Abstract

Inability to speak is considered to be true disability. People with this disability use different modes to communicate with others, there are a number of methods available for their communication one such common method of communication is sign language. Developing sign language application for deaf people can be very important, as they'll be able to communicate easily with even those who don't understand sign language. Our project aims at taking the basic step in bridging the communication gap between normal people and deaf and dumb people using sign language.

The main focus of this work is to create a vision based system to identify Finger spelled letters of ASL. The reason for choosing a system based on vision relates to the fact that it provides a simpler and more intuitive way of communication between a human and a computer.

We used two approaches for the classification of sign language :

1. In the first approach, features were extracted from the images using SIFT(scale invariant vector transform) which were then plotted into histograms and used for training hierarchical SVM.The accuracy of the model obtained using this approach was 44.259%.
2. In the second approach, Convolutional neural network was used. The accuracy of the model obtained using Convolutional Neural Network was 95.50.

Introduction

Deaf is a disability that impair their hearing and make them unable to hear, while mute is a disability that impair their speaking and make them unable to speak . Both are only disabled at their hearing and/or speaking, therefore can still do much other things. The only thing that separate them and the normal people is communication. If there is a way for normal people and deaf-mute people to communicate, the deaf-mute people can easily live like a normal person. And the only way for them to communicate is through sign language. While sign language is very important to deaf-mute people, to communicate both with normal people and with themselves, is still getting little attention from the normal people. We as the normal people, tend to ignore the importance of sign language, unless there are loved ones who are deaf-mute. One of the solution to communicate with the deaf-mute people is by using the services of sign language interpreter. But the usage of sign language interpreter can be costly. Cheap solution is required so that the deaf-mute and normal people can communicate normally.

Therefore, researchers want to find a way for the deaf-mute people so that they can communicate easily with normal person. The breakthrough for this is the Sign Language Recognition System. The system aims to recognize sign language, and translate it to the local language via text or speech. However, building this system cost very much and are difficult to be applied for daily use. Early researches have known to be successful in Sign Language Recognition System by using data gloves. But, the high cost of the gloves and wearable character make it difficult to be commercialized. Knowing that, researchers then try to develop a pure vision Sign Language Recognition Systems. However, it is also coming with difficulties, especially to precisely track hands movements.

American Sign Language (ASL) substantially facilitates communication in the deaf community. However, there are only ~250,000-500,000 speakers which significantly limits the number of people that they can easily communicate with. The alternative of written communication is cumbersome, impersonal and even impractical when an emergency occurs.

In order to diminish this obstacle and to enable dynamic communication, we present an ASL recognition system that uses Convolutional Neural Networks (CNN) in real time to translate a video of a user's ASL signs into text. Our problem consists of three tasks to be done in real time:

1. Obtaining video of the user signing (input)
2. Classifying each frame in the video to a letter

From a computer vision perspective, this problem represents a significant challenge due to a number of considerations, including:

- Environmental concerns (e.g. lighting sensitivity, background, and camera position)
- Occlusion (e.g. some or all fingers, or an entire hand can be out of the field of view)

Our system takes video of a user signing a letter as input through a webcam. We then extract individual frames of the video and generate letter probabilities for each using a CNN. Finally, we use a language model in order to output a likely letter to the user.

Problem Statement

Aim : The deaf people can not speak like normal human beings so they communicate with the help of sign language. One of the majorly known sign language is American Sign Language which is used very widely throughout by deaf people. So our aim is to diminish the communication barrier between such person and normal human beings. Thus the aim is to develop a system through which one can easily communicate with deaf people. Our system provides the functionality of converting the sign language into text.

.

Literature Survey

Sign language recognition using image based hand gesture recognition techniques

The paper aims to present a real time system for recognition of hand gesture on basis of detection of some shape based features like orientation, Centre of mass centroid, fingers status, thumb in positions of raised or folded fingers of hand.

Gesture recognition is gaining importance in many applications areas such as human interface, communication, multimedia and security. Typically Sign recognition is related as image understanding. It contains two phases: sign detection and sign recognition. Sign detection is an extracting feature of certain object with respect to certain parameters. Sign recognition is recognizing a certain shape that differentiates the object from the remaining shapes. Language, especially in the cases when no alternative communication is available. The technical point of view characteristic features of sign language communication are: its social direction and meaning; technical and technological convenience and easy to use. The system will use a webcam for capturing the images and pre-processing of the signs will be done by using Microsoft Visual Studio as an IDE and OpenCv library. On having the input sequence of images captured through web-cam here uses some image preprocessing steps for removal of background noise and employs slope distance based algorithm i.e. Fingertip Detection by convexity hull algorithm which generates a ratio with the help of which a template of the captured image is generated.

Sign language recognition using sensor gloves

This paper examines the possibility of recognizing sign language gestures using sensor gloves. Previously sensor gloves are used in games or in applications with custom gestures. This paper explores their use in Sign Language recognition. This is done by implementing a project called "Talking Hands", and studying the results. The project uses a sensor glove to capture the signs of American Sign Language performed by a user and translates them into sentences of English language. Artificial neural networks are used to recognize the sensor values coming from the sensor glove. These values are then categorized in 24 alphabets of English language and two punctuation symbols introduced by the author. So, mute people can write complete sentences using this application.

Sensor gloves are normally gloves made out of cloth with sensors fitted on it. Using data glove is a better idea over camera as the user has flexibility of moving around freely within a radius limited by the length of wire connecting the glove to the computer, unlike the camera where the user has to stay in position before the camera. This limit can be further lowered by using a wireless camera. The effect of light, electric or magnetic fields or any other disturbance does not effect the performance of the glove.

This project was meant to be a prototype to check the feasibility of recognizing sign languages using sensor gloves. The completion of this prototype suggests that sensor gloves can be used for partial sign language recognition.

Dataset

Dataset was taken from two different sources which provided different types of images.

Dataset 1:

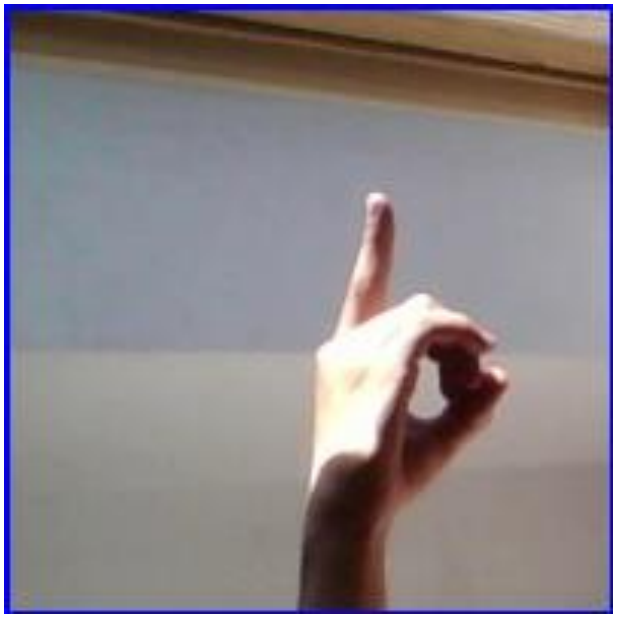
Size: 78000 images

Number of classes: 26 (A-Z alphabets)

Resolution of each image: 200 * 200

Images per class: 3000

In this dataset the images are non segmented images taken with webcam.



Dataset 2:

Size: 2671 images

(2517 images from dataset + 154 images of taken manually from camera)

Number of classes: 36 (A-Z alphabets, 0-9 digits)

Resolution of each image: 600 * 670

Images per class: 60

In this dataset the images were segmented images.

Methodology and Implementation

Machine Learning Approach

1. SIFT

Scale invariant feature transform (SIFT) is an algorithm in computer vision to detect and describe local features in images. The algorithm was patented in the US by the University of British Columbia and published by David Lowe in 1999.

Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving. SIFT can robustly identify objects among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes.

The ScaleInvariant Feature Transform (SIFT) bundles a feature detector and a feature descriptor. The detector extracts from an image a number of frames (attributed regions) in a way which is consistent with (some) variations of the illumination, viewpoint and other viewing conditions. The descriptor associates to the regions a signature which identifies their appearance compactly and robustly.

2. K-MEANS CLUSTERING

The k-means algorithm takes as input the number of clusters to generate, k, and a set of observation vectors to cluster. It returns a set of centroids, one for each of the k clusters. An observation vector is classified with the cluster number or centroid index of the centroid closest to it. A vector v belongs to cluster i if it is closer to centroid i than any other centroids. If v belongs to i, we say centroid i is the dominating centroid of v. The k-means algorithm tries to minimize distortion, which is defined as the sum of the squared distances between each observation vector and its dominating centroid. Each step of the k-means algorithm refines the choices of centroids to reduce distortion. The change in distortion is used as a stopping criterion: when the change is lower than a threshold, the k-means algorithm is not making sufficient progress and terminates. One can also define a maximum number of iterations. The centroid index or clustered index is also referred to as a “code” and the table mapping codes to centroids

and vice versa is often referred as a “code book”. The result of k-means, a set of centroids, can be used to quantize vectors. Quantization aims to find an encoding of vectors that reduces the expected distortion. As an example, suppose we wish to compress a 24-bit color image (each pixel is represented by one byte for red, one for blue, and one for green) before sending it over the web. By using a smaller 8bit encoding, we can reduce the amount of data by two thirds. Ideally, the colors for each of the 256 possible 8bit encoding values should be chosen to minimize distortion of the color. Running k-means with k=256 generates a codebook of 256 codes, which fills up all possible 8-bit sequences. Instead of sending a 3-byte value for each pixel, the 8bit centroid index (or code word) of the dominating centroid is transmitted. The code book is also sent over the wire so each 8-bit code can be translated back to a 24-bit pixel value representation.

Three key features of k-means which make it efficient are often regarded as its
biggest drawbacks:

- Euclidean distance is used as a metric and variance is used as a measure of cluster scatter.
- The number of clusters k is an input parameter: an inappropriate choice of k may yield poor results. That is why, when performing k-means, it is important to run diagnostic checks for determining the number of clusters in the data set .
- Convergence to a local minimum may produce counterintuitive ("wrong") results.

3. SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyperplane that differentiate the two classes very well. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier . There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one

that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum margin hyperplane and the linear classifier it defines is known as a maximum margin classifier; or equivalently, the perceptron of optimal stability.

Deep Learning Approach

1. LeNet5

The LeNet-5 architecture consists of two sets of convolutional and average pooling layers, followed by a flattening convolutional layer, then two fully-connected layers and finally a softmax classifier.

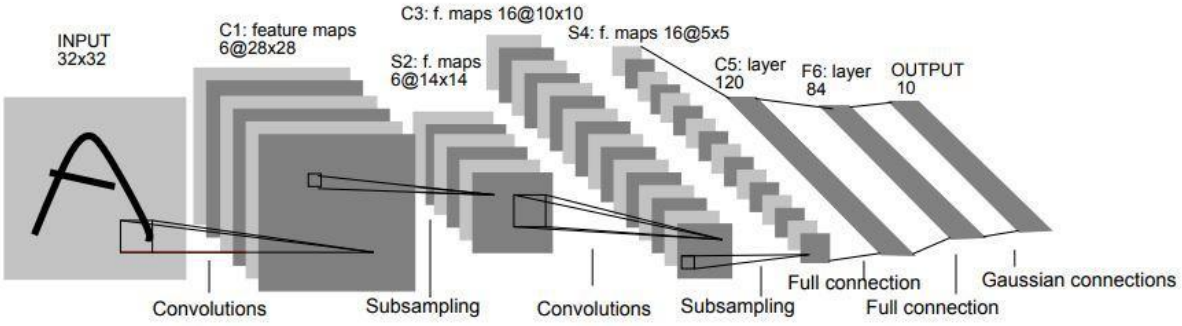


Fig. 1: LeNet5 Architecture

We have used LeNet5 architecture to our problem statement and gets the validation accuracy of 95.48%%. But the model does not predict accurately when given test images.

2. AlexNet

The AlexNet architecture contains 5 convolutional layers and 3 fully connected layers. Relu is applied after very convolutional and fully connected layer. Dropout is applied before the first and the second fully connected year.

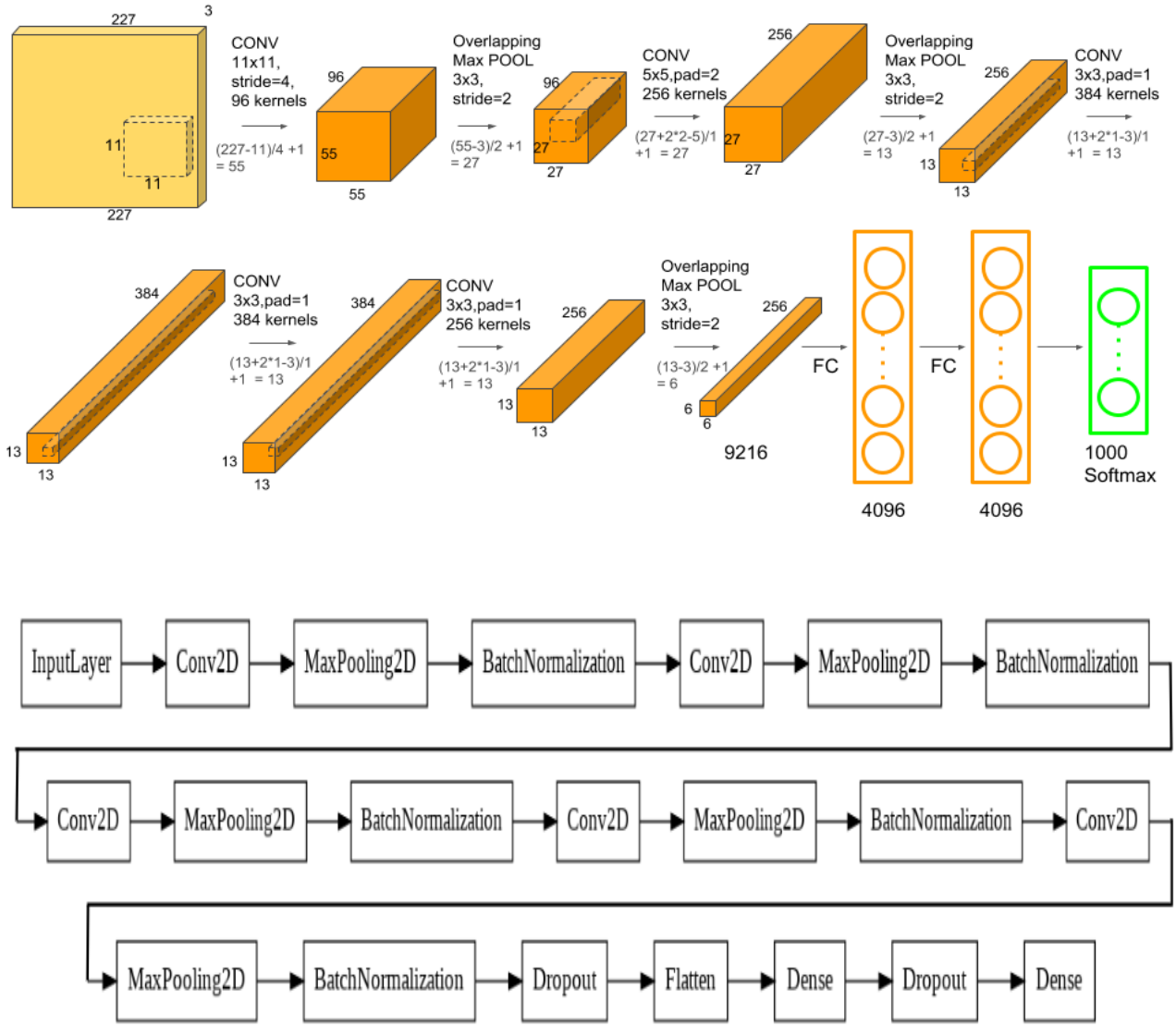


Fig 2: AlexNet and Modified AlexNet Architecture

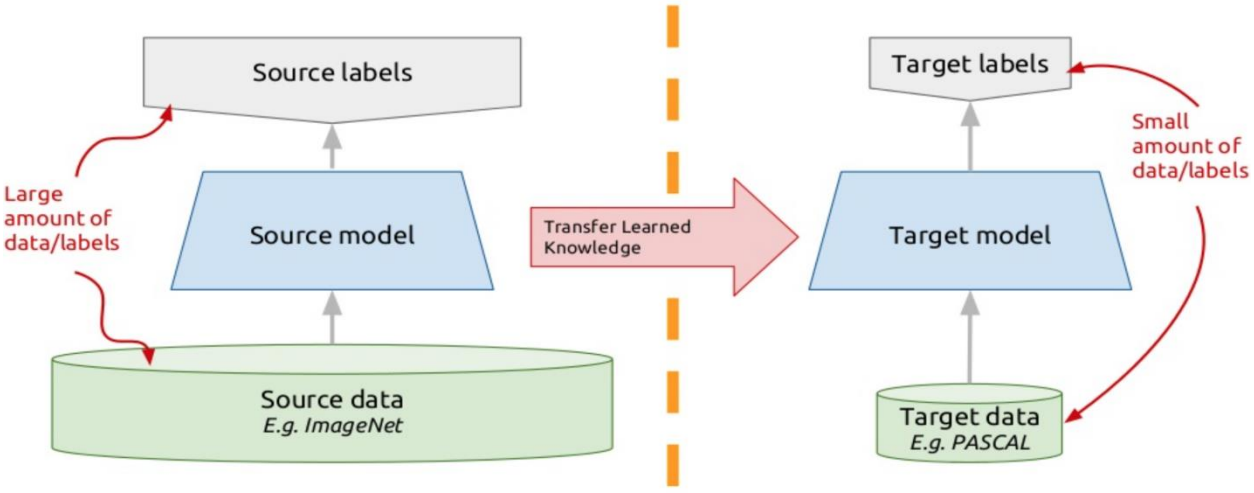
Modification done to original architecture:

Total 5 Convolution Layer and 3 Dense Layer

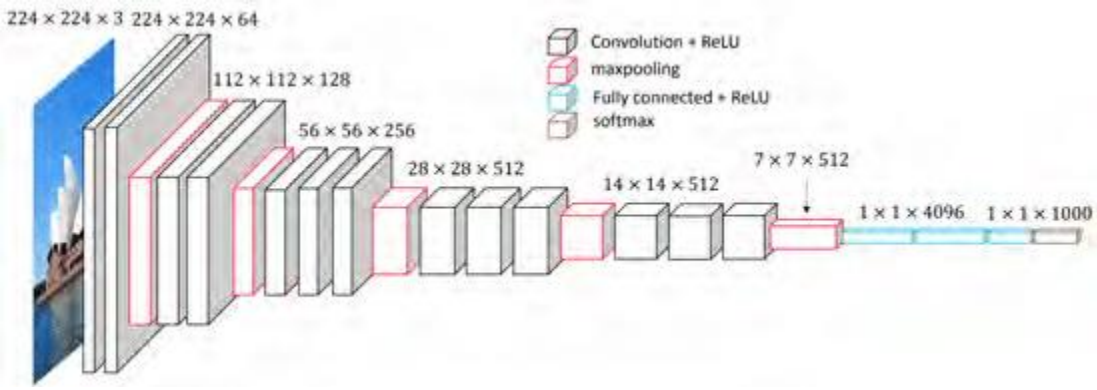
- I Layer: #Filters - 32, Kernel size - 5
- II Layer: #Filters - 64, Kernel size - 5
- III Layer: #Filters - 64, Kernel size - 5
- IV Layer: #Filters - 96, Kernel size - 5
- V Layer: #Filters - 32, Kernel size - 5
- I Dense Layer - #Neurons - 128
- II Dense Layer - #Neurons - 26/36 (Softmax Activation)

Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. In transfer learning, we first train a base network on a base dataset and task, and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specific to the base task.



3. VGGNet16



VGGNet consists of 16 convolutional layers and is very appealing because of its very uniform architecture. Similar to AlexNet, only 3x3 convolutions, but lots of filters. Trained on 4 GPUs for 2–3 weeks. It is currently the most preferred choice in the community for extracting features from images. The weight configuration of the VGGNet is publicly available and has been used in many other applications and challenges as a baseline feature extractor.

Segmentation

In computer vision segmentation is the digital image into multiple segments. The goal of segmentation is to simply add or change the representation of the image that is more meaningful and easy to analyze.

Segmentation was done using three approaches:

- Applying constraints on HSV and YCbCr colormaps.
- Then fed the output to Watershed algorithm.

Before Segmentation:

After Segmentation:

Results:

Model	Dataset	Train-Validation-Test Split	Train Accuracy	Validation Accuracy	Test Accuracy	K-Fold Cross Validation
LeNet	2517 images(36 images)	80-15-5	100%	95.48%	96.50%	
Modified AlexNet	2517 images(36 classes)	64-16-20	100%	99.75%	100%	97.91% (+/- 1.55%) (K=5)
	78000 images (26 classes)	80-15-5	96%	Only k-fold validation done	51%	73.75% (+/- 3.80%) (K=5)
	2517+154(real and segmented)	64-16-20	100%	95%	95%(unseen data from the same dataset) & 72%(154 real images that were added in training)	91.37% (+/-2.43%)(K=5)
	78000 images (26 classes)	72-18-10	91%	Only k-fold validation done	93.31%(unseen data from the same dataset) & 53%(154 real images that were added in training)	79.4% (+/- 2.60%)(K=5)

Conclusion and Future Work:

We implemented and trained an American Sign Language recognition system based on a CNN classifier. We were able to produce satisfactory results on character recognition using 78000 images dataset. Because of the lack of variation in our datasets, the validation accuracies we observed during training were not directly reproducible upon testing on real-time images. This project can be further extended for the recognition of words, thus formation of sentences by the gestures expressed by dumb people. An Android or IOS app can be developed for it to be easily usable by everyone as it does not use any depth sensors and only requires a working smartphone camera. With the help of a dataset with images taken in different environmental conditions, the models would be able to generalize with considerably higher efficacy and would produce a robust model for all letters.