A HAND GESTURE - BASED SOLUTION FOR THE SPEECH AND HEARING IMPAIRED USING DEEP LEARNING AND COMPUTER VISION

Pranav.P, Samuel Jude.S, Saurabh Singh,
Roshan.M
Electronics & Communication Engineering
Rajalakshmi Engineering College
Chennai, Tamilnadu, India
E-mail: pranavprakashp@gmail.com

Mrs.Saranya.J , Assistant Professor
Electronics & Communication Engineering
Rajalakshmi Engineering College
Chennai, Tamilnadu, India
E-mail: saranyaa23@gmail.com

Abstract—**Hand gesture recognition based solution addresses the community of the society who are physically challenged in terms of speech and hearing. There are approximately 466 million people in the world with hearing impairment and around 16 million with speech impairment. This serves as one of the serious problems that is still prevailing in the society which is desperately in need to be addressed as it encourages the physically challenged society to express their thoughts,views and ideas to the world. Our work addresses this group of society with a combined application of DL and Computer Vision .The concept implemented is the idea of developing a model which understands the sign language of the speech impaired person or the voice of the person in case of hearing impaired to appropriate audio outputs and picture outputs respectively aiding the needs of the both the speech and the hearing impaired with a greater accuracy which includes necessary analysis , preprocessing of images ,Usage of libraries for audio recording and the use of trained CNN model using a self created dataset which is specific to this application.**

Keywords- CNN, DL, COMPUTER VISION

I. INTRODUCTION

Deaf people use sign languages to communicate with other people in the community. Although the sign language is known to hearing-impaired people due to its widespread use among them, it is not known much by other people. In the setup developed in [1],the convolutional neural network was trained by using ASL dataset and 100% test accuracy was obtained. The problem that persisted with this paper is that it did not convert text into speech and also did not provide a means of communication for the hearing impaired.

System developed in [2], Basic Component Analysis Is used in the extraction of the feature vector in work done by Mahmoud Zaki and Samir Shaheen in 2011 and Hidden-Markov Model is used as the classifier

System developed in [3] K-Nearest Neighbors (k-NN) Classifier was used in American Sign language recognition by Dewinta Aryanie and Yaya Heriadi in 2015.

System developed in [4], the study of Joshi and his colleagues in 2017, American sign language translator was realized by using edge detection and cross-correlation methodologies.

System developed in [5] done by Adhitya Das and his colleagues used an Inception v3 model on a custom dataset and obtained a validation accuracy of 90%

II. METHODOLOGY

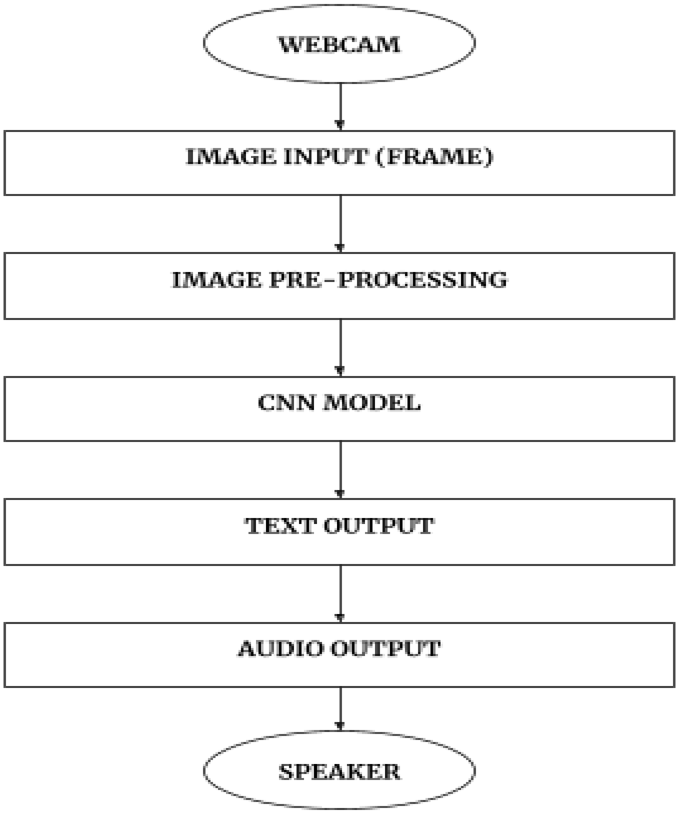


Fig 1.Gestures To Speech

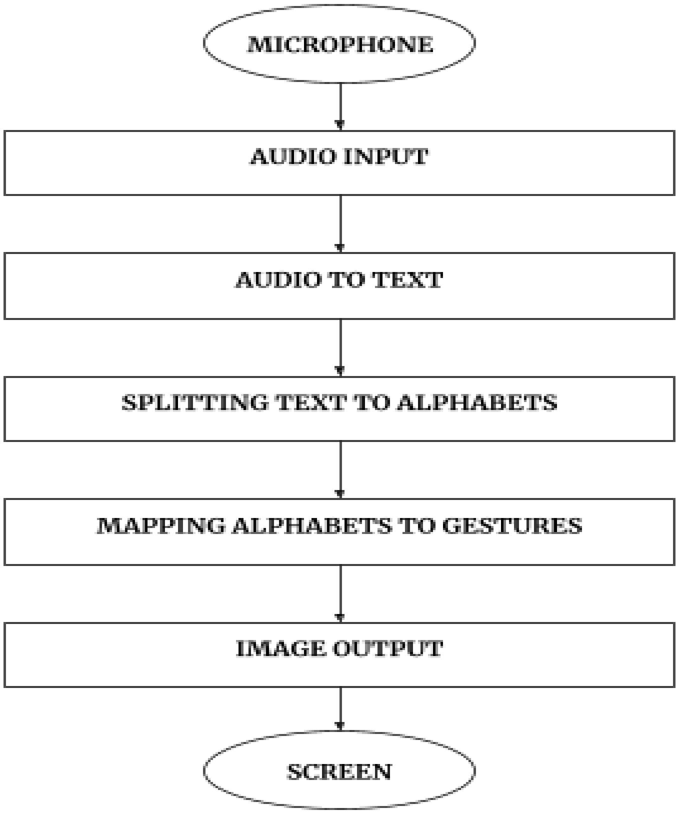


Fig 2. Speech To Gestures

A. DATASET

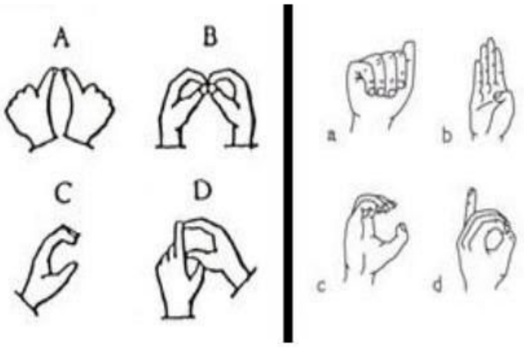


Fig 3. American Sign Language & Indian Sign Language Sample Gestures[5]

Fig 1 shows the illustrations of both ASL and ISL datasets . The ASL Dataset (Fig.2 left) and the ISL Dataset(Fig.2 right) had less data samples and had few two handed gestures and dynamic gestures, we chose to create our own dataset. The dataset we created has 27(alphabets + space) classes with 5000 images per class.

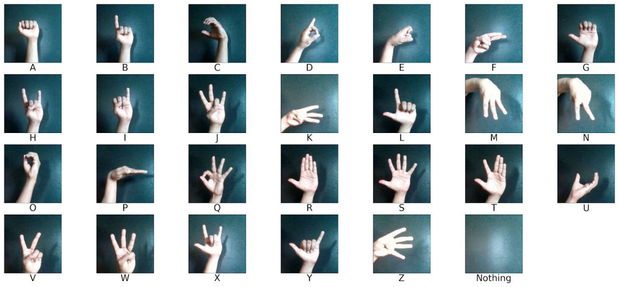


Fig 4. Custom Dataset

The images are then pre-processed by using Canny Edge Detection to make the gestures easily distinguishable and thereby increasing the accuracy of the model.

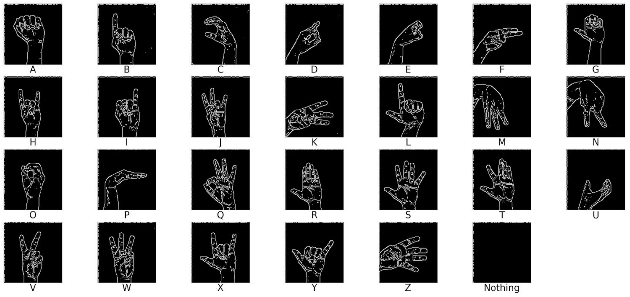


Fig 5. Custom Dataset After Pre-processing (Canny)

B. COMPUTER VISION

Computer Vision is perhaps the most intriguing and fascinating concept in artificial intelligence. Computer Vision is an interdisciplinary field that deals with how computers or any software can learn a high-level understanding of the visualizations in the surroundings. After obtaining this conceptual perspective, it can be useful to automate tasks or perform the desired action.[6]

The tasks that are obvious to the human brain are not so intuitive to the computers as they need to be trained specifically on these jobs to produce effective results. This process involves complicated steps like acquiring the data from the real world, processing the acquired data in a suitable format, analyzing the processed images, and finally teaching and training the model to perform the complex task with very high accuracy.[6]

OpenCV module is by far the best module for the execution of complex machine learning, deep learning, and computer vision tasks. It offers simplicity and high standards for the analysis and performance of the models being built. It is an open-source library and can be integrated with other python modules such as NumPy to accomplish complicated real-time applications. It is supported for a wide range of programming languages and runs remarkably on most platforms such as Windows, Linux, and MacOS.[6]

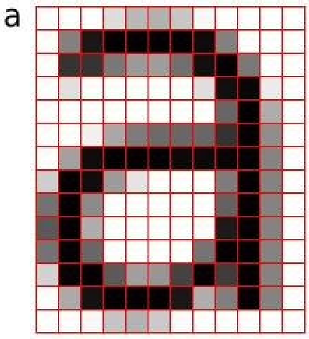
C. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Network(CNN or ConvNet)is a class of deep neural networks which is mostly used to do image recognition, image classification, object detection, etc.[7]

The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm — a Convolutional Neural Network.[7]

Image classification is the task of taking an input image and outputting a class or a probability of classes that best describes the image. In CNN, we take an image as an input, assign importance to its various aspects/features in the image and be able to differentiate one from another. The preprocessing required in CNN is much lesser as compared to other classification algorithms.[7]

Computers can not see things as we do, for computers image is nothing but a matrix.A CNN typically has three layers: a convolutional layer, pooling layer, and fully connected layer.[7]



1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	02	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	07	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Fig 6. Matrix representation of picture[7]

The convolution layer is the core building block of CNN. It carries the main portion of the network's computational load.The main objective of convolution is to extract features such as edges, colours, corners from the input. As we go deeper inside the network, the network starts identifying more complex features such as shapes,digits, face parts as well. [7]

Pooling layer is solely to decrease the computational power required to process the data. It is done by decreasing the dimensions of the featured matrix even more. In this layer, we try to extract the dominant features from a restricted amount of neighborhood. [7]

Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer.[8]

Fully Connected Layer is simply, feed forward neural networks. Fully Connected Layers form the last few layers in the network. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer. [9]

A lot of theory and mathematical machines behind the classical ML (regression, support vector machines, etc.) were developed with linear models in mind. However, practical real-life problems are often nonlinear in nature and, therefore, cannot be effectively solved using those ML methods. The activation function is the non-linear function that we apply over the output data coming out of a particular layer of neurons before it propagates as the input to the next layer. There are many types of activation functions like ReLu, Softmax, Leaky-ReLu,Sigmoid, Tanh, etc..[10]

Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate to reduce the losses. Optimizers are used to solve optimization problems by minimizing the function. There are different types of optimizers like Adam, Adagrad, Nadam, etc..[11]

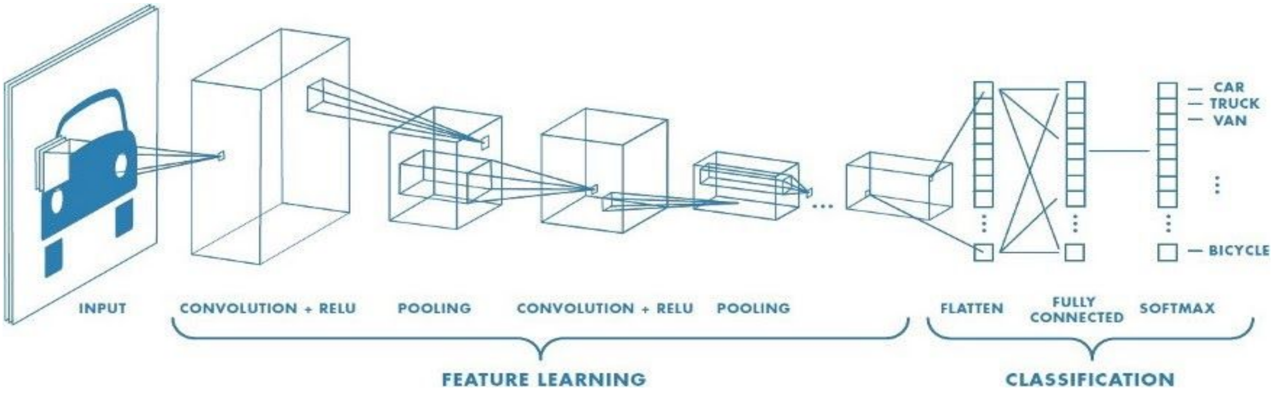


Fig 7.CNN Layers [7]

Our model has 4 Convolutional Layers and 2 Fully Connected Layers each with ReLu Activation layer , Batch Normalisation layer, MaxPooling layer and Dropout Layer. The convolution layer output is flattened before sending into the fully connected layer. The output layer has 27 possible outcomes and has a Softmax Activation Layer. The loss monitored is Categorical Cross Entropy and the Optimizer used is Adam.

ReLU is a non-linear activation function that was first popularized in the context of a convolution neural network (CNN). If the input is positive then the function would output the value itself, if the input is negative the output would be zero. [10]

The Softmax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. That is, softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels.[12]

Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.[12]

Batch normalization is a layer that allows every layer of the network to do learning more independently. It is used to normalize the output of the previous layers. The activations scale the input layer in normalization. Using batch normalization learning becomes efficient also it can be used as regularization to avoid overfitting of the model.[13]

Dropouts are the regularization technique that is used to prevent overfitting in the model. Dropouts are added to randomly switching some percentage of neurons of the network. When the neurons are switched off the incoming and outgoing connection to those neurons is also switched off. This is done to enhance the learning of the model.[13]

Categorical cross entropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one.Formally, it is designed to quantify the difference between two probability distributions.

D. SPEECH & TEXT CONVERSION

The CNN model gives us an output in text format which is then converted into audio. This conversion is done using the python library pyttsx3. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3.

The audio input is converted into text by using the python library SpeechRecognition. It can support various API's like Google Cloud Speech API , Wit.Ai , etc.. The python library PyAudio is required if and only if you want to use microphone input.

E. GRAPHIC USER INTERFACE

Kivy is a graphical user interface open source Python library that allows you to develop multi-platform applications on Windows, macOS, Android, iOS, Linux, and Raspberry-Pi. In addition to the regular mouse and keyboard inputs, it also supports multitouch events. The applications made using Kivy will be similar across all the platforms but it also means that the applications feel or look will differ from any native application.

III. RESULTS

The CNN model was trained for using different optimizers namely Adam, Nadam and RMSprop. Their validation accuracies are shown below in Fig 8. and validation losses are shown in Fig. 9.

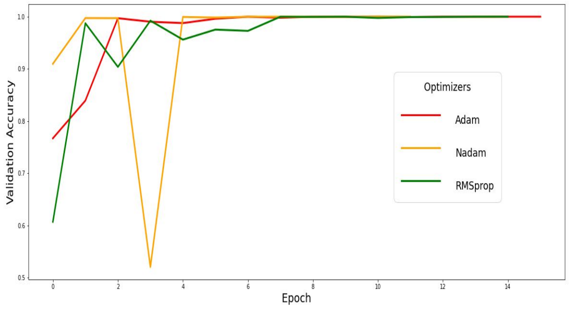


Fig 8. Validation Accuracies For Different Optimizers

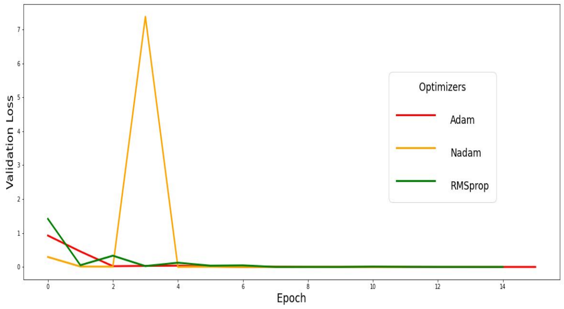


Fig 9. Validation Losses For Different Optimizers

We can see all the three optimizers give us 100% test validation accuracies. RMSprop took longer convergence time to give us the best results and Nadam had the problem of overfitting initially before giving us the best result, Therefore we chose the Adam optimizer since it had the least convergence time and suffered less overfitting.

The training and validation accuracy and loss of model using Adam optimizer is shown below in Fig 10 and Fig 11. respectively.

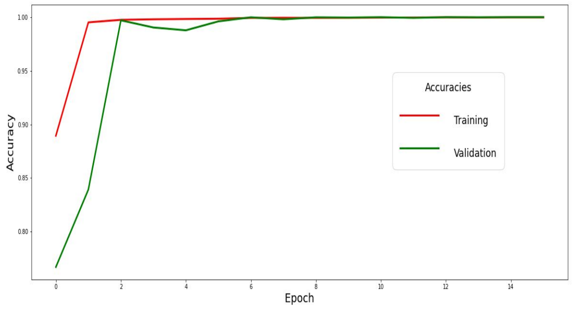


Fig 10, Accuracy v/s Epochs for Adam

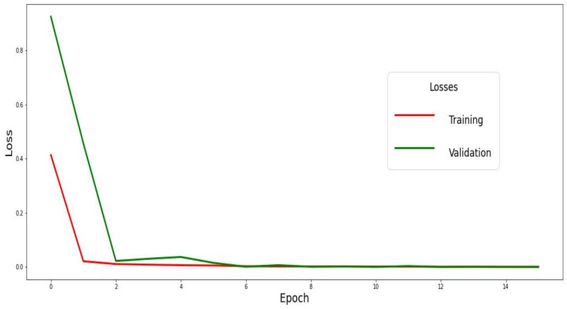
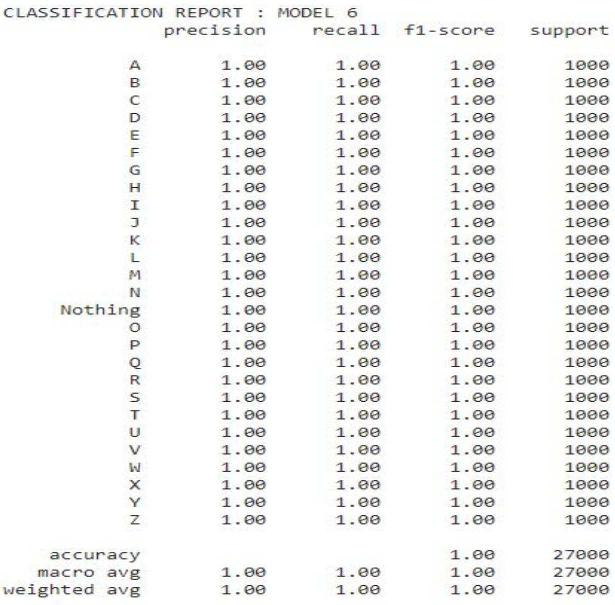


Fig 11, Loss v/s Epochs for Adam

Best validation accuracy and loss for Adam optimizer is obtained around the 6th epoch.



```
CLASSIFICATION REPORT : MODEL 6
              precision    recall  f1-score   support

           A       1.00      1.00      1.00      1000
           B       1.00      1.00      1.00      1000
           C       1.00      1.00      1.00      1000
           D       1.00      1.00      1.00      1000
           E       1.00      1.00      1.00      1000
           F       1.00      1.00      1.00      1000
           G       1.00      1.00      1.00      1000
           H       1.00      1.00      1.00      1000
           I       1.00      1.00      1.00      1000
           J       1.00      1.00      1.00      1000
           K       1.00      1.00      1.00      1000
           L       1.00      1.00      1.00      1000
           M       1.00      1.00      1.00      1000
           N       1.00      1.00      1.00      1000
     Nothing       1.00      1.00      1.00      1000
           O       1.00      1.00      1.00      1000
           P       1.00      1.00      1.00      1000
           Q       1.00      1.00      1.00      1000
           R       1.00      1.00      1.00      1000
           S       1.00      1.00      1.00      1000
           T       1.00      1.00      1.00      1000
           U       1.00      1.00      1.00      1000
           V       1.00      1.00      1.00      1000
           W       1.00      1.00      1.00      1000
           X       1.00      1.00      1.00      1000
           Y       1.00      1.00      1.00      1000
           Z       1.00      1.00      1.00      1000

    accuracy                           1.00     27000
   macro avg       1.00      1.00      1.00     27000
weighted avg       1.00      1.00      1.00     27000
```

Fig 12. Classification Report

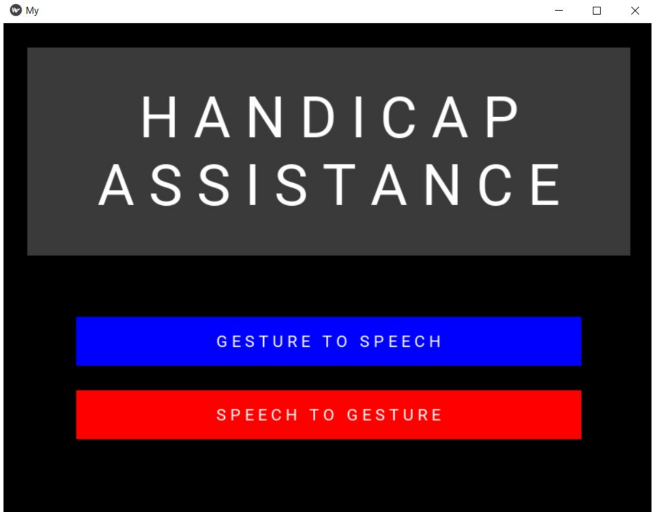


Fig 13. App GUI

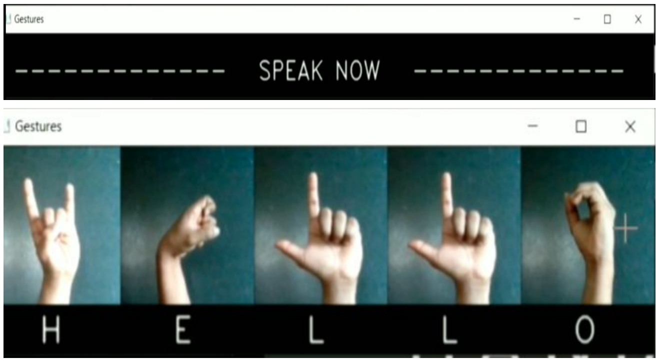


Fig 14. Speech To Gestures

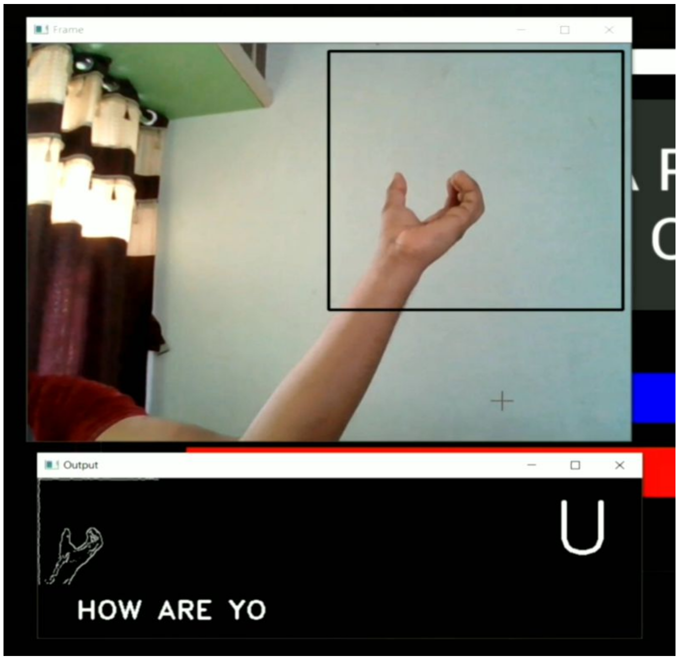


Fig 15. Gestures To Speech

IV. Conclusion

Using OpenCV for data collection and image processing, Convolutional Neural Network Model for sign language classification and python libraries for speech to text and text to speech conversion, we were successfully able to create a communication assistance system for the speech and hearing impaired. The CNN model validation accuracy is 100% and its real time accuracy is 93%.

References

[1] 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), Athens, 2018, pp. 1-5, doi: 10.1109/TSP.2018.8441304.

[2] M. M. Zaki, S. I. Shaheen, “Sign language recognition using a combination of new vision based features, ” Pattern Recognition Letters, vol. 32,pp. 572–577, 2011.

[3] D. Aryanie, Y. Heryadi, “American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier,” in Proc. 3rd International Conference on Information and Communication Technology

[4] A. Joshi, H. Sierra, E. Arzuaga, “American sign language translation using edge detection and cross correlation,” in Proc. IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena,Colombia, 2017, pp. 1–6.

[5] A. Das, S. Gawde, K. Suratwala and D. Kalbande, "Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images," 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, 2018, pp. 1-6, doi: 10.1109/ICSCET.2018.8537248.

[6] K. Barath (Oct 2020) OpenCV: Complete Beginners Guide To Master Basics Of Computer Vision With Codes![Online]. Available: https://towardsdatascience.com/opencv-complete-beginners-guide-to-master-the-basics-of-computer-vision-with-code-4a1cd0c687f9

[7] Manan Parekh (July 2019) A Brief Guide to Convolutional Neural Network(CNN)[Online] Available: https://medium.com/nybles/a-brief-guide-to-convolutional-neural-network-cnn-642f47e88ed4

[8] Jiwon Jeong (Jan 2019) The Most Intuitive and Easiest Guide for Convolutional Neural Network [Online]

Available: https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480#:~:text=Flattening%20is%20converting%20the%20data,called%20a%20fully%2Dconnected%20layer.

[9] Arunava (Dec 2018) Convolutional Neural Network [Online] Available: https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05#:~:text=Fully%20Connected%20Layer%20is%20simply,into%20the%20fully%20connected%20layer.

[10] Kevin Vu (Dec 2019) Activation Functions and Optimizers for Deep Learning Models [Online] Available: https://dzone.com/articles/activation-functions-and-optimizers-for-deep-learn#:~:text=ReLU%20is%20a%20non%2Dlinear,the%20output%20would%20be%20zero.

[11] Nagesh Singh Chauhan (Dec 2020) Optimization Algorithms in Neural Networks [Online] Available: https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html#:~:text=Optimizers%20are%20algorithms%20or%20methods,problems%20by%20minimizing%20the%20function.

[12] Jason Brownlee (Oct 2020) Softmax Activation Function with Python [Online] Available: https://machinelearningmastery.com/softmax-activation-function-with-python/#:~:text=The%20softmax%20function%20is%20used%20as%20the%20activation%20function%20in,more%20than%20two%20class%20labels.

[13] Rohit Dwivedi (Selp 2020) Everything You Should Know About Dropouts And BatchNormalization In CNN [Online] Available: https://analyticsindiamag.com/everything-you-should-know-about-dropouts-and-batchnormalization-in-cnn/

[14] Evgeny A. Smirnov et al (2014, Dec). "Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks"

[15] Anup Kumar et al, "Sign Language Recognition", in Recent Advances in Information Technology (RAIT), 2016 3rd International Conference, 2016. doi: 10.1109/RAIT.2016.7507939