REAL TIME SIGN LANGUAGE TRANSLATION

Kartikay Sapra(2018238), Shivang Saigal(2018310)

B.Tech CSAM, B.Tech. CSD
Indraprastha Institute of Information Technology, Delhi

ABSTRACT

Sign language is one of the oldest and most natural forms of communication. Nowadays, most people do not know sign language, and interpreters are very difficult to find. We have come up with a real-time method to interpret finger spelling based on American sign language. We have used neural networks in our working model. In our approach, the hand images is passed through a filtering algorithm. After applying the filter, a classifier predicts the type of gesture and the associated letter.

***Index Terms*—** blur, threshold, grayscale, Canny Edge, convolution

1. INTRODUCTION

Communication is an essential survival tool. More than 5 percent of the world's population is deaf, sign language is primarily used by these differently abled people to communicate. Sign language is complex in nature and a common man finds it difficult to interpret.

The above stated problem translates into a language barrier for the deaf and the people associated with them. Tradition script based language is based on hearing and pronunciation of words and syllables which is not possible in the case of deafness.

On the other hand, generally, very few people have experience with sign language. This causes multiple problems in physical message exchange.

The vision of the project is to develop a real-time conversion tool to bridge the gap between the differently abled deaf and the remaining population. The tool would cross-translate between American Sign Language and English.

The development process involves generating a dataset for the American Sign Language gestures. The dataset is processed using Digital Image Processing techniques like grayscaling, filtering(Gaussian Blur), thresh-holding and other image enhancement techniques, corresponding data is trained using Convolutional Neural Networks(CNN).

2. LITERATURE SURVEY

- **Text to ISL(Indian Sign Language):** This paper talks about how text can be converted into Indian Sign Language using image processing/ machine language. The ISL syntax was developed using Lexical Functional Grammar f-structure. The ISL output was displayed using pre-recorded ISL videos [1].
- **Real-time Conversion of Sign Language to Text and Speech:** This paper showcases the performance of different techniques that are all been used for the conversion of sign language to text/speech format. It talks about converting real-time American sign language to text/speech in an android application [2].
- **Korean sign language recognition based on convolution:** Developing a convolution-based neural network for Korean sign language conversion. Korean sign language videos were collected and then converted into images. Experiments were conducted to check the accuracy of this model [3].
- **This paper talks about developing an HMM:** based hand gesture recognition system. It recognizes hand gestures and converts them to Tamil spoken language. This model follows a glove based approach [4].

3. METHODOLOGY

Python libraries/frameworks used: *keras, tensorflow, cv2, PIL, matplotlib, pyttsx3, hunspell, tkinter, matplotlib, numpy, scikit-learn*

The process involved generating a dataset using Image Processing and then training a Convolutional Neural Network(CNN)

3.1. Image Processing

The following steps were followed:

3.1.1. ***Feature Selection***

Original images of hand gestures had multiple features which did not aid the classification process. The gestures are primarily distinguished by the position of fingers and palm. The RGB color channels were not useful.

Grayscaling: The RGB images were converted to grayscale images. Common grayscaling technique is to take the average, (R + G + B)/3 for each pixel. However, there is still a lot of extra information, we need to find the optimal number of features. The outlines of the parts of hand are desired.

3.1.2. Approach 1: Canny Edge Detection

This technique was used to get the outline. Canny Edge Detection Involves:

- Noise Reduction using Gaussian Blur
- Calculating Magnitude and Gradient of the Image using Sobel operators
- Non-max suppression: After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge.
- Hysteresis Thresholding decides which are all edges are really edges and which are not and then subtracting 255 from every pixel.

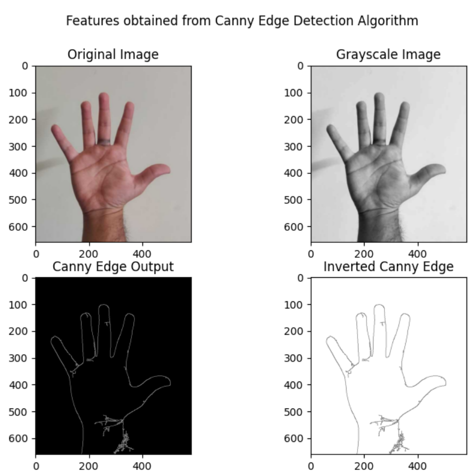

Fig. 1. Canny Edge Detection(Light Background)

3.1.3. Approach 2: Custom Algorithm

This technique was used to get the outline of the hand as well as palm lines:

- Noise Reduction using Gaussian Blur
- In this, the algorithm calculate the threshold for a small regions of the image. So we get different thresholds for different regions of the same image and it gives us better results for images with varying illumination.

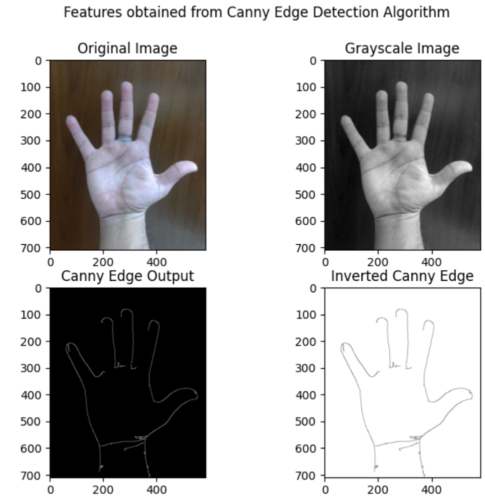

Fig. 2. Canny Edge Detection(Dark Background)

- Performing Otsu's binarization and subtracting 255 from every pixel.
- The obtained image had salt and pepper noise, so a Median Blurring Filter was used.

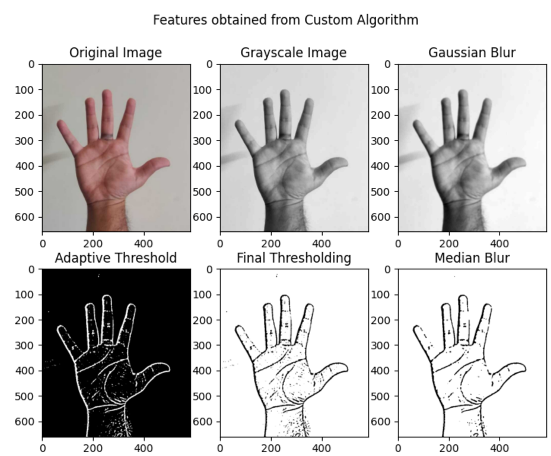

Fig. 3. Custom Algorithm(Light Background)

Canny Edge caused a loss of significant features and for letters which were modifications of a 'fist'. For images with dark background, Custom Algorithm performed better and preserved all the palm lines as well as the outline of the hand,

3.2. Classification

- Convolution2D: This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs.

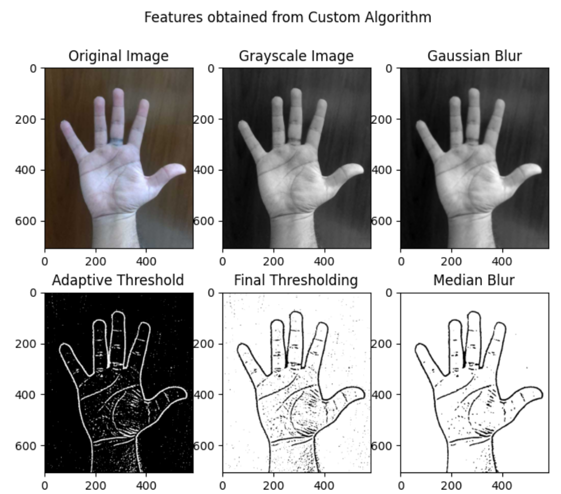

Fig. 4. Custom Algorithm(Dark Background)

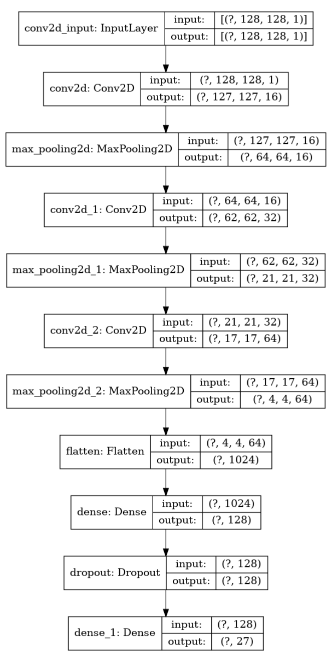

Fig. 5. CNN Model

- MaxPooling2D: Downsamples the input representation by taking the maximum value over the window defined by pool size for each dimension along the features axis.
- Flatten: Flattens the input. Does not affect the batch size. Dimensional reduction.
- Dense: Creates a fully connected Neural Network layer.
- Dropout: The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting.

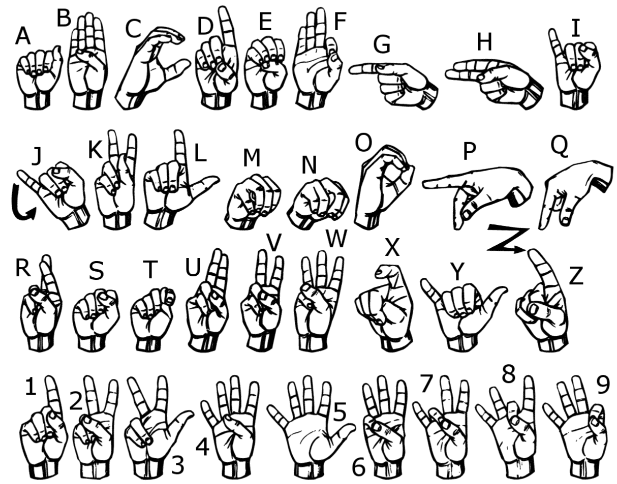

Fig. 6. American Sign Language

- Optimizer: Stochastic Gradient Descent
- Epochs: 9
- Batch Size: 10

4. RESULTS

The above stated model achieved convergence within 9 epochs with a test accuracy of 0.9960 and train accuracy of 0.9230.

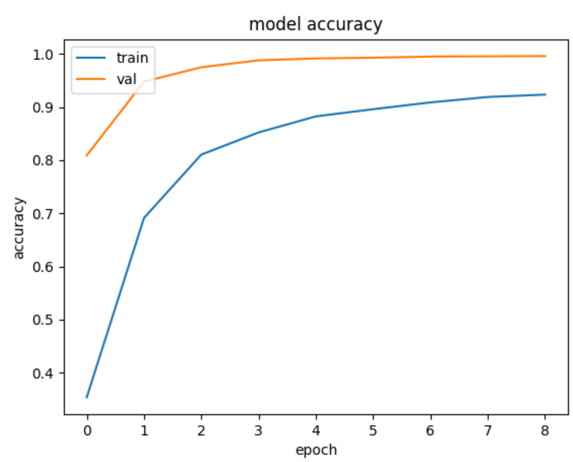

Fig. 7. Accuracy vs Epochs

Real-time prediction: Captures video frames and performs the image transformation same as data set generation

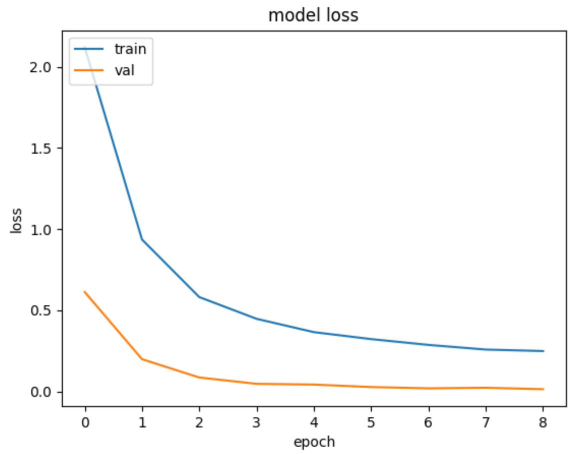

Fig. 8. Cross Entropy Loss vs Epochs

method and is then predicted using the saved model(s).

Speak Feature: Using pyttsx3, speaks the current text, speech settings such as speed and voices can be modified.

Auto-suggest Feature: Using hunspell, in order to overcome the wrong predictions, hunspell uses American English and automatically suggests corrections.

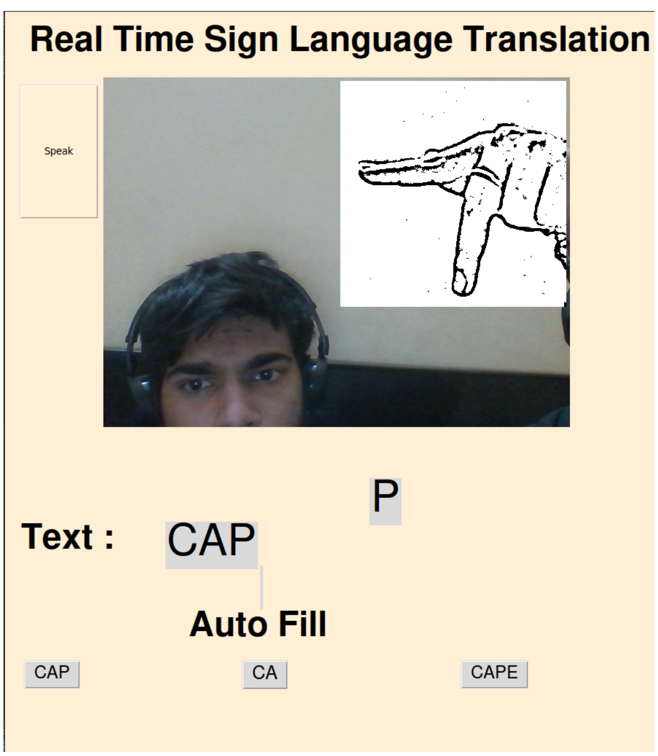

Fig. 9. GUI Implementation

5. CONCLUSION

In this report, a functional American sign language recognition system has been developed. With image transformation techniques, model complexity was significantly reduced without comprising a lot of accuracy. Alignment and aspect ratio plays an important role in the classification of new images. We improved accuracy by developing a technique similar to Canny Edge Detection, the key step was Adaptive Thresholding.

Future work comprises of background removal techniques(tried using Grabcut from opencv) which use neural networks and semantic segmentation and a robust model for multi-user application which required better accuracy for newer data.

6. REFERENCES

[1] T. Dasgupta and A. Basu, “Prototype machine translation system from text-to-Indian sign language,” in Proceedings of the 13th international conference on Intelligent user interfaces - IUI ’08, Gran Canaria, Spain, 2008, p. 313, doi: 10.1145/1378773.1378818.

[2] K. Tiku, J. Maloo, A. Ramesh and I. R., ”Real-time Conversion of Sign Language to Text and Speech,” 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 346-351, doi: 10.1109/ICIRCA48905.2020.9182877.

[3] H. Shin, W. J. Kim, and K. Jang, “Korean sign language recognition based on image and convolution neural network,” in Proceedings of the 2nd International Conference on Image and Graphics Processing - ICIGP ’19, Singapore, Singapore, 2019, pp. 52–55, doi: 10.1145/3313950.3313967.

[4] V. Aiswarya, N. N. Raju, S. S. J. Joy, T. Nagarajan, and P. Vijayalakshmi, “Hidden Markov Model-Based Sign Language to Speech Conversion System in TAMIL,” in 2018 Fourth International Conference on Biosignals, Images and Instrumentation (ICBSII), Mar. 2018, pp. 206–212, doi: 10.1109/ICBSII.2018.8524802.