**Gesture Recognition Assignment**

By: Nidhi Mantri & Kushagra Sengar

**Problem Statement-**

Imagine you are working as a data scientist at a home electronics company which manufactures state of the art **smart televisions**. You 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

**Common Hyperparameters used:**

* Image height and width: 84 x 84
* Batchsize:32
* Epoch: 50
* Initial learning rate: 0.001
* Decay rate: 1e-6
* Momentum:0.7
* Optimizer: SGD

**Observations-**

The below table consists of the experiments done to build a model to predict the gestures from the given data set.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D** | **Training Loss: 0.759**  **Training Acc:0.749**  **Val Loss:0.489**  **Val Acc :0.818** | **Three sets of convolutional blocks (we used Conv3D, Batch Normalization,elu as an activation function and Maxpooling3D ) along with dropout as a regularization technique.** |
| **2** | **VGG16+GRU** | **Training Loss: 0.795**  **Training Acc:0.753**  **Val Loss:0.876**  **Val Acc :0.777** | **We opted for VGG16 architecture to enable transfer learning and GRU units to find patterns amongst sequence of images(video).** |
| **3** | **VGG16+Conv2D+GRU** | **Training Loss: 0.142**  **Training Acc:0.994**  **Val Loss:0.506**  **Val Acc :0.791** | **Addes extra Conv2D block in the above experiment.** |
| **4** | **VGG16+Conv2D+LSTM** | **Training Loss: 0.060**  **Training Acc:0.998**  **Val Loss:0.546**  **Val Acc :0.841** | **Replaced GRU with LSTM** |
| **Final Model** | **Conv3D** | **Training Loss: 0.759**  **Training Acc:0.749**  **Val Loss:0.489**  **Val Acc :0.818** | **We opted experiment 1 as our final model as the difference between accuracies is comparable and validation loss is the lowest amongst all the models.** |

**Conclusion-**

The Model built using Conv3D architecture performed the best with a training accuracy of 75% and validation accuracy of 82%. Further to improve this model, we can fine tuning hyperparameters to improve our model performance. Also, we could try various other architectures like Conv2D+LSTM, ResNet,etc.