Udacity Machine Learning Nanodegree
Capstone Proposal

Sign Language Recognition-Computer Vision

Mohamed Gamal

August 2019

1- Domain Background

According to the World Health Organization, over 6% of the world's population (i.e. around 455 million people: 385 millions of deaf, 70 millions of mute have disabling hearing loss or speech problems, the majority of those people live in low and middle-income countries, and in Egypt alone there are around 7.5 million.
A person who is not able to hear as well as someone with normal hearing –thresholds of 25 dB or better in both ears – is said to have hearing loss. Hearing loss may be mild, moderate, severe or profound. It can affect one ear or both ears, and leads to difficulty in hearing conversational speech or loud sounds. 'Deaf' people mostly have profound hearing loss, which implies very little or no hearing. They often use sign language for Communication. So, this project is considered as a contribution to solve the problem of deaf-mute people, where deaf-mute is term continues to be used to refer to deaf people who cannot speak an oral language or have some degree of speaking ability, such people communicate using sign language.

The academic research relevant to this domain such as :

- http://cs231n.stanford.edu/reports/2016/pdfs/214_Report.pdf
- https://www.datacamp.com/projects/509 this project help me to build the model.

2- Problem Statement

The main objective of this project is to develop an automatic system used in recognizing the sign language of the hand that means much for the speechless and deaf persons.

The proposed system will make it easier to communicate with deaf persons by converting their sign language to text or sound. The project has the following specific aims:

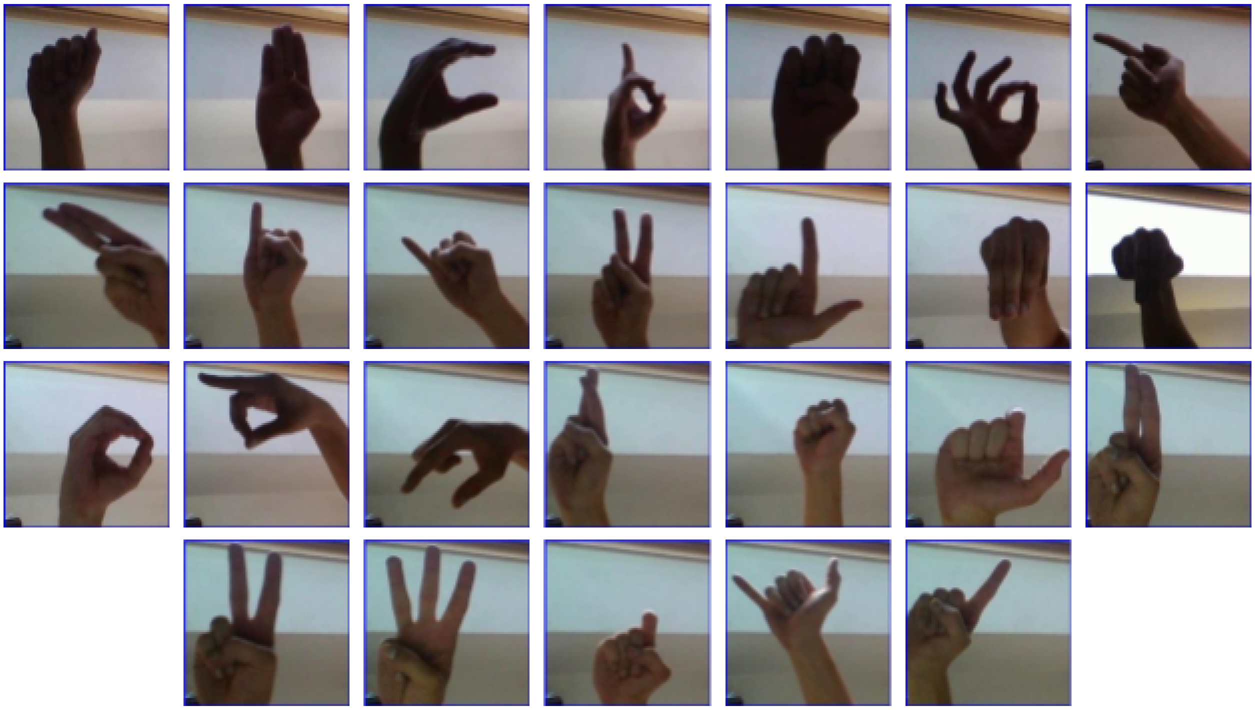
1. Low cost.
2. High accuracy.
3. Easy to use.
4. No physical devices attached to deaf-mute people.
5. Ease of communications between deaf-mute and other people using the developed system.

3- Datasets and Inputs

The data set might be the most important part in any detection system.The data set consists of 44 different gestures which are:

- Letters e.g(A,B,C,...).
- Numbers e.g (0,1,2,...,9).
- Words e.g (love ,peace,like,...).

A set of 1200 images per gesture (50 x 50 pixel each) are collected using a computer camera.The process of collecting data for every gesture is based on capturing images by the camera during streaming video, in a span of 10-15 seconds.This dataset will be split into 80:20 for training(80%) and testing(20%).

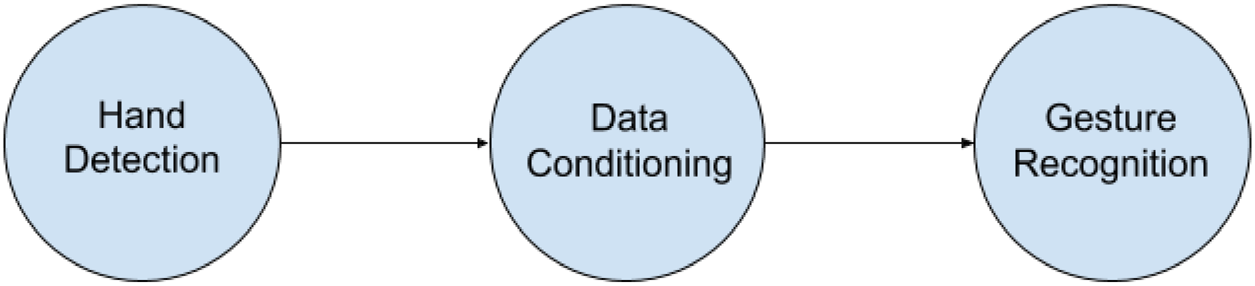


A sample gesture of each letter

4- Solution Statement

The proposed solution to this problem is to apply Deep Learning techniques that have proved to be highly successful in the field of image classification.

we can divide the project into three major steps which represent the major objectives in the project as shown in figure below.



5- Benchmark Model

Choosing a suitable network model After collecting the needed dataset and finishing its preprocessing, we are now ready to construct our model. Since we are dealing with images, the neural network we are going to build will be a convolutional neural network.

The proposed convolutional neural network has to be relatively small in order to perform calculations in a real-time fashion, hence we will build a network that is similar to one of the MNIST classifying models, and consists of only 9 layers. The details of each layer are shown in **Fig 5.1**.

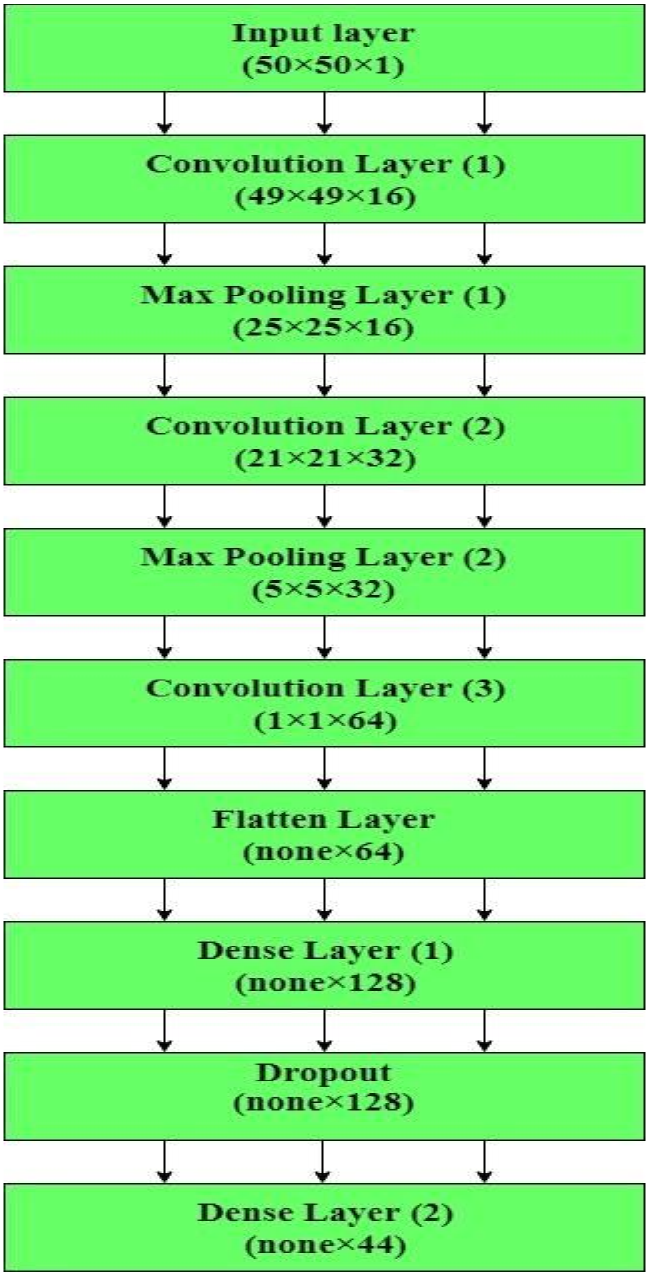

Input layer (50×50×1)
Convolution Layer (1) (49×49×16)
Max Pooling Layer (1) (25×25×16)
Convolution Layer (2) (21×21×32)
Max Pooling Layer (2) (5×5×32)
Convolution Layer (3) (1×1×64)
Flatten Layer (none×64)
Dense Layer (1) (none×128)
Dropout (none×128)
Dense Layer (2) (none×44)

Figure 5.1: Representation of the proposed network layers.

The model is trained for 1 epoch. The loss function that is chosen to assess the model accuracy is Cross Entropy, while the optimization method that is chosen is Stochastic Gradient Descent.

In **Fig** 5.2, we can see the model accuracy during training at each iteration, while in **Fig** .5.3, we can see the model accuracy during testing at each iteration.

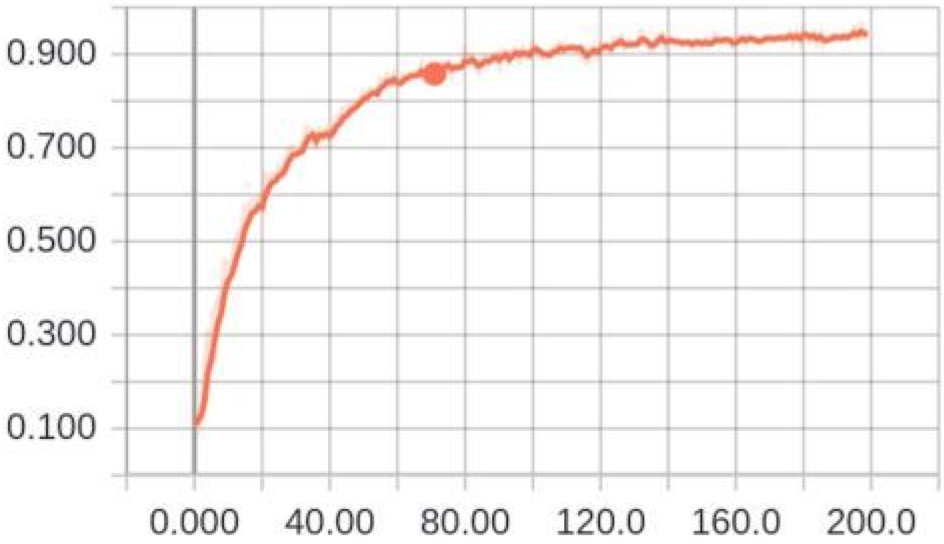


Figure 5.2 Training accuracy (vertical axis is accuracy, horizontal axis is no. of iterations).

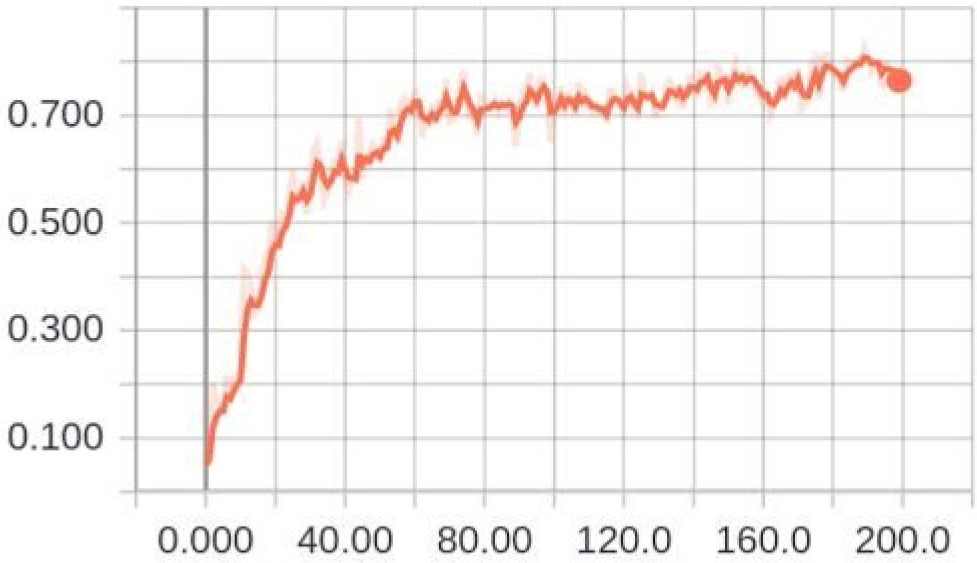


Figure 5.3 Testing accuracy (vertical axis is accuracy, horizontal axis is no. of iterations).

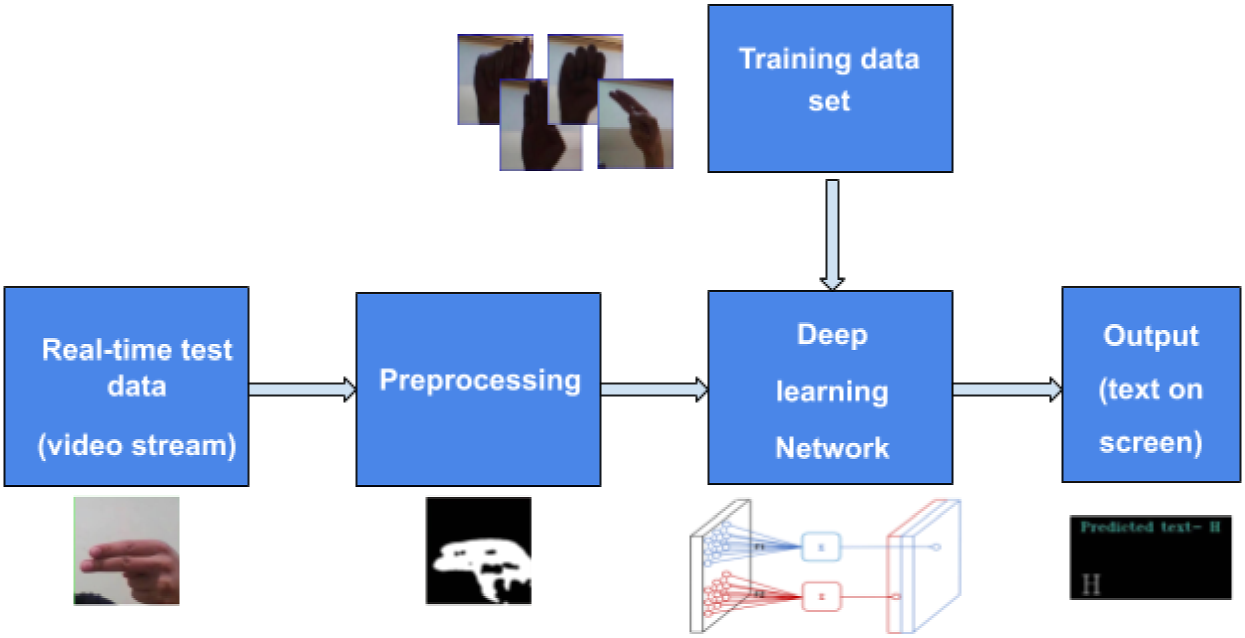
the model's accuracy produced after the end of training is around 92%. And the accuracy produced after using the testing data set is around 78%, which presents a valid assessment about the process of training and that the model results are reliable.

6- Evaluation Metrics

The evaluation metric for this problem is simply the Accuracy Score.

7- Project Design

We will start by defining the main structuring blocks of our system. This is shown in figure below.



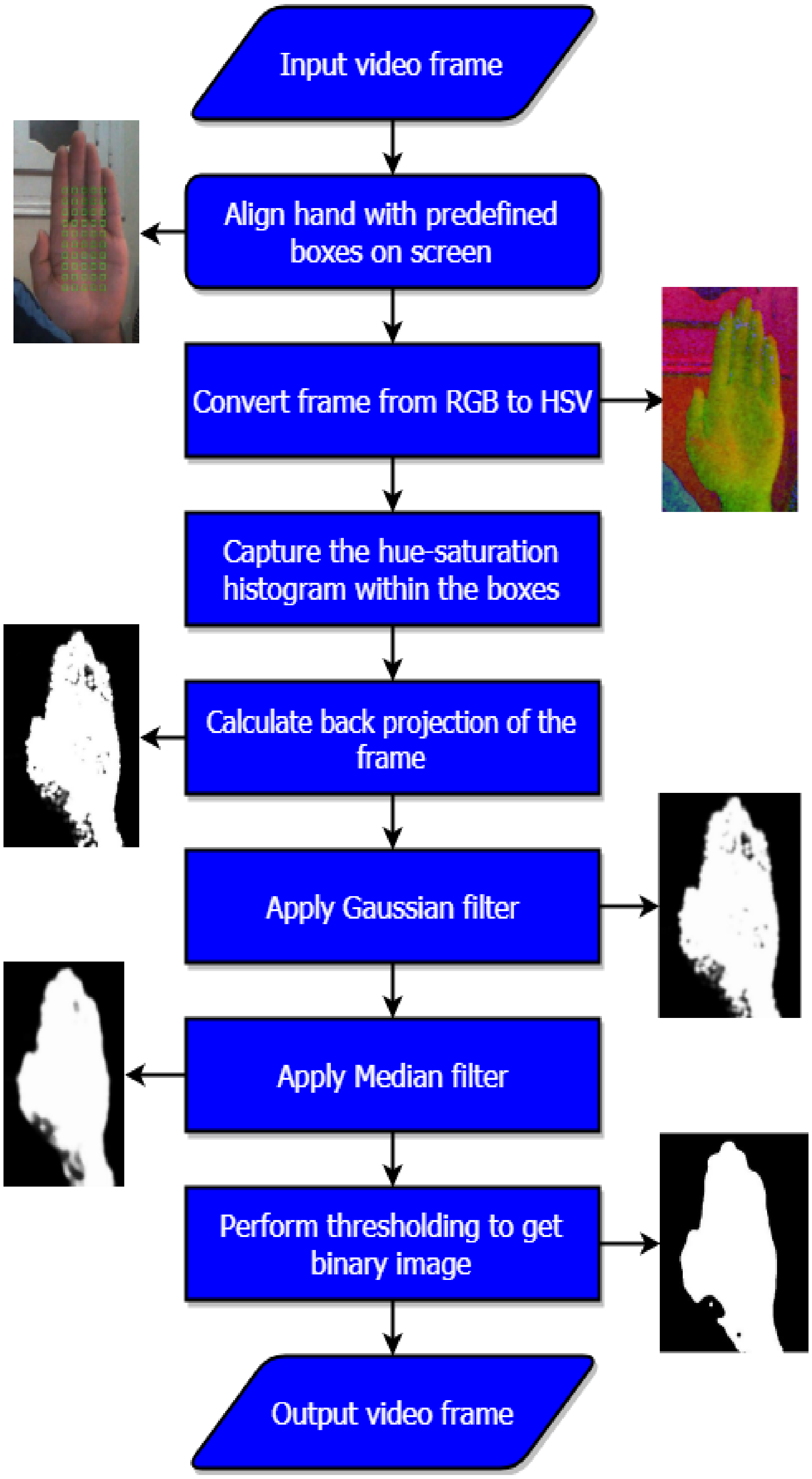
Real-time test data

The testing data enters the system through a video feed taken by a camera. The video is processed frame-by-frame by our system to identify the gesture shown on the video.
To avoid any unnecessary objects in the background. We defined a certain box shown on screen. The processing operation are done only inside that box.

Preprocessing

The objective of the preprocessing step is to separate the hand from the background. We are going to use multiple image processing and filtering techniques in order to get a binary image where the hand gesture is shown in white, and all its background is shown in black.

The techniques and methods used during preprocessing are shown in the flowchart in **Figure** below. We will discuss each step separately and explain their importance for our goal.


Input video frame
Align hand with predefined boxes on screen
Convert frame from RGB to HSV
Capture the hue-saturation histogram within the boxes
Calculate back projection of the frame
Apply Gaussian filter
Apply Median filter
Perform thresholding to get binary image
Output video frame

Training data set

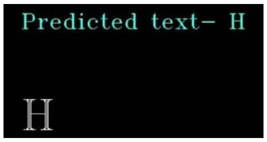
As we said before The data set consists of 44 different gestures which are:

- Letters e.g(A,B,C,...).
- Numbers e.g (0,1,2,...,9).
- Words e.g (love ,peace,like,...).

A set of 1200 image per gesture (50 x 50 pixel each) are collected using a computer camera.The process of collecting data for every gesture is based on capturing images by the camera during streaming video, in a span of 10-15 seconds.Also, we will do some data augmentation such as flip , right shifted , left shifted and so on. Then, The dataset will be increased to become 2400 image per gesture.

Output text

After the input data is processed and classified by the neural network, the gesture will be translated and written as a text on screen.



```
Predicted text- H

H
```

the gesture is recognized and translated to text on screen.