**Gesture recognition using NN**

**Problem Statement**

You are working as a data scientist at a leading electronics company. As an addition to the smart features of the televisions, you are given the responsibility to develop a model that can recognize 5 different gestures listed below which will help the customers in navigating through the **SMART TV** without using the remote.

A major application of gesture recognition will be used while watching a movie in the

Following manner:

1. Thumbs Up: Increase the volume

2. Thumbs Down: Lower the volume

3. Left Swipe: Fast Forward 10 seconds

4. Right Swipe: Fast Backward 10 seconds

5. Stop: Pause the movie

**Dataset Interpretation**

The dataset contains 30 frames of a 2 second long video. Since it is an image videos.

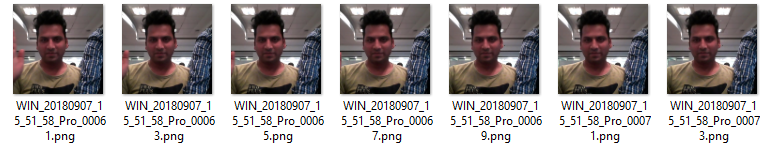
**Folders**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Folder Image** | | **Classes** | |
|  | | Left\_Swipe\_new;0  Right\_Swipe\_new;1  Stop\_new;2  Thumbs\_Down\_new;3  Thumbs\_Up\_new;4 | |
|  | |  | |
| Train.csv | Folders | | Images |
|  |  | |  |
|  |  | |  |

Observation:

1. Since these are images from videos some of the images in the beginning and end will not make sense for gesture identification?

For example : Here hand not visible.



1. May be video based deep learning will give better result here?
2. All image size is not same, so we need to normalize those

## **Processing the data**

We can use Image processing techniques like:

* Converting image size into different size:

|  |  |
| --- | --- |
| Image | Code |
|  |  |

* Normalize the image

|  |  |
| --- | --- |
| Image | Code |
|  |  |

* Flip Image
* Edge detection
* Crop image : We can crop unwanted portion of image etc.

|  |  |
| --- | --- |
| Image | code |
|  |  |

**I**

**Image augmentation:** We can use image augmentation for  artificially creating training images through different ways like such as random rotation, shifts, shear and flips, etc. and combination of same. Here I did not do anything as the data size is so huge and it was taking long to process.

### **Generator:** We can use python generators (which will have a yield) to avoid \_\_iter\_\_() and \_\_next\_\_() and get the control and control the caller and executers.

## **3D Convolutional Network, or Conv3D**

I have used 3D convolutional Neural network for this assignment. To use 2D convolutions, we first convert every image into a 3D shape : width, height, channels. Channels represents the slices of Red, Green, and Blue layers. So it is set as 3. In the similar manner, we will convert the input dataset into 4D shape in order to use 3D convolution for : length, breadth, height, channel (r/g/b).

Also used **Adam** optimizer with its default settings.

I tried to use below model but since it was running long and colab was crashing multiple time. So Cut down many things.



