**Problem Statement**:

We want to develop a cool feature in the smart-TV that can **recognize 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

The details of our experiment are given below:

**Experiment 1**:

Here we used 4 Conv3D layers with filter size of 8, 16, 32 and 64 respectively. Each followed by a batch normalization, a relu activation and a Max pooling layer to extract features. Followed by 2 dense layers with dropout of 0.5 to avoid overfitting and a softmax activation.

***Validation Accuracy***: 57% but we need to improve it.

**Experiment 2**:

We used 2 Conv3D with 32 filters and 2 Conv3D with 64 filters followed by ‘relu’ activation, Max Pooling and drop out of 0.25. Then we added a dense layer, followed by dropout of 0.5 and softmax activation.

***Validation Accuracy***: 21%, which is very less.

**Experiment 3**:

To increase the accuracy, we changed model as explained below:

* Using 2 Conv3D layers with filter as 32 then Max Pooling to extract features and dropout of 0.25 to avoid overfitting.
* Using 2 Conv3D layers with filter as 64 then Max Pooling to extract features and dropout of 0.25 to avoid overfitting
* Using 2 Conv3D layers with filter as 64 then Max Pooling to extract features and dropout of 0.25 to avoid overfitting
* Kernel Size is (3,3,3)
* Feeding it to MLP architecture. One Dense layer with 'Relu', adding batch normalization followed by dropout of 0.5 and final with Softmax activation

***Validation Accuracy***: 41% and it is still very low.

**Experiment 4**:

So, we again created a model with 4 conv3D layers with increasing filter size and adding batch normalization followed by Max Pooling and 2 dense layers with decreasing dense and final softmax activation. We tried with batch size 10 and epoch 20.

***Validation Accuracy***: 67%, this is better. But we need to improve it by customizing the model.

**Experiment 5**:

We customized the model by changing the pool size to (3,3,3) and kernel size to (3,3,3) and adding only 1 dense layer with 512 followed by a softmax.

***Validation Accuracy:*** It decreased and became 51%. So, we decided to customize previous model to increase the accuracy.

**Experiment 6:**

We customized the previous model (Experiment 4) with batch normalization and tried out with batch size 10 and number of epochs as 30.

***Validation Accuracy:*** We still got a validation accuracy of 68%.

**Experiment 7:**

Then we created a new generator function with data augmentation and used in the following models. We created a model with 4 conv3D layers with increasing filter sizes, followed by batch normalization, relu and max pooling with pool size (2,2,2). Also, the kernel size was (3,3,3) for layer 1and 2 and (1,3,3) for layer 3 and 4. It was followed by 2 dense layers and a softmax.

Batch size is 5 and number of epochs is 20.

***Validation Accuracy:*** The accuracy increased to 84%.

**Experiment 8:**

We again reran the model with batch size 10 and number of epochs 20 to see if we can increase the accuracy.

***Validation Accuracy***: It decreased to 74.5%.

**Experiment 9:**

We again reran the model with batch size 5 and number of epochs 30 to see if we can increase the accuracy.

***Validation Accuracy***: It increased to 87%.

**Experiment 10:**

We changed the generator function to refine the image using cv2 and also used augmentation. We created a new model and ran with batch size 10 and number of epochs 20.

***Validation Accuracy***: 82.5%.

**Experiment 11**:

We again reran the model with batch size 5 and number of epochs 20 to see if we can increase the accuracy.

***Validation Accuracy***: It increased to 89%. It is the best model.

**Experiment 12**:

Then we again customized the model and tried to run but the accuracy decreased.

***Validation Accuracy***: 74%.

So, the best model is the model from Experiment 11. In the notebook it is the **Model 9** **with** **batch size 5 and number of epochs 20 with 89% accuracy**.