RESEARCH ARTICLE OPEN ACCESS

Sign Language Recognition using Convolutional Neural Networks

Adusumilli Yagna Gayathri[1], Suraka Maha Lakshmi Reddy[2]
1(Computer science & Engineering, G. Narayanamma Institute of Technology and Science for Women, Hyderabad, Email: gayathri.nov17@gmail.com)
2(Computer science & Engineering, G. Narayanamma Institute of Technology and Science for Women, Hyderabad, Email: mahasuraka1977@gmail.com)

Abstract:

Hand gesture interpretation is an appealing research area with numerous applications, including video games and telesurgery. Another important application of hand gesture recognition is the translation of sign language for non-verbal communication. It is most commonly used by deaf and dumb people who have hearing or speech problems to communicate among themselves or with physically enabled people. It is a successful way of communication among verbally and aurally impaired humans.

The primitives of complex expressions in sign language are the configuration of the fingers, the orientation of the hand, and the relative positioning of the hand in terms of the body. The proliferation of touchless apps and the fast rise of the hearing-impaired population have enhanced the necessity of hand gesture recognition. However, while developing an efficient recognition system one needs to overcome the challenges of hand segmentation, local hand shape representation, global body configuration representation, and gesture sequence modeling.

A platform that helps to communicate with and for deaf and dumb people and makes the communication possible by conveying the message of one person to the other using multiple deep learning architectures and classification techniques(CNN) for hand segmentation, local and global feature representations, and sequence feature globalization and recognition. The gestures and the voice message will be transcripted to text or voice messages according to the requirement of the user. The vision-based framework can be developed to allow the users to interact with the opposite person through human gestures. The hand gestures of the opposite person will be identified using image processing frameworks. Those gestures are recognized and the particular messages are predicted. The messages are sent to that person in the form of text or audio. In this way, communication between two people can be made simpler.

Keywords **— Gesture, sign language, Segmentation, vision-based, communication, CNN.**

I. INTRODUCTION

According to a recent survey by the WHO (World Health Organization), around 5% of the population in the world, that is over 460 million individuals worldwide have a hearing problem. But, there are less than 1000 certified interpreters who can understand and translate sign language. Hence, there is a need to address the huge demand for interpreters in various educational and working sectors. To satiate this need and solve the problem it is possible to design an efficient portable assistive device by employing technological advancements in computing devices, machine learning, and the Internet of Things.

II. OBJECTIVES

To create an effective sign language recognition model:

- Create a database for various phrases of ASL(American Sign Language)[2]
- To recognize hand gestures made in real-time using the above-created database.
- To use American sign language as a communication medium between the aurally, and verbally impaired and others.
- To create an interface to communicate with hearing and verbally impaired individuals, this interface takes input as words and displays images of respective signs.

III. METHODOLOGY

Machine learning is used nowadays in all aspects of life, and this can also be used to provide a solution to solve the problems encored by disabled people. The only means for conveying any message by hearing impaired people is sign language. In order to establish communication between the specially-abled and others is to make use of machine learning algorithms, to recognize the signs shown in words. [2][4][5]

Android mobile applications are the most common way of developing user interface platforms. In order to enable a normal person to talk to a specially-abled person with hearing imparity, Android Studio can be used to build an android application.

A. Module 1: Gesture Recognition

This deals with the conversion of sign language into words. The module is further divided into four modules, each dealing with different topics.

- Module 1.1: In order to identify any sign, the first step is to create a sign and later train the model such that the sign is recognized by the model. The first step in the creation of gestures takes place in this module. [2]

B. Module 2: Text to Speech Conversion

This is used to convert text to speech which is the output of the captured or recorded input by the user.

- The text-to-speech (TTS) is the process of converting words into a vocal audio form. The program, tool, or software takes an input text from the user, and using methods of natural language processing understands the linguistics of the language being used, and performs logical inference on the text. This processed text is passed into the next block where digital signal processing is performed on the processed text[7]. Using many algorithms and transformations this processed text is finally converted into a speech format. This entire process involves the synthesizing of speech. Below is a simple block diagram to understand the same.
- gTTS (Google Text-to-Speech), a Python library and CLI tool to interface with Google Translates text-to-speech API.[6]

C. Module 3: Android Application

The creation of the Android Application is done in this module. This module has two sub-modules.

- Module 3.1: This includes front and back-end development of the application where the growth of the app is observed using the android studio. Thus developed frontend is integrated with the backend which resulted in a full functioning application that is used to capture the hand gestures of a person being recorded and also allows the app to redirect to the gallery where the pictures/videos that have been previously taken can be uploaded.

- Module 3.2: The application is later merged with module 1, where the text is converted into speech adding to its features. So, when the video is recorded or uploaded the software splits the video into a set of frames and the relevant gestures are recognized, and the identified hand movements are translated into text which can be converted into speech.

IV. IMPLEMENTATION

A. *Data gathering*

- **Data Preparation**

Video data set was made to train the model. (no./phrases) _ number of phrase signs from American Standard Language were considered for this project. A folder containing 30 videos of each phrase was created, where each video taken was 30 frames in length.

- **Training**

The collected data sets are used as an input to train the machine in order to recognize signs from the given video data(module 2) and after identifying a sign it gets converted into text and speech(module 1) then this data is shown on the Android application(module 3)

B. *Algorithm: Convolutional Neural Network*

- In module 1 where the signs are converted into letters and words, Convolution Neural Network (CNN) is created using Keras. CNNs play a very important role in training the data involving images. CNNs apply a series of filters to the raw pixels data of an image to extract and learn higher-level features, which the model can then use for classification. CNNs typically contain three important components, namely convolutional layers, pooling layers, and dense layers [1].
- Convolutional layers apply a specified number of convolutional filters to images. The layer performs a set of mathematical functions called an activation function to produce a single value in the output feature map, for each sub-region. The most commonly used activation function is the ReLU function also called rectified linear unit, which helps in introducing non-linear ties into the model. The function is defined as f(x) = x+ = max (0, x) , where x is input to a neuron [3].
- The Pooling layer down samples the image data extracted by the above layer to reduce the dimensionality of the feature map and to decrease the processing time. A commonly used pooling algorithm is a max pool, which extracts sub-regions and keeps only max values by discarding the rest of the details.
- The dense layer performs classification on features extracted from the above two layers. In the dense layer, every node is connected to any other node in the preceding layer.
- Typically, a CNN is a collection of convolutional modules that perform feature extraction. Each module consists of a convolutional layer followed by a pooling layer. The last convolutional module is followed by one or more dense layers that perform classification. The final dense layer in a CNN contains a single node for each target class in the model (all the possible classes the model may predict), with a softmax activation function to generate a value between 0–1 for each node (the sum of all these softmax values is equal to 1). Softmax values can be interpreted, for a given image as relative measurements of how likely it is that the image falls into each target class [5].
- A summary of CNN working is explained pictorially in Fig.1

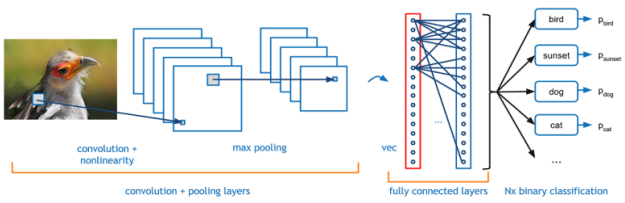

Fig.1: CNN working model

The architecture for the project is as follows:

1. Convolutional Layer #1: Applies 16 2x2 filters (extracting 2x2-pixel subregions), with ReLU activation function.
2. Pooling Layer #1: Performs max pooling with a 2x2 filter and stride of 2 (which specifies that pooled regions do not overlap).[8]
3. Convolutional Layer #2: Applies 32 3x3 filters, with ReLU activation function.
4. Pooling Layer #2: Again, performs max pooling with a 3x3 filter and stride of 3.
5. Convolutional Layer #3: Applies 64 5x5 filters, with ReLU activation function.
6. Pooling Layer #3: Again, performs max pooling with a 5x5 filter and stride of 5.
7. Dense Layer #1: 128 neurons, with a dropout regularization rate of 0.2 (probability of 0.2 that any given element will be dropped during training)

V. RESULTS AND DISCUSSIONS

A. Datasets and Performance Measures

The dataset of this project is raw videos of different English phrases in sign language. All the phrases were stored in a folder, with each phrase having a folder with 30 videos divided into 30 frames. This data is collected using OpenCV. Using media pipe detections were made and styled landmarks were drawn. While training the model categorical accuracy and losses were measured on a tensor board and are graphically displayed in the following figures.

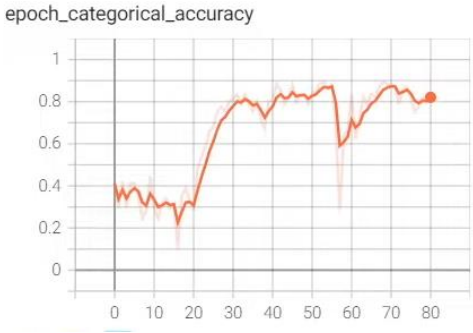

Fig.2: Epoch categorical accuracy

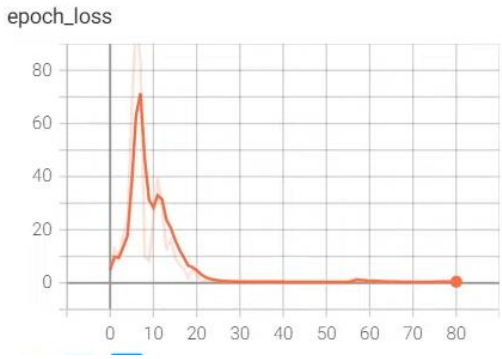

Fig.3: Epoch loss

The confusion matrix and accuracy score is used to measure the performance of the model built.

B. Recognizing Gestures:

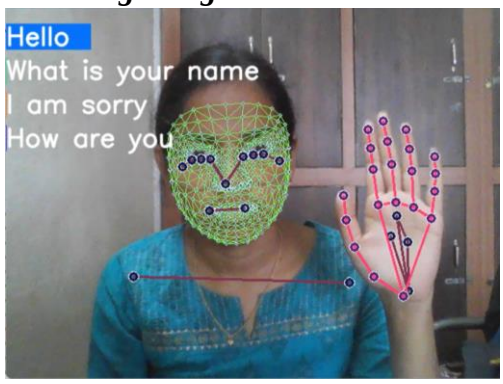

Fig.4: Gesture recognition_1

Fig.5: Gesture recognition_2

Fig.6: Gesture recognition_3

Fig.7: Gesture recognition_4

C. *Text to Speech Conversion*

gTTS (Google Text-to-Speech), a Python library and CLI tool are used to interface with Google Translate's text-to-speech API, this API was used to convert text to speech and the results are as shown in the following diagrams.[6]

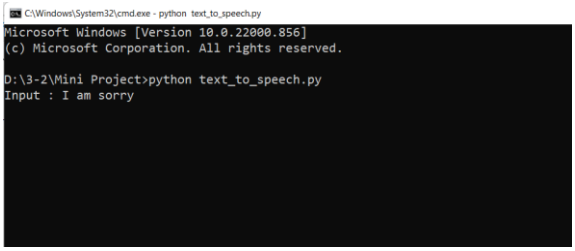

Fig.8: Text-to-speech conversion

Fig.9: Output of text-to-speech conversion

D. *Android Application*

Once the app is opened you can either choose to log in, in case the account already exists, create an account, or choose to continue as a guest as shown.

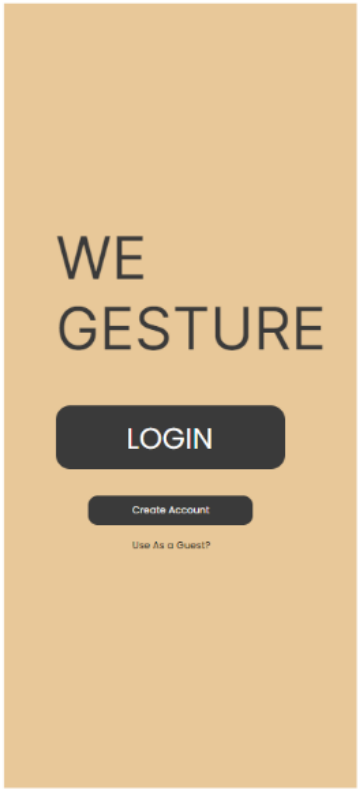

Fig.10: Home page of We Gesture

VI. CONCLUSIONS

In this work, all the sections namely, the gesture recognition, text-to-speech conversion, and sAndroid application have been successfully implemented sufficing the objectives of the work. There is a scope for future enhancements such that it can take new gestures in any sign language and store them in the database as the current model is only limited to American sign language(ASL). The further extension to this model is to make it a two-way communication platform, thus not only converting sign language to text & speech but also vice-versa.

REFERENCES

1.http://mospi.nic.in/sites/default/files publication_reportsDisabled_persons_in_India_2016. pdf

2.http://www.lifeprint.com/asl101/pageslayout/concepts.html
3.https://developers.google.commachine-learning/problem-framing/
4.https://github.com/kevinam99capturing-images-fromwebcam-using-opencvpython
5.https://jayrambhia.wordpress.com2012/0510/capture-images-and-videofromcamera-in-opencv-2-3-1/
6.https://gtts.readthedocs.io/en/latest/
7.https://towardsdatascience.comhow-to-getstarted-with-google-text-tospeech-using-python-485e43d1d544
8.https://docs.w3cub.com/tensorflow~guidetutorials/layers.html