**Gesture Recognition Case Study**

* **Jitendar Nath Vyas - Group Facilitator**
* **Sachin Kumar Kesharwani**

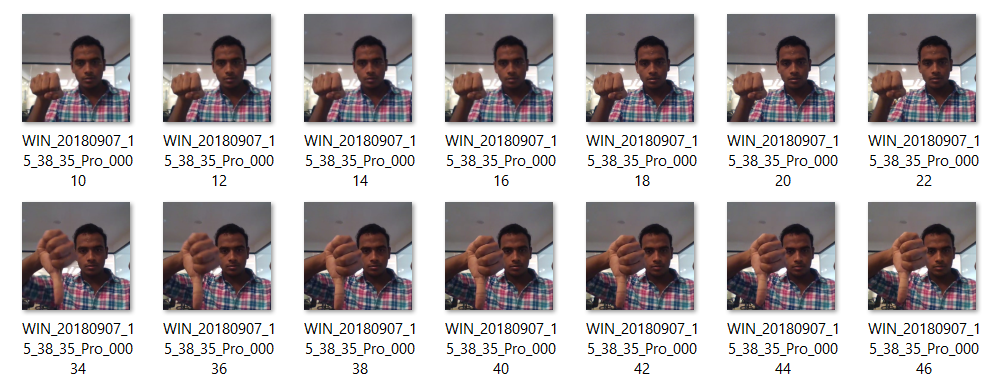
# Problem Statement

As a data scientist at a home electronics company which manufactures state of the art smart televisions. We 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.

* 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.

# 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.

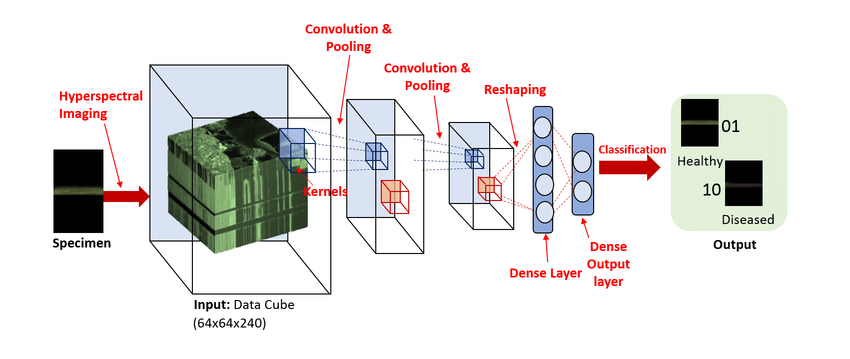
****

**Objective:**

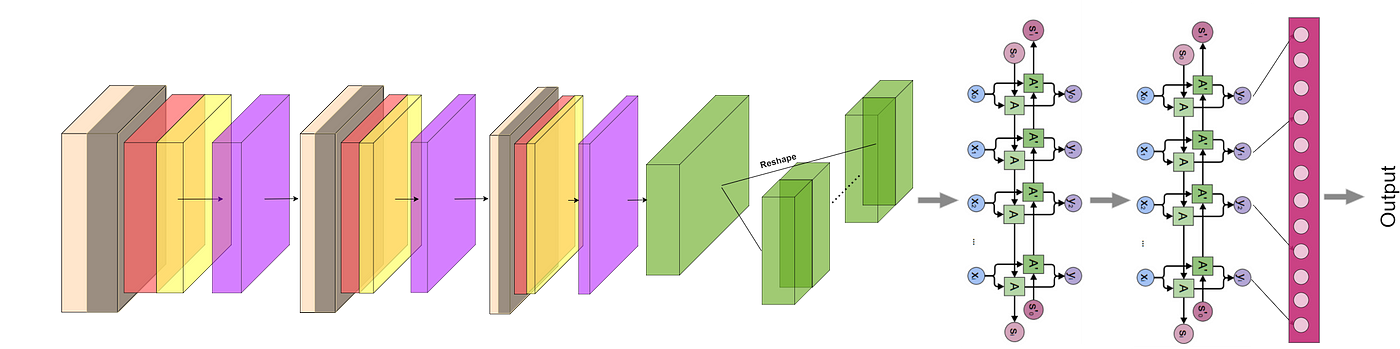
Our task is to train a model on the 'train' folder which performs well on the 'val' folder as well (as usually done in ML projects). We have withheld the test folder for evaluation purposes - your final model's performance will be tested on the 'test' set.

**Two types of architectures suggested:**

1. **3D Convolutional Neural Networks (Conv3D)**



**2. Convolutions + RNN**



**Architecture development and training:**

* Experimented with different model configurations and hyper-parameters and various iterations and combinations of batch sizes, image dimensions, filter sizes, padding and stride length were experimented with.
* We experimented with SGD() and Adam() optimizers but went forward with Adam() as it lead to improvement in model’s accuracy by rectifying high variance in the model’s parameters.
* We also made use of Batch Normalization, pooling and dropout layers when our model started to overfit, this could be easily witnessed when our model started giving poor validation accuracy despite having good training accuracy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Hyper- parameters** | **Result** | **Decision + Explanation** |
| **1** | Conv3D | 2067013 | Training Accuracy -90% Validation Accuracy-87% | Conv3D Model with Batch Normalisation and Drop Out layer. We Use n\_frames=16,num\_epochs=20,Batch Size= 64. Overall good train and validation accuracy. |
| **2** | Conv3D | 9005637 | Training Accuracy -93% Validation Accuracy-72% | Conv3D Model with Batch Normalisation and Drop Out layer. We Use n\_frames=30,num\_epochs=30,Batch Size= 64. We Observed Model Is Overfitting. |
| **3** | Conv3D | 5617637 | Training Accuracy -92% Validation Accuracy-63% | Conv3D Model with Batch Normalisation and Drop Out layer. We Use n\_frames=30,num\_epochs=20,Batch Size= 64. We Observed Model Is Overfitting. |
| **4** | Conv3D | 1907045 | Training Accuracy -92% Validation Accuracy-62% | Conv3D Model with Batch Normalisation and Drop Out layer. We Use n\_frames=20,num\_epochs=20,Batch Size= 64,Same Pooling. We Observed Model Is Overfitting. |
| **5** | Conv3D | 1300565 | Training Accuracy -92% Validation Accuracy-62% | Conv3D Model with Batch Normalisation and Drop Out layer. We Use n\_frames=20,num\_epochs=20,Batch Size= 64,Same Pooling.Reduced kernel size to (2,2,2),switching BatchNormalization before MaxPooling. We Observed Model Is Overfitting. |
| **6** | Conv2D + LSTM | 3083141 | Training Accuracy -95% Validation Accuracy-56% | CNN - LSTM model - we didn't get a good validation accuracy of 56%. |
| **7** | Transfer Learning With LSTM | 1380869 | Training Accuracy -100% Validation Accuracy-100% | Transfer Learning With LSTM we are getting 100% Traing and Validation accuracy. |

**Conclusion:**

The Model Built with Conv3D with n\_frames=16, epochs=20, batch\_size=64 are giving better results compared to all other model.