Sign Language Translator

Prashant Kannaujia[1] Om Parkash Bhati[2] Prakhar Chaurasia[3] Niharika Varshney[4] Gulshan K Dubey[5]
[1,2,3]Student [4]Assistant Professor [5]Senior Assistant Professor
[1,2,3,4,5]Department of Electrical & Electronics Engineering
[1,2,3,4,5]ABESEC, Ghaziabad, UP, India

Abstract— The intention behind making that project is to provide an aim and digital autonomous platform to hearing and speech impaired people. Sign Language is a kind of a language which is often used by hearing and speech impaired people to convey their message to normal people. They use simultaneous combination of hand gestures, its orientation, hand shapes etc. That system is capable of recognizing gestural symbols and are often used as a method of communication with hard of hearing individuals. That project proposes a system to help normal individuals to simply communicate with deaf and dumb individuals, instead they tend to use a camera as a tool to implement the sign language system.

Keywords: CNN (Convolutional Neural Network), FC (Fully Connected), ANN (Artificial Neural Network), GPU (Graphics Processing Unit)

I. INTRODUCTION

The sign language translation system uses web-camera to induce pictures or continuous video image (from deaf or dumb people) which might be interpreted by the appliance. Acquired gestures are assumed to be translation, scale and rotation invariant. During this method the steps of translation are acquisition of pictures, binarized, classification, hand form edge detection and feature extraction. After obtaining vectors feature extraction state then pattern matching is done by comparison existing information. The user interface application is displaying and sending the message to the receiver. This method makes traditional individuals to speak simply with deaf/dumb person.

Human-Computer Interaction moves ahead in the field of sign language simplification. Sign Language Interpretation system is an efficient way to help the hearing and speech impaired people to interact with normal people with the help of computer. In vision-based hand gesture recognition system hand plays a vital role in communication. Vision based system have many challenges over conventional hardware-based approach, by efficient use of computer vision and pattern recognition, it is possible to work on such system which will be natural and accepted, in general. In this project we have used to classify the signs into readable English text/alphabets.

The objective of this project is to make automatic ISL translator - i.e., given input video of somebody signing a word, the model ought to be ready to predict that word is being signed. The model designed ought to be fairly sturdy to variations in angle, video quality, distance from the camera to the topic, variations in lighting etc. Such a model has many use-cases, and might be deployed either through automatic app or website for individuals to use. Its high connectedness within the lives of not solely the deaf-dumb, however additionally anyone who interacts with them.

There is a measure of 1.1 million deaf-mute individuals in Asian country. 98% of those individuals are measured as illiterate, creating linguistic communication their solely methodology of communication. This trend isn't expected to vary anytime shortly, as 2% of deaf children are able to attend school.

The number of individuals within the country with disability was calculable at 13 lakhs within the 2011 Indian census.

However, this might be a case of great under-reporting: The National Association of the Deaf estimates that there are 1.8 crore people who are deaf. The important numbers can be even larger as most countries have regarding 3% of their population as experiencing hearing disorder, that interprets to regarding 4 crore people.

Though the signs utilized in the country vary from region to region, many trial measures being created to systemize a convention that is noted. However, this development remains in its emergent stage and isn't reaching most deaf schools. Even otherwise, only 2% of deaf youngsters attend formal education. These twin issues of a developing customary and also the restricted reach of formal education necessitate tools which will be centrally developed and deployed at scale to accelerate awareness, adoption, and inter-operability of ISL.

II. TOOLS USED

A. TensorFlow:

TensorFlow is a computational framework for building machine learning models. TensorFlow provides a variety of different toolkits that allow you to construct models at your preferred level of abstraction. Wecan use lower-level APIs to build models by defining a series of mathematical operations.

B. OpenCV:

OpenCV is an open source C++ library for image Processing and computer vision, originally developed by Intel and now supported by Willow Garage. It is a library of many inbuilt functions mainly aimed at real time image processing.

C. CNN (Inception):

Convolutional neural network (ConvNets or CNNs) is one of the important categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used. CNN image simplification takes an input picture, process it and classify it under certain categories.

III. CONVOLUTIONAL NEURAL NETWORK

Practically, deep learning of Convolutional Neural Network models is to train and test each input which will be passing through a series of convolution layers with filters (Kernals), pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The fig 1 below is a working flow of Convolution Neural Networks to process an input image and classifies the objects based on values.

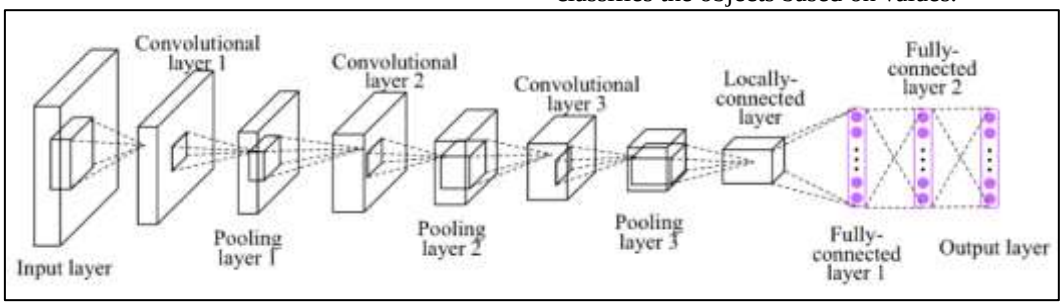


Fig. 1: Working flow of CNN

There are primarily six layers in any CNN model, namely:

A. Input Layer:

Layer that deals with taking input and supplying a similar to the input layer.

B. Convolutional Layer:

Convolution is the layer to extract features from an input image. Convolution maintains the interrelation between pixels by recognizing and learning image features using tiny squares of input data. It's like a mathematical process which takes two inputs one is like image matrix and other one is filter or kernel.

C. Non-Linearity (ReLU):

RELU is simply a non-linearity that is applied similar to neural networks.

D. igmoid:

takes a real-valued input and compress it to vary between 0 and 1.

$$\sigma(x) = 1 / (1 + \exp(-x))$$

tanh: takes a real-valued input and squashes it to the vary [-1, 1].

$$\tanh(x) = 2\sigma(2x) - 1$$

E. Pooling Layer:

Pooling layers section would cut back the number of parameters when the pictures are too large. spatial pooling also referred to as subsampling or down sampling that reduces the dimensionality of every map however retains the important information.

F. Fully Connected layer:

We flatten our matrix into vector and fed it to a fully connected layer like neural network.

ANN is non-linear model that is simple to use and perceive, compared to statistical ways. ANN is non-parametric model while most of statistical ways are parametric model that need higher background of statistic. ANN with Back propagation (BP) learning algorithm is widely used in solving varied classification and foretelling issues. Even though BP convergence is slow however it is guaranteed. However, ANN is black box learning approach, cannot interpret relationship between input and output and can't deal with uncertainties. to overcome this several approaches are combined with ANN like feature choice and etc.

IV. IMPLEMENTATION

The program applies Transfer Learning to an existing model and retains it to classify a new set of pictures. The program shows the way to take an inception v3architecture model which was trained on ImageNet and train a new top layer which will recognize different classes of pictures or images. We will be able to replace the image_dir arguments with any folder containing subfolders of pictures. The label for each image is taken from the name of the subfolder it is kept in.

Modern image recognition models have various parameters. Training them from scratch requires plenty of labelled training information and plenty of computing power (hundreds of GPU-hours or more). Transfer learning is a technique that actually shortcuts the abundant of this by taking a bit of a model that has already been trained on a connected task and reusing it in a new model.

It is quite obvious that it works not good as training the whole model but this thing works tremendously productive for several things.

Existing models of image recognition contains a lot of parameters. It takes a great deal of labeled training knowledge and plenty of computing power to train them from the scratch. Transfer learning somehow shortens the step by taking small chunks that is already trained.

A. Bottlenecks

To complete the script, it takes about 30 minutes or more to finish, much depending upon the working speed of the processing machine. First thing it do is to analyze the image present in the database and calculate the bottlenecks values of every character.

The script can take 30 minutes or more to finish, depending on the speed of the machine. First part analyses all the pictures in our database and calculate the bottleneck values of each of them. Bottleneck is not the formal term we simply use for the layer before the ultimate output layer that generally does the classification (TensorFlow called it as "image feature vector"). The next layer which we call penultimate layer is trained for the output values that is requisite for the user to differentiate between all the categories. It has a relevant and compact outline of the

images, since it has enough data for the classifier to make a good selection in a tiny set of values. Sorting a data that is required to distinguish between all the classes in ImageNet that is often useful to distinguish between new sort of objects is the reason our final layer retrains.

B. Basic Algorithm

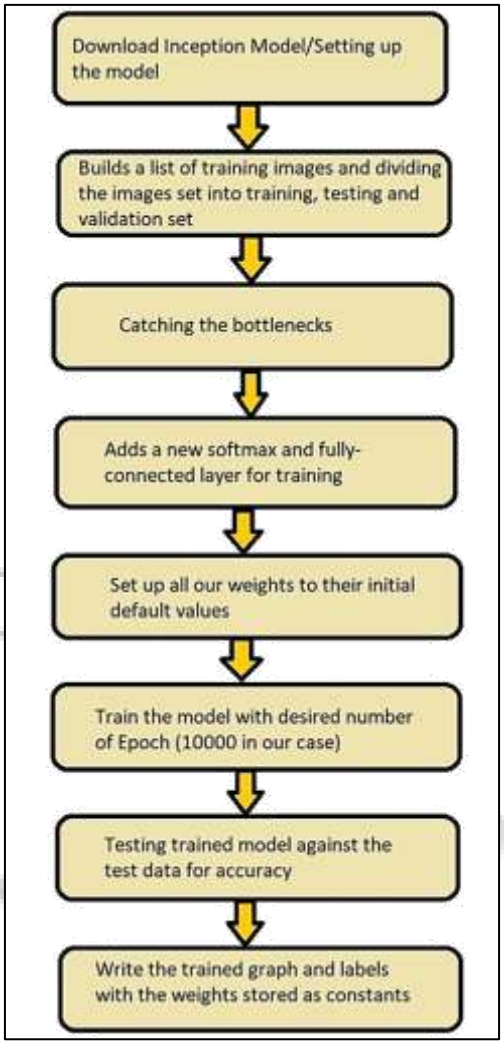


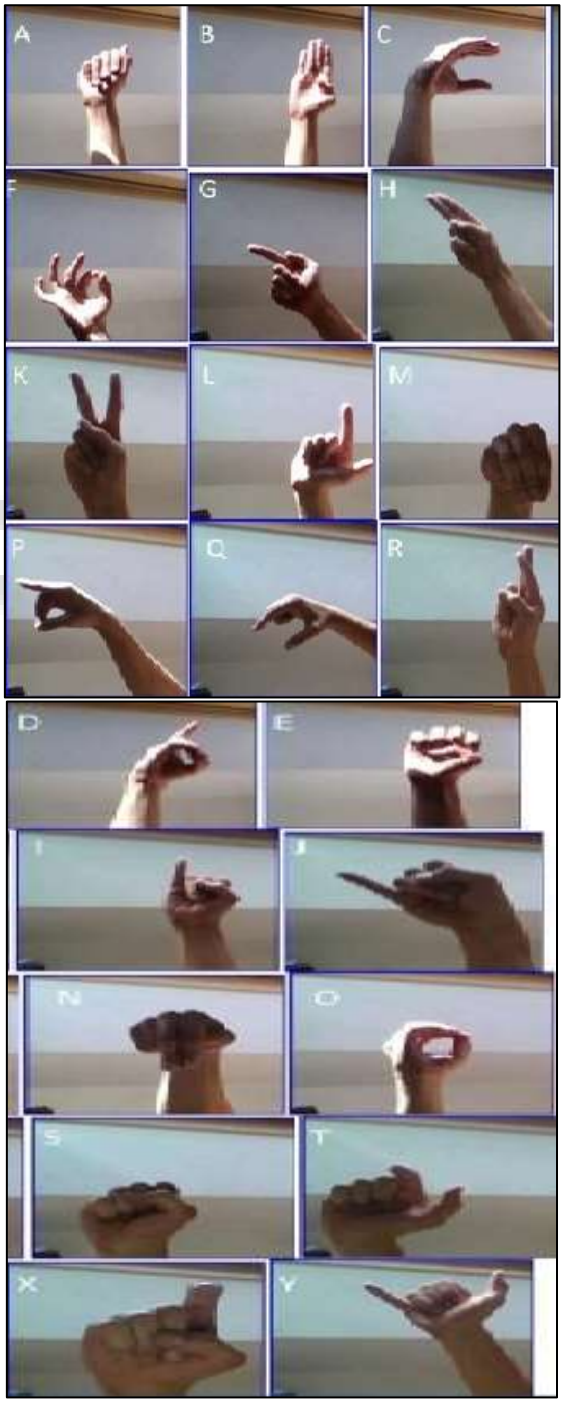
Fig. 2: Algorithm

C. Training

The training of the highest layer of the network gets started, once the bottlenecks gets completed. We can see a series of outputs, all showing training precision, validation accuracy and cross entropy. What percent of image used in the existing training scratch were labeled with the right class, this is what training accuracy shows. The accuracy on a arbitrary chosen cluster of images from a unique set is called validation accuracy. The major distinction is that the accuracy relies on images that the network can overfit to the noise within the data. A true measure of the performance of the network is to measure its performance on data set not contained within the training data -- this is often measured by the validation accuracy. If the train accuracy is high, the validation accuracy remains less, meaning the network is overfitting and memorizing explicit options within the training pictures that are not useful a lot of usually. Cross entropy can be a loss function which gives a glimpse of how well the learning method is progressing. The training's objective is to make the loss as tiny as possible, thus we can tell if the learning is functioning by keeping a watch on whether or not the loss keeps trending downwards, ignoring the short-run noise.

If we somehow managed to get the script acting on the sign gesture example pictures, we can be able to start looking at train it to recognize categories we care about. In theory, we need to point it at sub-folders group, all we need to do is point it at a group of sub- folders, containing solely pictures from that class. If we pass the folder of the sub-directories as the argument to image_dir, the script ought to train rather like it did for the sign gesture.

D. Dataset



These above sample images are the sign language of all alphabets including some special characters too.

V. RESULT

Result for character 'T'.

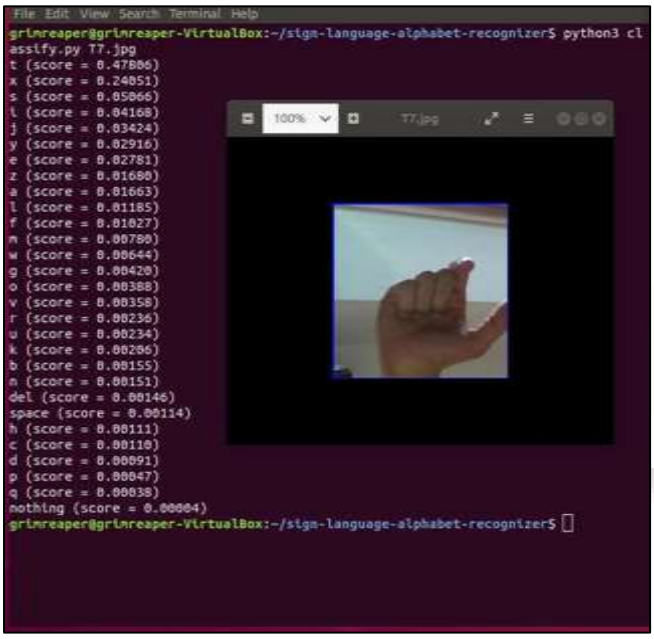


Fig. 3: Result of the above gesture i.e., 'T'

Result for character 'Y'.

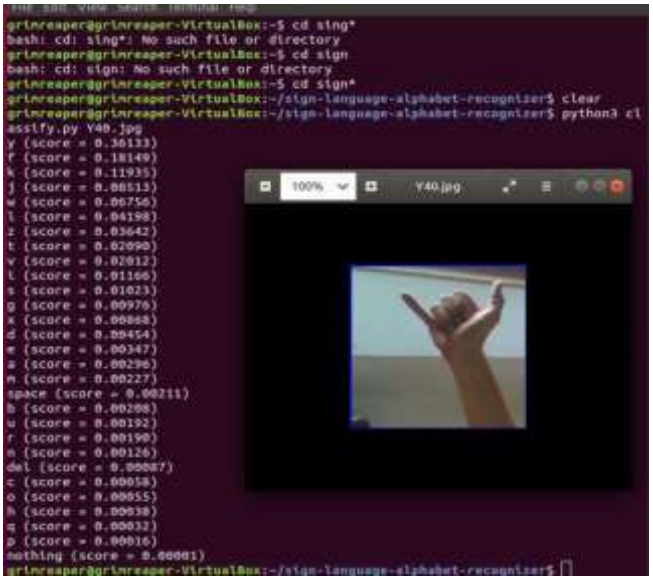


Fig. 4: Result of the above hand gesture i.e., 'Y'

We can see that in the above result screenshots that the gesture of the hand is compared with every character or alphabet present in the dataset, whosoever alphabet is more symbolic or similar to that image is that character.

VI. CONCLUSION

The software is intended in such a simplest way that future modification will be done simply. The subsequent conclusions will be deduced from the development of the project: It provides an abstract platform between the users. CNN has improved the efficiency i.e.; it had reduced the time consumption taken by other methods. This application will avoid the manual work i.e.; people can easily communicate or coordinate with the deaf or dumb people.

There are some reforms that we can do for future implications that are: Incorporation of alternative sign languages. A mobile application that can do a similar. This work will be extended to the applications of dominant of robot, video gaming etc. When no image is supplied or an illegal image is providing then too the CNN would classify the false image into a class. This can be the most important drawback in using CNN. Reducing false positive would also be a priority work. The dataset needs to be diverse enough in order to accommodate people of all skin tones and in all environments. A bias in data could possibly disadvantage deaf people of a certain ethnic group.

REFERENCES

[1] Pratibha Pandey, Vinay Jain, "Hand Gesture Recognition for Sign Language Recognition: A Review", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 3, March 2015.
[2] Neelam K. Gilorkar, Manisha M. Ingle, "Real Time Detection and Recognition of Indian And American Sign Language Using Sift", International
[3] Journal of Electronics and Communication Engineering & Technology (IJECET), Volume 5, Issue 5, pp. 11-18, May 2014.
[4] Wario, Ruth & Nyaga, Casam. (2018). Sign Language Gesture Recognition through Computer Vision.
[5] Vivek Bheda & Dianna Radpour Using Deep Convolutional Networks for Gesture Recognition in American Sign Language (arXiv:1710.06836)
[6] https://tensorflow.org/versions/master/tutorials/mnist/beginners/index.html

