

Gesture Recognition - Deep Learning

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.

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

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

Model #	Details of Model	Params	Size	Epoch	Performance	Observation
Model-1	Conv3D+MaxPooling+Batch Normalisation	65,84,997	25.12 MB	20	categorical_accuracy: 0.8700 - loss: 0.3296 - val_categorical_accuracy : 0.5600 - val_loss: 1.7863 - learning_rate: 3.1250e-04	Loss is least, accuracy is >87%. Gap between train and validation accuracies, hence model might not perform well with unseen data. Also the size of model is large.
Model-2	Conv3D+MaxPooling+ Flatten	3,75,18,353	143.12 MB	40	categorical_accuracy: 0.2267 - loss: 1.6099 - val_categorical_accuracy : 0.2100 - val_loss: 1.6068	Very low accuracy and high loss, size of the model is also more.
Model-3	Conv3D+MaxPooling+Batch Normalisation+Dropouts	7,12,453	2.72 MB	25	categorical_accuracy: 0.7301 - loss: 0.6974 - val_categorical_accuracy : 0.5500 - val_loss: 1.2168	Accuracy is good, loss is less. We will continue to train further to find more tuned version and lower the gap of train and validation accuracies.

Model-4	TimeDistributed Conv2D+Max Pooling+Batch Normalisation + Dropouts +GRU	13,781	53.83 KB	30	categorical_accuracy: 0.8122 - loss: 0.4267 - val_categorical_accuracy: 0.7800 - val_loss: 0.5161	Loss is less & accuracy >80%. Size of model is also small. The difference between train and validation accuracies is <4%. This model is the best model.
Model-5	TimeDistributed Conv2D+Max Pooling+Batch Normalisation +LSTM	13,781	53.83 KB	30	categorical_accuracy: 0.4089 - loss: 1.2513 - val_categorical_accuracy : 0.3200 - val_loss: 2.3300	Loss is more, accuracy <<80%.
Model-6	Model-4 + More Layers + More epochs	13,781	~54 KB	35	categorical_accuracy: 0.7812 - loss: 0.5636 - val_categorical_accuracy : 0.7400 - val_loss: 0.7115	Loss is more than Model-4, accuracy <80%. Hence, considering Model-4 as the best model.

I have selected the **Model-4 (TimeDistributed Conv2D+Max Pooling+Batch Normalisation + Dropouts +GRU)** as the best model for the task because of the reasons listed below:

1. Loss is less & Model Accuracy > 80%
2. Size of model is small(~58KB)
3. Difference between train & validation accuracies <4%.