**PROBLEM STATEMENT**:

Develop a 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 Goal of this project is to build a model to recognize 5 hand gestures using different architectures (CNN+RNN, CONV3D) that can be used as feature for the Smart TV.

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

**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).

**ARCHITECTURES USED:**

**CNN + RNN:**

The conv2D network will extract a feature vector for each image, and a sequence of these feature vectors is then fed to an RNN-based network. The output of the RNN is a regular softmax (for a classification problem such as this one)

1. Transfer learning along with 2D CNN layer

2. LSTM/GRU in RNN

**3D Convolution Network or Conv3D:**

3D convolutions are a natural extension to the 2D convolutions. Just like in 2D conv, we move the filter in two directions (x and y), in 3D conv, the filter is moved in three directions (x, y and z). In this case, the input to a 3D conv is a video (which is a sequence of 30 RGB images).

**Transfer Learning:**

Let’s use pre-trained model and check if their existing knowledge helps us classify our cases correctly. These pre-trained models which have been already trained on millions of images may prove vital in solving our problem efficiently. For transfer learning we have used the below Architectures,

1. EfficientNetB0

2. InceptionResNetV2

3. MobileNet

**COMPARISON OF DIFFERENT MODELS:**

We have tabulated the different models along with their Accuracy and Loss for comparison and evaluation purpose. The models are being compared and evaluated based on the below metrics provided the model history callback

* Loss incurred during the training phase
* Loss incurred during the validation phase
* Training accuracy
* Validation accuracy

| **Model Name** | **Model Type** | **Batch Size**  **/**  **Epochs** | **Number of parameters** | **Highest Validation accuracy** | **Corresponding Training accuracy** | **Observations** |
| --- | --- | --- | --- | --- | --- | --- |
| Conv2D | Conv2D + RNN + LSTM | 30 / 40 | 26,202,197 | % | % | Model is over-fitting. Augment data using cropping |
| Conv2D | Conv2D + RNN + GRU | 30 / 40 | 20,437,077 | 92% | 84% | Model is over-fitting. Augment data using cropping |
| Conv3D\_model1 | Conv3D | 30 / 20 | 1,105,093 | 95% | 35% | 6 layers + 20 epochs |
| Conv3D\_model2 | Conv3D | 30 / 30 | 1,105,093 | 95% | 82% | Rerunning in 30 epochs |
| Conv3D\_model3 | Conv3D | 30 / 30 | 1,105,093 | 96% | 86% | Adding Kernel regularizing |
| modlelstm | MobileNet + LSTM | 30 / 30 | 3,840,453 | 99% | 96% | Using a Pretrained model as other models did not reach a good validation accuracy |
| modlegru | MobileNet + GRU | 30 / 30 | 3,693,253 | 100% | 92% |  |

This is a sample write-up. The write-up need not be in tabular form.

It doesn’t state that ConvLSTM will give you better results than Conv3D. The explanation should be as detailed as possible so that the logic behind the decision is conveyed. Also, there are a lot of things you can experiment with in the generator function and elsewhere. Please do not forget to specify the exact metric values, here Accuracy which drives your decision.

You can draw inspiration from the concepts taught in the Industry demo in CNNs to experiment with the data and different architectures.

This should contain the detailed procedure followed in choosing the final model. The write up should start with the reason for choosing the base model, then highlight the reasons and metrics taken into consideration to modify and experiment to arrive at the final model.