Translation of Gesture-Based Static Sign Language to Text and Speech

Sabitha Gauni*, Ankit Bastia, B Sohan Kumar, Prakhar Soni, and Vineeth Pydi
Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India.
Email: sabithag@srmist.edu.in

Abstract. As human beings, most of us convey our thoughts by speech and facial expressions, but according to the latest survey conducted, it was found that roughly 1% of the population in India is deaf and mute. These people communicate with others using hand gestures and facial expressions. However, most people find it difficult to understand gestures. To eliminate this gap, we develop static gesture classification based on sign language standards and then converting to text and speech of a given local dialect.

Keywords: Gesture Recognition, Pre-Processing, Background Isolation, Image Thresholding, Convolutional Neural Network.

1. Introduction
There is a communication gap between the people who use sign gestures to communicate and people who do not. So, there comes a need for devices or translators to translate sign language to text and speech of a given local dialect [1].

Indian Sign Language has predominantly used sign language in South Asia. As we know that most deaf people in India tend to use sign language but still, it is not widely used for teaching purposes in schools. Many efforts have been made to encourage ISL by organizations working for deaf and mute people [8]. Given the complexity of these signs, they are not usually known to the general folk, thus making the conversation difficult between the people who are deaf and mute and the normal people. The alphabets in ISL are shown in figure 1.

With the recent advancement in ML, DL, and CV, image recognition has been quite easy. With the motivation to make speech-impaired life more comfortable, our paper focuses on recognizing gestures using different Convolutional Neural Network architecture [4].

The general approach to building a system with the desired functionality includes hardware-based systems with embedded components in-house and software-based systems like computer vision. A hardware-based system requires different types of equipment, like gloves and sensors. We focus on computer vision and machine learning algorithms to reduce the complexity and hardware requirement in the glove-based approach [9].

Sign language consists of both words described in gestures and finger spellings. Gestures can be both static as well as dynamic. This paper presents a comprehensive model for classifying static signs and then converting them to text and speech of a given local dialect supported by gTTS. The signs considered in this paper include 25 common words [10].
2. Related Work

Sign language can be recognized using either of the two approaches. The first one is through the Hardware-based system. In this approach, the user is required to wear gloves. The second approach is the vision-based approach, where the gesture is recognized using the concept of computer vision.

In the glove-based approach, the main components are sensors and gloves. In the paper presented by S Yarisha Her, V S, Madhuri K Murthy [5], they attach flex sensors with the gloves to detect the gesture and then convert it into speech using a Bluetooth module and a smartphone.

Kusurnikai D, Anil K, S K Raju [6] developed a dual-handed sign language recognition system using the least eigenvalue algorithm. The image is processed using MATLAB, and then the output is converted to text and speech.

Nachiket Deo, Akshay Rangesh, and Mohan Trivedi [7] have proposed a system where the Hidden Markov Models technique is used to recognize one of Hand gesture applications recognition. They have trained the HMMs on HOG and CNN features, which are more complex shape descriptors than the typical HMM-based approach. Also, to reduce the overfitting of HMMs, dimensionality reduction and data augmentation were taken into consideration.

One of the other uses of gesture recognition is that it can be interfaced in the video game system. Instead of using keyboard buttons or joystick, gestures can be used as a video game command [8]. The hand gesture is detected and tracked from a cluttered background using contour comparison and skin detection algorithm. The multiclass support vector machine (SVM) and the bag-of-features concept are used to further the image processing [11-12].

Alex Krizhevsky and Ilya Sutskever had a system developed training a deep convolutional network to classify a million high-res images into several different classes. The error percentage obtained in their test data was 37.5. They used non-saturating neurons and multiple GPUs to make the training faster. Also, they used a method called dropout to reduce the overfitting in the fully connected layers [13].

Hand gestures were tested on a CNN model using Inception v-3 in the paper presented by Aditya Das, Shantanu G., Dr. Dhannjay K, and Khayati Suratwala [2]. A ratio of 1:5 images from each class was taken for testing and training. For the model proposed by them, they got a validation accuracy of above 0.9.

In the paper [3], gestures are segmented using a novel contour and GrabCut algorithm. The GrabCut algorithm was used to detect the features of the image from complex background. On the other hand, a novel contour algorithm was used to detect the image’s features with a simple background. They discovered that the number of iterations was highly dependent on the image background, and thus, the novel contour gave better results than the GrabCut algorithm.
3. Proposed Methodology

A convolutional neural network (CNN) architecture is used to create a model which helps in computer vision problems.

![Figure 2. Yellow Threshold data](image)

The entire process can be separated into three phases, namely preprocessing, training and testing. The first and foremost step in a complete computer vision-based recognition system is to preprocess the pictures in the dataset to extract the noise-freehand region, which is then converted into a tensor using a NumPy array. Skin color extraction has many computation complexities due to skin color variation from person to person, so here we made the color constant by using a glove of certain defined color. The tensor enters the input layer of CNN, which are the pixel values. A CNN architecture contains a convolutional layer, a max-pooling layer, also, a fully connected dense layer/hidden layer. At the convolution layer, the number of filters is set along with the kernel size and the strides at which the kernel moves. The convolution performed by the filters is then fed into the max-pooling layer to get the maximum values to reduce the dimensionality and learn the important features. Convolution layers and several max-pooling layers are stacked upon each other to generate a sequential CNN architecture which is then flattened and connected to the dense layer. The flattening refers to making tensors one-dimensional.

Model is trained on the principle of minimizing loss which can be done by gradient descent. Many optimizers can help us achieve our desired result, e.g., SGD, Adam etc. In this project, an Adam optimizer has been used. The convolution layers have been activated using the activation function rectified linear unit (ReLU) instead of sigmoid since sigmoid activation would take ages for the model to be trained as the coefficients become very low due to back propagations, which is possible due to advancements in GPU. The 12 GB GPU of Google Collab was utilized for training. Dropout is added after the dense layer to avoid overfitting. The output layer has the activation function of softmax.

Alexie is a CNN architecture that won the Image net large-scale visual recognition challenge in 2012. The dataset was trained in the following models in color and grayscale configurations. The color chosen here is yellow as shown in figure 2, and the dimension of the training images was set to 200x200.

3.1 Customized CNN

Each of the training pictures fed into this architecture is resized to a size of 200x200. The custom model has six different layers as shown in figure 3; the three convolution layers have 64, 128 and 64 filters of kernel sizes of 3, 5, and 3 (square matrix). The three maximum pooling layers have window sizes of 2x2. These sequential layers are stacked one after the other. Dropout is used to avoid overfitting. After going through the layers above, the data are flattened and enter the dense layer (fully connected layer) of size 80. The Corrected linear unit activation function is used in all the convolution layers. The output
layer has softmax activation. The loss function used here is sparse uncompromising cross-entropy for multiclass classification [14,15].

Figure 3. The architecture of custom model (\( ? \) denotes the number of images)
4. Result

From the comparison between models trained with yellow images and grayscale images, it can be inferred that due to the grayscale image’s dimensional nature (200x200), it takes less time per epoch trained the yellow ones, which pose as three-dimensional tensors (200x200x3).

Figure 4. Architecture summary based on Alex Net

Figure 5. Model comparison of Alex net and six-layered CNN model with colored (multidimensional) and grayscale datasets
The time taken to train with grayscale images was 4 minutes 38 seconds in the custom model and 4 minutes 50 seconds for the Alexnet model as in figure 4 with a test accuracy of 97.22% and 99.22%, respectively. The models trained on yellow images took 15 minutes 19 seconds and 15 minutes 28 seconds for 10 epochs with a test accuracy of 96.65% and 98.67%. Overfitting, to some extent, was observed from the plots obtained from Matplotlib and the pyplot library. Overfitting to a little extent was observed during real-time classification.

The model loss of training and validation set were plot against the epochs is shown in figure 5. Similarly, model accuracy was observed against the number of epochs. Due to its lower dimension, the grayscale model has a lower size hence is lighter on the system.

5. Conclusion
The static gestures were trained on several models without background thresholding to varying levels of success, which led to implementation isolating a single color due to computational constraints of taking all skin colors into account. The fair classification was observed with the extraction of frames from a live feed put into models trained in grayscale and yellow images. The grayscale ones were faster in training and had an optimal size compared to models trained on yellow images. Overfitting was observed in many models, as discussed, but most of the results were accurate to the naked eye. It was also observed that the custom CNN model used here performs at par with the deep Alexnet model. The prediction was then fed into a Google Translator module in python to translate the predicted text into the desired language using Google Translate. Using TTS text to speech service speaks in the respective languages was generated.

6. Future Works
Due to the large size of the dataset, which was 2000 images per class for about 25 classes, it was virtually impossible to annotate each image individually. An attempt was made to automate the annotation process with little success. Annotation is necessary to implement object detection using state-of-the-art architectures like YOLO, ResNet, Inception, etc.

The aforementioned project involved a limited set of distinct static gestures. There is a scope to work with complex dynamic gestures involving sentiments of a subject and have an NLP implementation for smoother translation.

There is also a desire to implement the architecture to work at Edge taking cues from Edge computing.

References
[1]. Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, "ImageNet Classification with deep convolutional neural networks," 2012, Proceedings of the 25th International Conference on Neural Information Processing Systems, pp-1097-1105.
[2]. Das, A., Gawde, S., Suratwala, K., & Kalbande, Dr. Dhananjay Kalbande, "Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images," 2018 International Conference on Smart City and Emerging Technology (ICSCET). doi:10.1109/icscet.2018.8537248
[3]. Malladi Sai Phani Kumar, Veerapalli Lathasree and S.N. Karishma," Novel Contour Based Detection and GrabCut Segmentation for Sign Language Recognition,” 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE,2017, pp. 738-742.
[4]. Neel Kamal Bhagat, Vishnusai Y, Rathna G N, "Indian Sign Language Gesture Recognition using Image Processing and Deep Learning,” 2019 Digital Image Computing: Techniques and Applications (DICTA). doi:10.1109/dicta47822.2019.8945850.
[5]. S Yarisha Heera, Madhuri K Murthy, Sravanti V S, “Talking hands — An Indian sign language to speech translating gloves,” 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), IEEE, 2017, pp. 746-751.
[6]. Kusurnikai Dutta, Satheesh Kumar Raju, Anil Kumar, “Double-handed Indian Sign Language to speech and text,” 2015 Third International Conference on Image Information Processing (ICIIP), IEEE, 2015, pp. 374-377.

[7]. Nachiket Deo, Akshay Rangesh and Mohan Trivedi," In-vehicle Hand Gesture Recognition using Hidden Markov models" 2016 IEEE 19th International Conference Intelligent Transportation Systems, IEEE, 2016, pp. 2179-2184.

[8]. Nasser H. Dardas and Nicolas D. Georganas, “Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques,” IEEE Transaction on Instrumentation and Measurement, volume 60, no. 11, pp.132-137 November 2011.

[9]. Neha Sharma, Vibhor Jain, Anju Mishra, "An Analysis Of Convolutional Neural Networks For Image Classification," Procedia Computer Science, Volume 132, 2018, Pages 377-384, ISSN 1877-0509.

[10]. Clebeson Canuto dos Santos, Jorge Leonid Aching Samatelo, Raquel Frizera Vassallo, "Dynamic gesture recognition by using CNNs and star RGB: A temporal information condensation," Neurocomputing, Volume 400,2020, Pages 238-254, ISSN 0925-2312.

[11]. Mahidhar, B. V. S., Sankeerthana, D. L., Reddy, K. B., Nikhath, G. A., & Poovaraghan, R. J. MEDICAL TRANSCRIPTION USING SPEECH RECOGNIZER.

[12]. Sudha, S., & JothiLakshmi, S. (2013). Tamil Sign Language to Speech Translation. International Journal of Computer Applications, 82(11).

[13]. Patel, R., Everett, M., & Sadikov, E. (2006, October). Loudmouth: Modifying text-to-speech synthesis in noise. In Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility (pp. 227-228).

[14]. Abushariah, M. A. (2017). TAMEEM V1.0: speakers and text-independent Arabic automatic continuous speech recognizer. International Journal of Speech Technology, 20(2), 261-280.

[15]. Beena, M. V., Namboodiri, M. A., & Dean, P. G. (2017). Automatic sign language fingerspelling using convolution neural network: analysis. Int J Pure Appl Math, 117(20), 9-15.