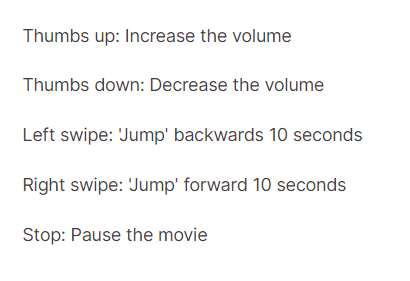
**Gesture Recognition Case study IIITB Assignment**

Developed By:

1. **Shamseena**
2. **Remella Satya Ramana Kumar**

**Problem Statement :**

Imagine you are working as a data scientist at a home electronics company which manufactures state of the art smart televisions. You want to develop a cool feature in the smart-TV that can recognise five different gestures performed by the user which will help users control the TV without using a remote.



**Objectives:**

**Generator**: The generator should be able to take a batch of videos as input without any error. Steps like cropping, resizing and normalization should be performed successfully.

**Model**: Develop a model that is able to train without any errors which will be judged on the total number of parameters (as the inference(prediction) time should be less) and the accuracy achieved. As suggested by Snehansu, start training on a small amount of data and then proceed further.

**Write up**: This should contain the detailed procedure followed in choosing the final model. The write up should start with the reason for choosing the base model, then highlight the reasons and metrics taken into consideration to modify and experiment to arrive at the final model.

# Understanding the Dataset:

The training data consists of a few hundred videos categorized into one of the five classes. Each video (typically 2-3 seconds long) is divided into a **sequence of 30 frames (images)**. These videos have been recorded by various people performing one of the five gestures in front of a webcam - similar to what the smart TV will use.

**A picture containing photo, many, various, sitting

Description automatically generated**

**Model Analysis :**

|  |  |  |
| --- | --- | --- |
| **Model 1** | **Frames = 16, Epoch = 10, Batch Size = 8** | **Ideal Model** |
| **Model 2** | **Frames = 30, Epoch = 10,Batch Size = 16** | **Ideal Model** |
| **Model 3** | **Frames = 30, Epoch = 20, Batch Size = 64** | **Reduced parameters** |
| **Model 4** | **Frames = 20, Epoch = 20 , Batch Size = 64,** | **Same pooling in Conv3D layer** |
| **Model 5** | **Frames = 20, Epoch = 20, Batch Size = 64** | **Reduced kernel to (2,2,2), switching BatchNormalization before MaxPooling.** |
| **Model 6** | **Frames = 20 , Epoch = 10, Batch Size = 32** | **Con2D and LSTM** |

**Final Model**

* *Switching Model architecture to Conv2D+LSTM*
* *Training Accuracy 0.7759103775024414*
* *Validation Accuracy 0.8125*
* *We have achieved 81.25 percent validation accuracy which is a good sign.*

**End**