Gesture Recognition – Deep learning

# Problem Statement:

We need 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.

| **Gesture** | **Corresponding Action** |
| --- | --- |
| 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. |

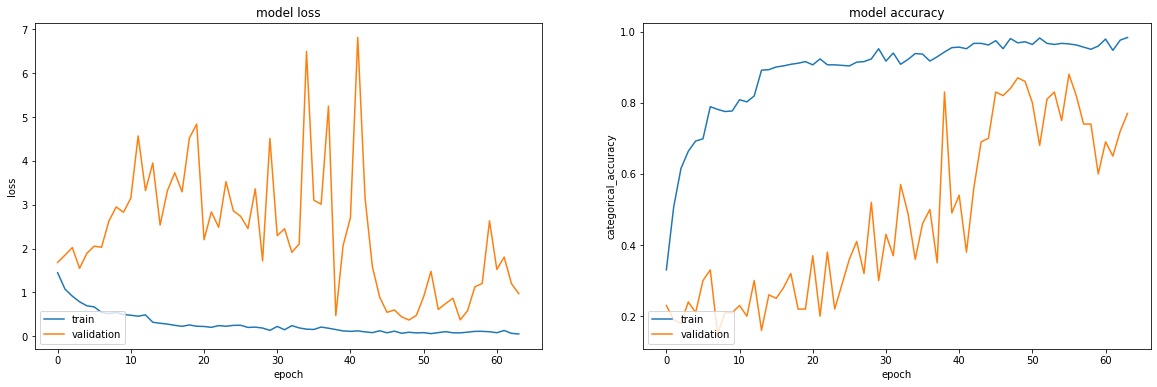
The following table consists of different builds of the model to predict the gestures from the given data set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Build #** | **Model** | **Hyper- parameters** | **Result** | **Explanation** |
| **1** | **Conv3D** | **Batch size = 64,**  **Seq Length = 10,**  **Epochs = 16, Dim = 160x160** | **Train Accuracy: 1.0,**  **Validation Accuracy: 0.70** | **The Model is overfitting and there is a gap between train and validation accuracy. Increasing the layers. Model has to learn more from the data.** |
| **2** | **Conv3D** |  | **Train Accuracy: 0.98,**  **Validation Accuracy: 0.29** | **The model is terribly overfitting and the gap between train and validation accuracy has increased a lot. Adding dropout might help to unlearn certain patterns so that model doesn’t overfit.** |
| **3** | **Conv3D** | **Dropout = 0.2** | **Train Accuracy: 0.92,**  **Validation Accuracy: 0.43** | **There is only a bit of increase in the model validation accuracy. So, changing the layers.**  **Trying to build the model with Time Distributed Conv2D & GRU for improving the performance and reduce the number of Trainable params.** |
| **4** | **Time Distributed Conv2D & GRU** |  | **Train Accuracy: 0.89,**  **Validation Accuracy: 0.26** | **The Model seems to improve the accuracy and decrease in loss with increase in number of epochs. But still training accuracy itself is reduced. Thus, trying out Time Distributed Conv2D & Dense layer instead of GRU.** |
| **5** | **Time Distributed Conv2D & Dense** |  | **Train Accuracy: 0.92,**  **Validation Accuracy: 0.23** | **The model seems overfit a lot and also the trainable params has increased when compared to the previous model. Also increasing the number of epochs to 64 may give a better result.** |
| **6** | **Convo LSTM2D** | **Epochs = 64**  **LR = 0.01** | **Train Accuracy: 0.96,**  **Validation Accuracy: 0.88** | **The validation accuracy is better and the numbers of trainable params are very less when compared to all other models, which is 13,589. The model size is also small with a size of 254KB.** |

# Conclusion:

The Model built with ConvLSTM2D (Build # 6) gave better results compared to all the other models and also the model has very least number of parameters compared to other models.

The Loss tends to 0 and accuracy tends to 1 as shown below.



With the increase in batch size / epochs, model can train and achieve a better accuracy. The gap between training and validation accuracy is not too far, tweaking the Hyper-parameters will end up in a better model.