## Neural Networks Project - Gesture Recognition

**|! Nilesh Raut !|**

**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.

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

# Two Architectures: 3D Convs and CNN-RNN Stack:

For analysing videos using neural networks, **two types of architectures** are used commonly. One is the standard **CNN + RNN architecture** in which you pass the images of a video through a CNN which extracts a feature vector for each image, and then pass the sequence of these feature vectors through an RNN.

A Conv3D network processes three-dimensional data, like videos or 3D images, using 3D convolutional layers, pooling, and activation functions. It extracts spatial and temporal features through sliding filters, reduces dimensionality with 3D pooling, and employs fully connected layers for high-level reasoning. With applications in video analysis, action recognition, and medical imaging, Conv3D networks capture intricate spatiotemporal patterns in data for tasks involving volumetric information.

**Data Generators:**

Data generators are essential components in machine learning, particularly in scenarios where dealing with large datasets or continuous data streams is required. They are used to efficiently load, preprocess, and feed data to machine learning models, often in batches, without loading the entire dataset into memory at once. This is especially useful when working with datasets that are too large to fit into memory or when data is continuously generated.

| **Model Number** | **Model** | **Result** | **Explaination** |
| --- | --- | --- | --- |
| **1** | Conv3D | Training: 39%  Validation: 46% | with batch size: 8, image size: 100, frames: 15 |
| **2** | Training: 53%  Validation: 58% | Redused the size of the Image from 100 to 50 |
| **3** | Training: 58%  Validation: 77% | Reduced the size of the Image from 50 to 25 |
| **4** | Training: 55%  Validation: 68% | Decreased Batch from 8 to 4 |
| **5** | Training: 57%  Validation: 72% | Changed Optimizer from **SGD** to **Adam** |
| **6** | Training: 58%  Validation: 62% | Reduced Number of Frames 15 to 10 |
| **7** | Training: 09%  Validation: 85% | Reduced Dropout Rate from 0.5 to 0.25 |
| **8** | Training: 92%  Validation: 84% | Increased Number of epochs from 25 to 35 |