Indian Sign Language Translation System Using Deep Convolutional Neural Network

Manas Jashnani[1], Jai Bhavsar[2], Faiz Shaikh[3] and Sharmila Wagh[4]
[1,2,3] *Student, Modern Education Society's College of Engineering*
[4] *Faculty, Modern Education Society's College of Engineering*

Abstract **- The deaf and mute individuals are a very important part of our society. Due to lack of aware-ness of sign languages among the general population, these individuals are often isolated. An effective solution for breaking this barrier is needed. In this paper we have discussed the implementation of a sign language translator by using Convolutional Neural Network (CNN). A few preprocessing techniques such as Skin Masking and Canny Edge Detection is applied. Finally, the model of Convolutional Neural Network is applied by taking into account its recent advances for better detection of hand movements and classification. It has an excellent performance in machine learning problems.**

Index Terms **- Skin Masking, Canny Edge Detection, Sign Language Translation, Convolutional Neural Network (CNN)**

INTRODUCTION

Deaf people are a short portion of our society, but they are way far from being treated as equally important or intelligent. They have been great influencers for most of the people out there just like us. Indian sign language is used not only in India but also Pakistan and parts of Bangladesh. It is but obviously the only way of communication among the hearing and speech impaired. The common people need to realize the importance of such people. Even in the corporate sectors, the bright minds of such people are seen to be considered as capable as of the clerks and ignored. They have a true potential to bring a change in society. Basically, if people are not that much aware about this very serious condition, we at least need to bring this situation to their attention. Also, we can try and make use of technology. Technology is the only log of wood using which we can bridge the gap between the normal and the deaf and mute people. Sign languages are a combination of hand gestures, arm movement, head movements and also facial expression. Considerable amount of research is done in this field. There are many different ways to achieve this using the technology. As the advancements seen in do-main of ML and AI, we can consider this as a tough but possible job. Using Covolutional Neural Networks along with the other techniques of which to name a few: SIFT, SVM, KNN could be used to solve the problem.

"Bangla Sign Language Detection using SIFT and CNN", Shirin Sultana Shanta, Saif Taifur Anwar, and Md. Rayhanul kabir [1]. In this paper, Convolutional Neural Network is used with preprocessing techniques such as Edge detection, Skin Masking etc. They are used along with Scale Invariant Feature Transform (SIFT) to recognise Bangla letters although their model was not able to detect two hand gestures. The authors concluded by suggesting using SURF in place of SIFT by looking at the accuracy and performance of the system. They also brought into notice the increase in the accuracy when preprocessing techniques are applied. "Real-time American Sign Language Recognition with Convolutional Neural Networks" Brandon Garcia and Sigberto Alarcon Viesca [2]. In this paper they have implemented a system by taking video input, recognizing the alphabet in the particular frame of the video, and trying to display the word that it can be according to the output scores. They utilize a Google Net architecture already trained on two different datasets and then use transfer learning. The Transfer Learning is an ML technique used to train on big data sets and then normalized to fit more concise data. They used the CNN for image processing and classification was done using an SVM. They were able to get very accurate results on the first time user for the letters from A-K. "Traffic Sign Classification using Hybrid HOG-SURF Features and Convolutional Neural Networks", Rishabh Madan, Deepank Agrawal, Shreyas Kowshik, Harsh Maheshwari, Siddhant Agarwal and Debashish

Chakravarty [3]. Basic aim of the paper is to try to use Histogram of Oriented Gradient (HOG) in combination with Speed Up Robust Features (SURF) along with Convolutional Neural Network to classify the traffic signs. We are interested in the use of SURF that they have done. The scale and rotation invariance of SURF increases reliability for practical purposes. SURF makes use of box filters which gives a computational advantage. SURF's feature vectors' dimensionality should be fixed, and clustering should also be applied. "Object Detection using Convolutional Neural Networks", Reagan L. Galvez, Argel A. Bandala and Elmer P. Dadios [4]. The basic aim of this paper is to detect objects in medical imaging and navigation used by robots. Here, the basic working of Convolutional Neural Networks is conveyed. Every layer of CNN is explained deeply. Like in paper [2], here also the technique of Transfer learning is used so as to save time which could have been wasted to train the model from the start. Also, it is seen that CNN has a better accuracy than R-CNN. "Deep Convolutional Neural Networks for Sign Language Recognition", G.Anantha Rao, K.Syamala , P.V.V.Kishore, A.S.C.S.Sastry [5]. Selfie mode Sign Language gestures are taken as a dataset input for the sign language recognition. This paper as likely the other papers also does the same with the help of CNN. Plain and simple tweakings in the pooling layer of CNN are performed to get the desired results with a good amount of accuracy. "American Sign Language Recognition using Deep Learning and Computer Vision", Kshitij Bantupalli and Ying Xie [6]. In this paper, use of RNN is done for temporal features along with CNN. No facial gestures were taken. Background change lowered the accuracy hence the model was needed to be trained more. The paper emphasizes to use a region of interest so that more accuracy as well as less background change will take place. "A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK", Shaharyar Ahmed Khan Tareen and Zahra Saleem [7]. This paper presents a comparison of various feature detection techniques viz. SIFT, SURF, KAZE, AKAZE, ORB and BRISK. We are, however, more interested in the comparison of SIFT and SURF. Both the techniques in question are scale-invariant, i.e. they work well even with images with objects in different scales. The accuracy of SIFT is more than SURF overall for all geometric transformations. However, SURF works more efficiently than SIFT. "Understanding of a Convolutional Neural Network", Saad ALBAWI, Tareq Abed MOHAMMED and Saad AL-ZAWI [8]. Here, the authors discuss how each individual parameter affects the performance of Convolutional Neural Network. The authors note that the convolutional layer is the most time-consuming layer, being the most important one. The number of levels in the network also affects the performance. The accuracy of the network increases with more levels, but as the levels increase, the processing time of the network increases as well. "Comparison of Feature Detection and Matching Approaches: SIFT and SURF", Dr Darshana Mistry, Asim Banerjee [9]. Similar to [7], [9] discusses the difference between SIFT and SURF in terms of descriptors, size of descriptors, orientation, scale space, etc. They use one image and its compare features detected by SIFT and SURF with its transformed versions. It is concluded that SIFT works well in case of scale and is faster while SURF works better for other transformations. Both methods seem equally effect against different illumination. "Region-based Convolutional Networks for Accurate Object Detection and Segmentation", Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik [10]. The model developed gives significantly better results over traditional object detectors. This is done by using high capacity CNNs along with bottom-up region proposals that localize segment objects. It is very useful to pre-train the model with huge amounts of image classification data. The network is then fine-tuned for scarce data. Thus, a combination of classical image Vision tools and deep learning using RCNN provide really good results. "Recent Developments in Sign Language Recognition Systems", M.F. Tolba, A.S.Elons [11]. It was observed that with previously developed systems, the accuracy achieved was high. However, all those systems were developed with certain constraints taken into consideration which when changed, the model failed to produce decent results. A semantic oriented post-processing module based on Natural Language Processing rules can be used for the detection and correction of semantic level and lexical level errors. "Hand Gesture Recognition Based on Karhunen-Loeve Transform", Joyeeta Singha, Karen Das [12]. For preprocessing, skin filtering, Canny edge detection were used. Then the next step was K-L

transform and finally angle based classifier is used to detect the
gesture. Then the results from Euclidean distance-based classifier and those from angle-based classifier were compared and it was found that both gave similar results. Thus, accurate result was obtained. However, the model faced difficulties while classifying the similar hand gestures.

III. SYSTEM DETAILS

The system architecture is as shown in 1. This translation system would be able to detect hand and the gesture which a person is performing. It will take input as an image, taking frames from the live video input or the recorded video, perform preprocessing such as mask skin, detect edges of the hand, applied to CNN which finally after getting a trained model ready obtain the bag of words from which it will give one alphabet/word as an output. This alphabet/word can then be further converted from text to speech. The modules highlighted by the system architecture are as follows. Video capture module: In this module, frames are extracted through a live video input or an already recorded video as the input required for preprocessing is an image. Preprocessing: The preprocessing module is used for applying preprocessing techniques like skin masking and Canny edge detection for better results while using CNN. Also, in this module feature extraction also takes place for a better input to be given to CNN.CNN: The CNN module involves training of the model using CNN. The layers used in CNN are Convolutional, Max Pooling and Dropout. Activation functions used are ReLU and SoftMax. Once a model is trained, we can perform gesture recognition. Interface: The interface module uses the trained model to recognise the gestures in the video input. The output is then converted into text and speech.

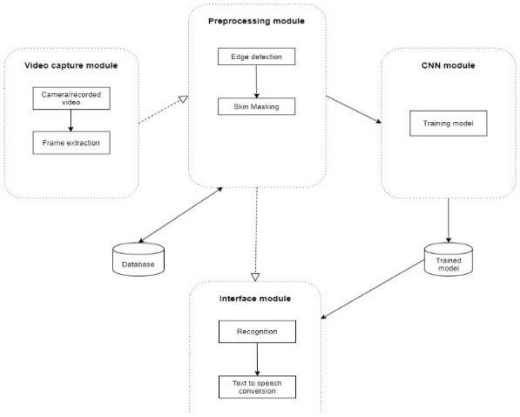

Figure 1. Basic structure of Sign Language Translator

IV. MATHEMATICAL MODEL

$$S = \{Images,\ Gesture,\ CNN,\ KMeans,\ Skin\ \ Masking,\ Canny\ edge,\ Feature\ reduction\ test\ parameter -\ \ Execution\ time,\ speed,\ accuracy,\ size\}$$

$$S = SampleSpace$$

$$I = \{Images(64 \times 64 \times 3)\ jpg/png\ format\}$$

$$I = Input$$

$$O = \{Gesture\ alphabets - A\ to\ Z\}$$

$$O = Output$$

$$Algorithms = \{CNN, Kmeans, Skin\ masking,\ Canny\ edge, Feature\ reduction\}$$

$$CNN \rightarrow z = x * f$$

$$x \rightarrow Image$$

$$* \rightarrow Convolution$$

$$f \rightarrow Filter$$

$$ReLU = f(x) = x^{+} = max(0, x)$$

where $x \rightarrow$ *input to a neuron*

$$Max\ \ Pooling\ :\ a_j\ =\ max(a_i^{nxn}\ \ u(n,n))$$

$$Window\ function : u(x,y)$$

$$nxn\ \rightarrow patch\ of\ units\ of\ input$$

$$Fully\ connected :$$

$$Input : x \in R^m$$

$$Outputi : y_i \in R$$

$$y_i = \sigma(\omega_1 x_1 + ... + \omega_m x_m)$$

$$\sigma \rightarrow non - linear\ function$$

$$\omega_i \rightarrow learnable\ parameters$$

$$y = \begin{bmatrix} \sigma(\omega_{1,1}x_{1,1} + ... + \omega_{1,m}x_m) \\ . \\ . \\ . \\ \sigma(\omega_{n,1}x_1 + ... + \omega_{n,m}x_m) \end{bmatrix}$$

$$SkinMasking :$$

$$Threshold\ values\ adjusted\ as\ needed$$

$$Min = 3, 50, 50,\ \ Max = 33, 255, 255$$

$$Softmaxfunction :$$

$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

V. IMPLEMENTATION

Training:

1) Dataset: The dataset comprises of images extracted from ten to twenty-second videos of different gestures, both one-handed and two-handed. The dataset contains numbers 0-9 and letters A-F.

2) Preprocessing: We blur the images to avoid the effect of creases and folds of the skin on the accuracy. We then convert the image from RGB to HSV followed by skin masking. =We perform further erosion and dilation on the images. Finally, we perform Canny Edge Detection on the images.

3) Splitting: Once the preprocessing is completed, we split the images into training and testing sets. We then pass the training set through the CNN layers.

4) Training using CNN: We train a model by passing the dataset through multiple layers of the CNN. We first use the Convolution and Max Pooling layers thrice. In the Convolution layer, we multiple our image matrix with a 5x5 filter matrix. The output is a matrix called a feature map. We use the ReLU function for convolution. The Max Pooling layer, we reduce the feature map by selecting the largest element in every 2x2 section of the feature map. Then, we use the Dropout layer. The Dropout layer randomly removes half the nodes, thus preventing the problem of overfitting. Finally, we use the Dense or Fully Connected layer as our output layer. We use the SoftMax function in this layer.

Prediction:

1) Input: For prediction, we use the same steps as the training layer for the processing of the input. The input can be a video or a live camera feed.

2) Processing: We use blur, erosion, dilation, skin masking and edge detection methods, same as in training. We use prebuilt face detection function to avoid its contours being detected in the next step.

3) Contour Detection: We use contours for detecting the largest continuous area in the input. This helps in detection of the gestures in the image. The largest area is extracted as the output.

4) Prediction: Once the area of interest has been detected, we crop the raw image to only retain the gesture. After the same processing of the cropped image, the translation of the gesture is given as the output by the system.

VI. EXPERMIENTAL ANALYSIS

We have tried to implement this system for ISL i.e. the Indian Sign Language. We created our own database by taking images with our mobile devices. What we did exactly was to record a video of about 30seconds with slight variation in hand movements in the process to build a robust dataset. All the video recordings were done indoor with ample lighting conditions. After getting the video we extracted about 500 images per video which in total for every alphabet and digits went to41,361 images. After a split 80% for training and 20% for testing we obtained the following result. Here are some of the images from the actual execution in 2. The accuracy of the model is also shown in 3.

VII. RESULTS

The model works well on input from cameras of high quality in the presence of ample lighting. The testing accuracy of the model in such conditions was observed to be around 96%. The accuracy decreases upon decrease of camera quality and brightness. The model successfully detects one-handed as well as two-handed gestures, which was our aim.

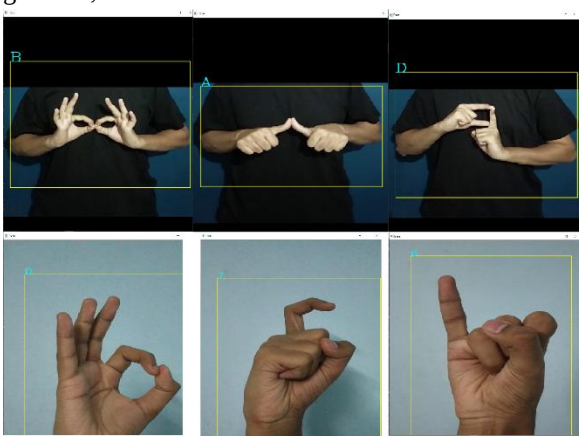

Figure 2: Letters'/Digits' Output

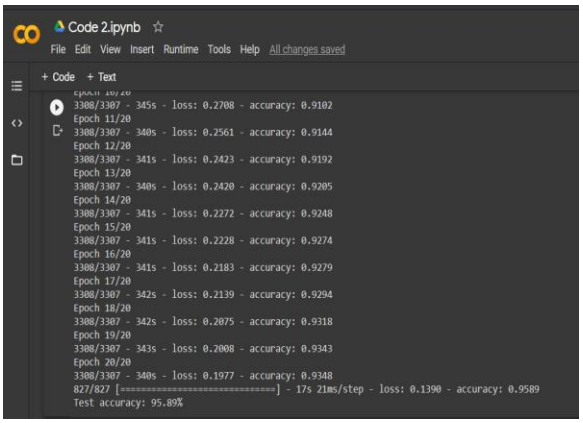

Figure 3: Accuracy of Model

VIII. CONCLUSION

By this implementation, it is clear that preprocessing increases efficiency. Based on our experiment, further systems developed can be made more advanced and can achieve even more efficiency. This system can also be upgraded to translate multiple sign languages. Also, if paired with the face expression detection could provide a complete solution.

REFERENCES

[1] Shirin Sultana Shanta, Saif Taifur Anwar, and Md. Rayhanul kabir, "Bangla Sign Language Detection using SIFT and CNN" (2018).

[2] Bandon Garcia and Sigberto Alarcon Viesca. "Real-time American Sign Language Recognition with Convolutional Neural Networks" (2016).

[3] Rishabh Madan, Deepank Agrawal, Shreyas Kowshik, Harsh Maheshwari, Siddhant Agarwal and Debashish Chakravarty, "Traffic Sign Classification using Hybrid HOG-SURF Features and Convolutional Neural Networks" (2019).

[4] Reagan L. Galvez, Argel A. Bandala and Elmer P. Dadios, "Object Detection using Convolutional Neural Networks" (2018).

[5] G.Anantha Rao, K.Syamala2 , P.V.V.Kishore, A.S.C.S.Sastry, "Deep Convolutional Neural Networks for Sign Language Recognition" (2018).

[6] Kshitij Bantupalli and Ying Xie, "American Sign Language Recognition using Deep Learning and Computer Vision" (2018).

[7] Shaharyar Ahmed Khan Tareen and Zahra Saleem, "A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK" (2018).

[8] Saad ALBAWI, Tareq Abed MOHAMMED and Saad AL-ZAWI, "Understanding of a Convolutional Neural Network" (2017).

[9] Dr Darshana Mistry, Asim Banerjee, "Comparison of Feature Detection and Matching Approaches: SIFT and SURF" (2017).

[10] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, "Region-based Convolutional Networks for Accurate Object Detection and Segmentation" (2015).

[11] M.F. Tolba, A.S.Elons, "Recent Developments in Sign Language Recognition Systems" (2013).

[12] Joyeeta Singha, Karen Das, "Hand Gesture Recognition Based on Karhunen-Loeve Transform" (2014).