

## Gesture Recognition

### Problem statement:

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.



Actions	Controls
Thumbs up	Increase the volume
Thumbs down	Decrease the volume
Left swipe	Backward 10 seconds
Right swipe	Forward 10 seconds
Stop	Pause

Below are the details of the experiments conducted on the data....

Experiment No.	Model	Result	Decision + Explanation
1	Conv3D	Train Accuracy: 0.21 Validation Accuracy: 0.20	Batch size = 128 Ablation = 20 Epoch = 20 Model did not learn anything, no loss reduction observed, lets add more layers to the model so that it can learn from data. Total params: 12,904,581 Trainable params: 12,904,581 Non-trainable params: 0

2	Conv3D	Train Accuracy: 0.23 Validation Accuracy: 0.20	Batch size = 35 No improvement in the model, after making some changes to kernel size and adding more layers, will add batch normalization layer after CNN and dense layers Total params: 8,311,813 Trainable params: 8,311,813 Non-trainable params: 0
3	Conv3D	Train Accuracy: 0.90 Validation Accuracy: 0.71	Ablation = None Epoch = 50 Model is over fitting, will add few dropouts in the next experiment so that the model can be generalized Total params: 8,317,701 Trainable params: 8,314,757 Non-trainable params: 2,944
4	Conv3D	Train Accuracy: 0.97 Validation Accuracy: 0.77	Dropout=0.2 There is slight increase in validation accuracy, but still model is over fitting. Let's increase dropout in the next experiment Total params: 22,732,549 Trainable params: 22,730,629 Non-trainable params: 1,920
5	Conv3D	Train Accuracy: 0.95 Validation Accuracy: 0.55	Dropout=0.5 After increase dropout, the validation accuracy is reduced further and

			<p>model is still over fitting, will try a new architecture – GRU in the next experiment</p> <p>Total params: 712,453</p> <p>Trainable params: 710,533</p> <p>Non-trainable params: 1,920</p>
6	Time Distributed + GRU	<p>Train Accuracy: 0.88</p> <p>Validation Accuracy: 0.83</p>	<p>The model is working quite well on validation with less number total trainable parameters.</p> <p>Now, will replace GRU with a plain Dense Layer Network and add global average pooling</p> <p>Total params: 99,845</p> <p>Trainable params: 99,269</p> <p>Non-trainable params: 576</p>
7	Time Distributed + Dense	<p>Train Accuracy: 0.94</p> <p>Validation Accuracy: 0.90</p>	<p>This is good model so far; training and validation accuracy are close. Let's use different architecture of model with time distributed and ConvLSTM2D</p> <p>Total params: 129,477</p> <p>Trainable params: 128,517</p> <p>Non-trainable params: 960</p>
8	Time Distributed + ConvLSTM 2D	<p>Train Accuracy: 0.66</p> <p>Validation Accuracy: 0.55</p>	<p>Trainable parameters are less, but accuracy is not satisfactory</p> <p>Total params: 13,781</p> <p>Trainable params: 13,589</p> <p>Non-trainable params: 192</p>
<b>Final Model</b>	<b>Time Distributed + Dense</b>	<p><b>Train Accuracy: 0.94</b></p> <p><b>Validation Accuracy: 0.90</b></p>	<p><b>This is the best model so far, the train and validation accuracy look good and total parameters are of moderate number.</b></p>