**Gesture Recognition Assignment**

# 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 has hundreds of videos, each one 2-3 seconds long and split into 30 frames. Different people recorded these videos with a webcam, doing one of the five gestures - like the smart TV will use. These videos belong to one of the five classes.

# Objective

We need to train various models on the 'train' folder that can identify the action in each sequence or video and do well on the 'val' folder too. The final test folder is hidden - we will test the final model's performance on the 'test' set.

**Consolidated final models:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model Name | Model Type | No. of Params | Augment Data | Model Size(MB) | Highest validation accuracy | Corresponding Training accuracy | Remarks |
| Conv3D Model | Conv3D | 1,736,389 | Yes | 20 | 0.31 | 0.69 | Test model. |
| Model– 1 Models with increased Batch size | Conv3D | 1,117,0611 | Yes | 41.8 | 0.29 | 0.97 | Highly overfitting |
| Model – 2 model with added dropout layers and decreased batch size | Conv3D | 3,638,981 | Yes | 20.3 | 0.43 | 0.74 | Improved in terms of overfitting |
| Model – 3 : Reduced filter size (2,2,2) and Image resolution to 120x120 | Conv3D | 1,762,613 | yes | 19.1 | 0.2 | 0.63 | Worst model. Very high overfitting |
| Model 4 : Changing to CNN and LSTM | CNN - LSTM | 1,657,445 | yes | 32.4 | 0.47 | 0.89 | Still problem with Validation score |
| Model 5: | CNN-LSTM with GRU | 2,573,925 | yes | 29.6 | 0.25 | 0.75 | Not good, Overfitting |

Contributor - [**Sreehari G Varma**]

- [**Bhargav**)( <https://github.com/bhagmuniverse>)

# Observations

* Overfitting was a major issue, which we could not address completely
* *CNN+LSTM* based model had better performance than *Conv3D.*
* *Data Augmentation* helped in overcoming the problem of overfitting which our initial version of model was facing.
* It was observed that as the Number of trainable parameters increase, the model takes much more time for training.
* Increasing the batch size greatly reduces the training time but this also has a negative impact on the model accuracy. If we want our model to be ready in a shorter time span, choose larger batch size else you should choose lower batch size if you want your model to be more accurate.
* Suggestions for improvement:
  + CNN-*LSTM* appears to be a good choice. Trainable Parameters of a *GRU* are far less than that of a *LSTM*. Therefore would have resulted in faster computations. However, its effect on the validation accuracies could be checked to determine if it is actually a good alternative over LSTM.
  + Experimenting with other combinations of hyperparameters like, activation functions (*ReLU, Leaky ReLU, mish, tanh, sigmoid*), other optimizers like *Adagrad()* and *Adadelta()* can further help develop better and more accurate models. Experimenting with other combinations of hyperparameters like the *filter size, paddings, stride\_length, batch\_normalization, dropouts* etc. can further help improve performance.