Gesture Recognition Problem

# Problem Statement:

 Creating A Deep Leering Model for developing 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

# Experiments with The Model:

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| 1 | Conv3D | Throws input image error | added correct input layers |
| 2 | Conv3D | OOM error | Reduced the number of images and reduced the number of batch size |
| 3 | Conv3D | Accuracy: 0.47 | Increased the number of filters and added more pooling layers and one dense layer |
| 4 | Conv3D | Accuracy: 0.63 | Added one more CNN layer with pooling increased image size |
| 5 | Conv3D | Accuracy: 0.67 | Trying ConvLSTM as Conv3D overfitting |
|  |
| 6 | ConvLSTM | Accuracy: 0.32 | Tried with one layer and max pooling |
| 7 | ConvLSTM | Accuracy : 0.54 | Added two more layers and Batch Normalization |
| Overfitting |
| 8 | Conv3D | Accuracy:0.73 | Tried with SGD optimizer and reduced the image dim to 148 from 160 pixels |
| Overfitting |
| 9 | Conv3D | Accuracy: 0.55 | Added dropouts |
| Not overfitted |
| 10 | Transfer Learning with LSTM | Accuracy: .66 | Imagenet model and LSTM |
| Less overfitting |
| 11 | TL with LSTM | Accuracy: 0.60 | Increased the LSTM filters and dense filters |
| Overfitting |
| 12 | TL with LSTM | accuracy: 0.78 | increased dropout to 0.5 |
| not overfitting |
| 13 | TL with GRU | Accuracy: 0.83 | used GRU instead of |
| 14 | TL with GRU | Accuracy: 0.92 | train using all weights |
| Final model | TL with GRU | Accuracy:0.96 | Reduced the image |

# Conclusion:

* We have Considered 15th Model i.e Transfer Learning from ImageNet where we are retraining all layers, as our final model.
* The reason being the 96% validation accuracy which was the best among the other models we trained with total parameters around 3.6M and the train time was around 42secs per epoch
* The model is approx. 42MB which is lesser than the other models that we trained