# Gesture Recognition Case Study

**Problem Statement:**

We need 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.

The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

- Thumbs up:  Increase the volume
- Thumbs down: Decrease the volume
- Left swipe: 'Jump' backwards 10 seconds
- Right swipe: 'Jump' forward 10 seconds
- Stop: Pause the movie

**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.

The following table consists of the experiments done to build a model to predict the gestures from the given data set.

| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
|---|---|---|---|
| **1** | **Conv3D** | **Train Accuracy: 0.15,**<br>**Validation Accuracy: 0.15** | **The model isn't showing any improvement during training, as the loss remains unchanged across epochs. Hence** |

| | | | |
|---|---|---|---|
| | | | **the batch size reduced further to address this** |
| **2** | **TimeDistributed Conv2D + GRU** | **Train Accuracy: 0.9554, Validation Accuracy: 0. 8203** | **Reduce the size of the image/Reduce the number of layers along with dropout layer** |
| **3** | **Time Distributed + ConvLSTM2D** | | **This seems to be the best model we've achieved so far. The validation accuracy is solid, with 13,589 parameters. Additionally, the model size is quite smaller compared to other models.** |

Please note that, due to storage limitations on the JarvisLab instances, I was unable to continue training the final model, which also caused a delay in the submission.