In addition to splitting the info into training and validation data, a generator was also wont to augment the images (rotate them, shift them sideways, etc.) in order that the info became less similar and therefore the model would need to generalize instead of just memorizing certain images.

3.2Algorithm

3.2.1 CNN

A convolutional neural network is a feed-forward neural network that is most commonly used to deconstruct visual images using a lattice-like topology to handle input. It's also known as "ConvNet." To detect and organize items in a picture, a convolutional neural network is used.

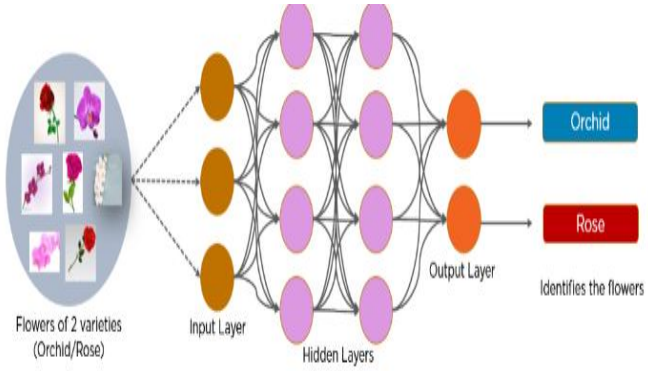

Figure 1: Working Of Algorithm

3.2.2CNN Layers

Different secret layers of a convolution neural network aid in the removal of data from a picture. In CNN, there are four important layers:

- Layer of convolution
- Layer of ReLU
- a layer for pooling
- Layer that is completely interconnected

CONVOLUTION LAYER:

This is the first part of the process of removing significant highlights from a photograph. The convolution action is carried out by a few channels in a convolution layer. Each image is regarded as a grid of pixel values.

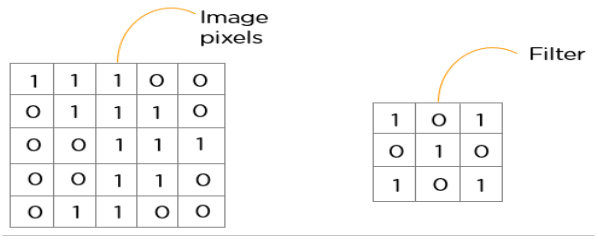

Figure 2: Working of Convolution Layer

ReLU LAYER:

The modified direct unit is denoted by ReLU. After extracting the component maps, the next step is to shift them to a ReLU layer.

ReLU performs a component insightful activity and resets all negative pixels to zero. It acquaints the organisation with non-linearity, and the result is a redressed include map. A ReLU work is depicted in the diagram below

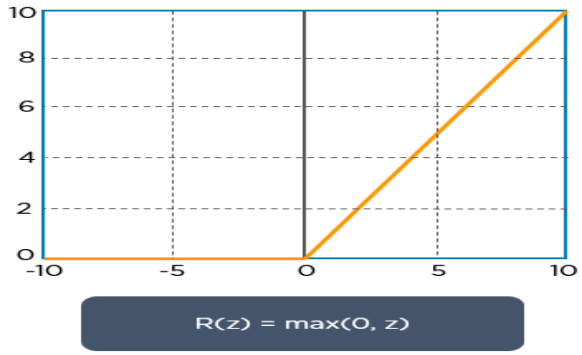

Figure 3: ReLU Layer Graph

POOLING LAYER:

Pooling is a down-testing operation that reduces the component map's dimensionality. To generate a pooled highlight map, the redressed include map is now passed via a pooling layer.

Distinct channels are used by the pooling layer to differentiate different parts of the image, such as edges, corners, body, plumes, eyes, and nose.

FULLY CONNECTED LAYER:

This Layer is used to get the final output of the shown image. This layer help in prediction of the output result.

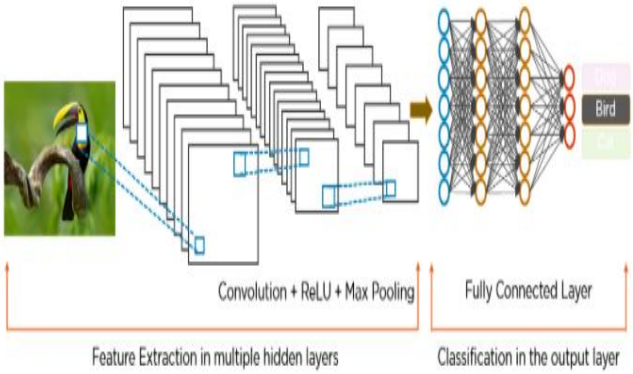

Figure 4: Fully Connected Layer

3.3Process

3.3.1Capturing Accurate Image

Capture Image and check for the shape and size of the image and after capturing the image next step in training model.

3.3.2 Training Model

To prepare the model, we will unfurl the information to make it accessible for preparing, testing, and approval purposes.
The Training Accuracy for the Model is 100% while test exactness for the model is 91%.

In the following stage, we will utilize Data Augmentation to take care of the issue of overfitting.

3.3.3Data Augmentation

There can be a few highlights/direction of pictures present in the test dataset that are not accessible in the preparation dataset. Also, Hence, our model can't recognize those examples. This is can be addressed by expanding the data.
Data Augmentation is a fundamental advance in preparing the neural organization. For instance, in the preparation dataset, we have hand indications of the correct hands yet in reality, we could get pictures from both right hands just as left hands. Data Augmentation permits us to make unanticipated data through Rotation, Flipping, Zooming, Cropping, Normalizing, and so on
Tensor flow gives an ImageDataGenerator work that expands data in memory on the stream without the need of adjusting nearby data. This additionally gives us space to attempt distinctive augmentation boundaries. We will Augment the data and split it into 80% preparing and 20% approval.

3.3.4Background Elimination

Since the pictures acquired are in RGB shading spaces, it turns out to be harder to section the hand signal dependent on the skin shading as it were. We, thusly, change the pictures in HSV color space. It is a model that parts the shade of a picture into 3 separate parts specifically: Hue, Saturation, and worth. HSV is an amazing asset to improve the solidness of the pictures by separating splendor from chromaticity. The Hue component is unaffected by any sort of brightening, shadows, and shadings and would thus be able to be considered for foundation evacuation. A track-bar having H going from 0 to 179, S going from 0-255, and V running from 0 to 255 are utilized to distinguish the hand signal and set the foundation to dark. The locale of the hand motion goes through expansion also, disintegration tasks with a circular piece. The primary picture is acquired subsequent to applying the 2 as shown in this Figure 5.

Figure 5: Image Background Elimination

3.3.5Segmentation

The primary picture is then changed to grayscale. However much this interaction will bring about the deficiency of shading in the locale of the skin motion, it will likewise improve the vigor of our framework to changes in lighting or brightening. Non-dark pixels in the changed picture are binarised while the others stay unaltered, subsequently dark.

The hand signal is fragmented initially by taking out every one of the joined segments in the picture and besides by letting just the part which is monstrously associated, for our situation is the hand motion. The edge is resized to a size of 64 by 64 pixels. Toward the finish of the division interaction, parallel pictures of size 64 by 64 are acquired where the region in white addresses the hand signal, and the dark hued territory is the rest.

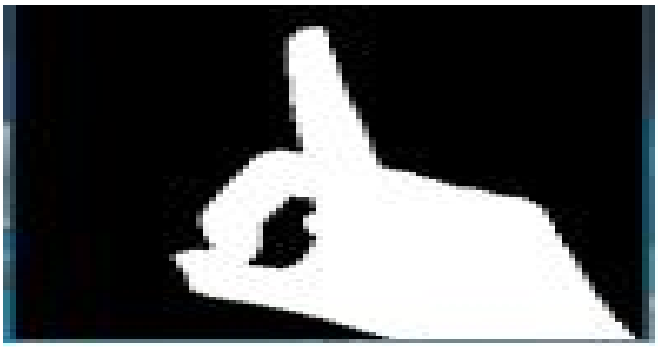

Figure 6: Image After Segmentation

3.3.6 Feature Extraction

Perhaps the most pivotal pieces of picture handling is to choose and separate significant highlights from a picture. Pictures when caught and put away as a dataset typically take up a ton of room as they are included a gigantic measure of information. Highlight extraction assists us with tackling this issue by diminishing the information subsequent to having removed the significant highlights consequently. It moreover adds to keeping up the precision of the classifier and works on its intricacy.

For our situation, the highlights discovered to be urgent are the double pixels of the pictures. Scaling the pictures to 64 pixels has driven us to get adequate highlights to successfully group the American Sign Language motions. Altogether, we have 4096 highlights, gotten subsequent to increasing 64 by 64 pixels.

3.4 Modules and It's Working

3.4.1 Sign language detection and converting them into sentences.

- The CNN is prepared on a custom dataset containing letter sets A-Y (barring J) of American Sign Language
- In this Gesture Recognize file
- This will start the webcam. Press C at that point place your hand inside the green box while playing out a signal.
- furthermore, you will get the letter to which the separate signal relates. Press Q to quit.
- And for the train your own model
- We are using an image capturing file
- Spot your hand in the green box and press C to begin catching the information.
- Now set up the paths in the Image_preprocessing.py file to preprocess the dataset.
- And for preprocessing the image
- Image_preprocessing.py file
- After preprocessing set up the path in the model.py file to get the preprocessed data for training.

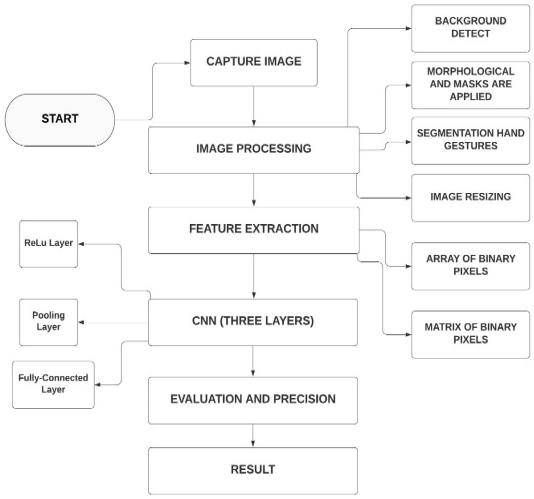

Figure 7: Flowchart for Internal Working of Module-1

3.4.2Text to speech

This module will change text over to human-reasonable language, here English, with the goal that a hard of hearing or a moronic individual can type anything he desires to address a typical individual and the remainder of crafted by addressing the individual will be finished by the product UDSLD given by us.

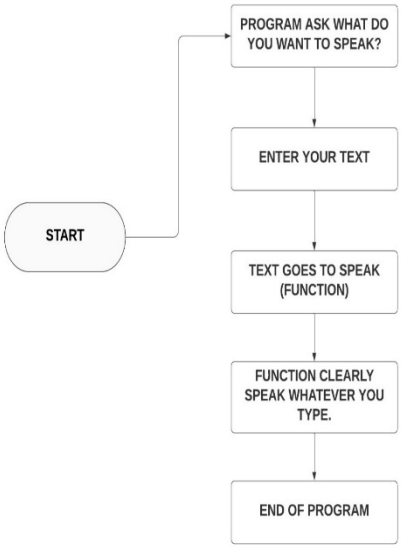

Figure 8: Flowchart for Internal Working of Module-2

3.4.3Voice to text scrapping

This module is isolated into two sections i.e.,

1. Deciphering voice from the client and changing it over to message so a hard of hearing individual can peruse whatever an ordinary individual is talking,

2. Deciphering from a sound document to message with the goal that a hard-of-hearing individual can peruse whatever is spoken in a recorded sound document.

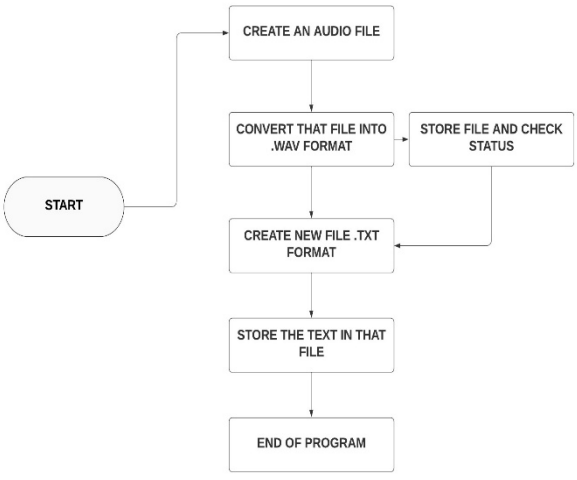

Figure 9: Flowchart for Internal Working of Module-3

4. RESULTS AND DISCUSSION

When using the ASL dataset, we observed 79.5 percent accuracy on letter gestures and 95 percent validation set accuracy on common sign gestures detection.

We found substantially lower accuracy measures on our self-generated dataset, which was to be expected given that our data was less uniform than that obtained in studio settings with superior equipment.

On letters of the alphabet, we saw 78 percent accuracy, and on common gestures, we saw 80 percent accuracy. In terms of temporal complexity, letter gestures took over 30 minutes to converge, while common motions took about 10 minutes.

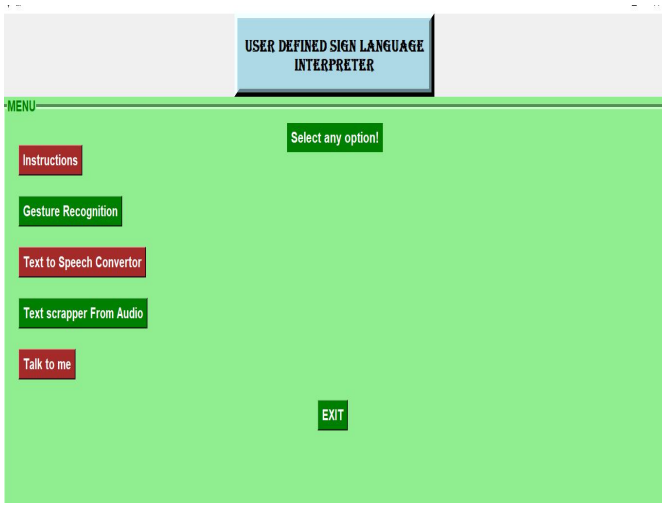

Figure 10: GUI

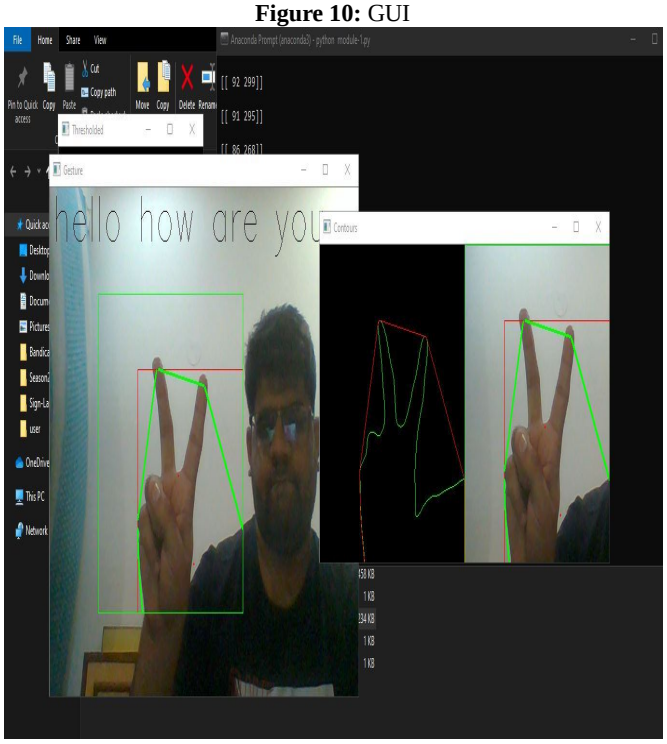

Figure 11: Gesture Representing Hello How are you

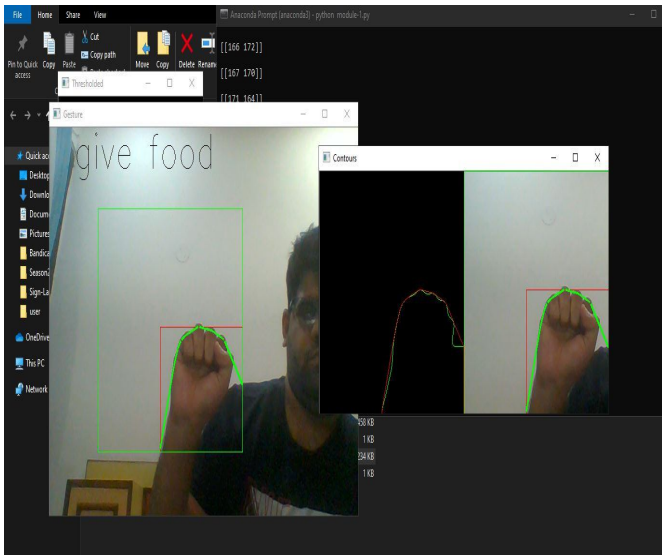

Figure 12: Gesture Representing Give Food

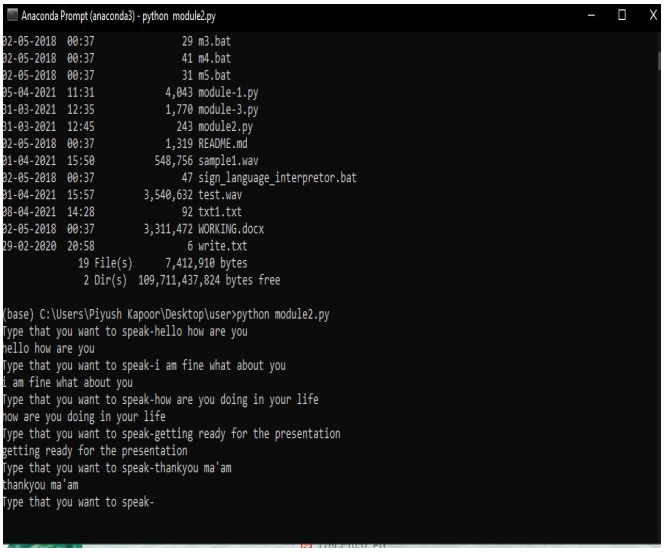

Figure 13: Text to Speech

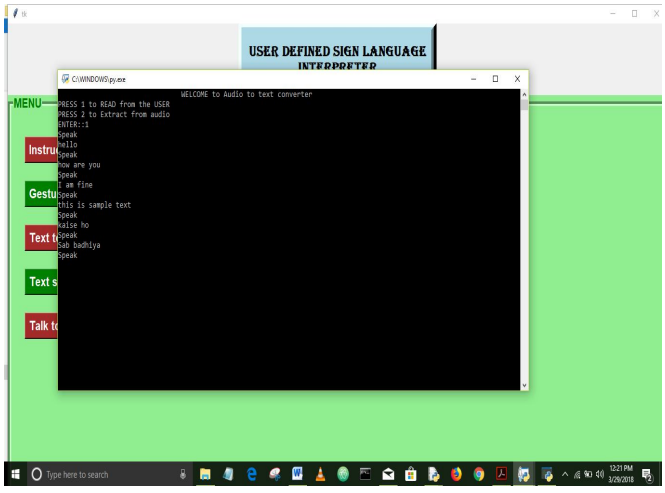

Figure 14: Text Scrapping Module

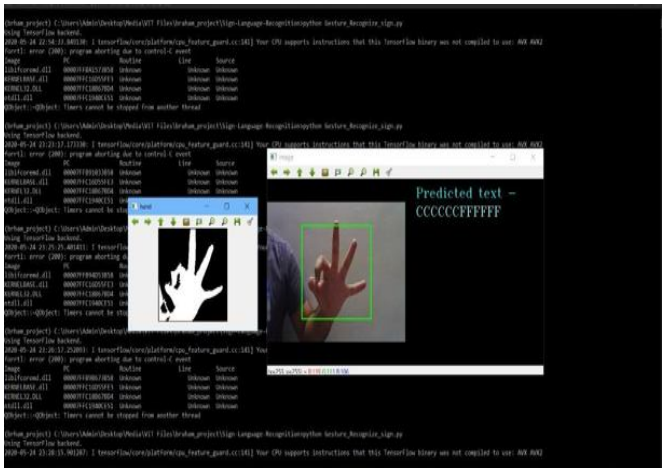

Figure 15: Result of Sign Capture

5. CONCLUSION

This work shows that convolutional neural organizations can be utilized to precisely perceive various indications of gesture-based communication, with clients and environmental factors not remembered for the preparation set. This speculation limit of CNNs in spatial-worldly information can add to the more extensive examination field on programmed gesture-based communication acknowledgment. We additionally effectively execute discourse to text and the other way around with exact outcomes.

6. FUTURE WORK

We anticipate utilizing more letters in order in our datasets and improve the model with the goal that it perceives more sequential highlights while simultaneously get high exactness.

We might likewise want to improve the framework by adding discourse acknowledgment so that visually impaired individuals can profit too

REFERENCES

1. Kumar, K. Thankachan and M. M. Dominic, "Sign language recognition," *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 422-428, doi: 10.1109/RAIT.2016.7507939.
2. K. Dixit and A. S. Jalal, "Automatic Indian Sign Language recognition system," *2013 3rd IEEE International Advance Computing Conference (IACC)*, 2013, pp. 883-887, doi: 10.1109/IAdCC.2013.6514343.
3. H. Muthu Mariappan and V. Gomathi, "Real-Time Recognition of Indian Sign Language," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862125.
4. D. Konstantinidis, K. Dimitropoulos and P. Daras, "SIGN LANGUAGE RECOGNITION BASED ON HAND AND BODY SKELETAL DATA," 2018 -

3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2018, pp. 1-4, doi: 10.1109/3DTV.2018.8478467.

5. M. Taskiran, M. Killioglu and N. Kahraman, "A Real-Time System for Recognition of American Sign Language by using Deep Learning," 2018 41st International Conference on Telecommunications and Signal Processing (TSP), 2018, pp. 1-5, doi: 10.1109/TSP.2018.8441304.
6. T. Karayılan and Ö. Kılıç, "Sign language recognition," 2017 International Conference on Computer Science and Engineering (UBMK), 2017, pp. 1122-1126, doi: 10.1109/UBMK.2017.8093509.
7. Yasir Niaz Khan, Ali Ahmad, Muhammad Usama, "Face Recognition Techniques: A Survey", *International Journal of Advanced Trends in Computer Science and Engineering*, Volume 10, No.2, March - April 2021.
8. Amitha Khan K H, Ankitha Chinnu Mathew, Ansu Raju, Navya Lekshmi M, Raveena R Maranagttu, Rani Saritha R,"**Speech Emotion Recognition Using Machine Learning**", *International Journal of Advanced Trends in Computer Science and Engineering*, Volume 10, No.2, March - April 2021.
9. Brashear H. Improving the efficacy of automated sign language practice tools. ACM SIGACCESS Accessibility and Computing - ASSETS no. 89, 2007, ACM New York, NY, USA.
10. Michaud L. N. and McCoy K. F. Modeling user language proficiency in a writing tutor for deaf learners of English. '99 Proceedings of a Symposium.