Gesture Recognition – Deep learning

Problem Statement

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

Experiments with the data and different architectures.

Experiment Number	Model	Hyperparameters	Result	Decision + Explanation
1	Conv3D	Batch size: 128 Epochs: 30 Dim: 120x120 Augmentation = False LR = 0.001	Throws Generator error: The Out of Memory (OOM) error occurs because model is trying to allocate more memory on the GPU than is available. This usually happens when dealing with large tensors, such as in 3D Convolutional Networks or when batch sizes are too large	Reduce Batch size

2	Conv3D	Batch size: 32 Epochs: 50 Dim: 120x120 Augmentation = False LR = 0.001	Train Accuracy: 88.52% Validation Accuracy: 50 %	Model is highly overfitted. Add dropout in Dense layer and add more layers.
3	Conv3D	Batch size: 32 Epochs: 50 Dim: 120x120 Augmentation = False LR = 0.001	Train Accuracy: 55.46% Validation Accuracy: 43.75 %	Model is underfitted. Accuracy is going down. Adding BatchNormalization.
4	Conv3D	Batch size: 32 Epochs: 30 Dim: 120x120 Augmentation = False LR = 0.001	Train Accuracy: 20.17% Validation Accuracy: 18.75 %	No improvement. Adding dropout layers with CCN layers
5	Conv3D	Batch size: 32 Epochs: 30 Dim: 120x120 Augmentation = False LR = 0.001	Train Accuracy: 22.13% Validation Accuracy: 25 %	No improvement. Adding (RNN) LSTM(128) layer after CNN Layers

6	Conv3D with LSTM(128)	Batch size: 32
		Epochs: 30
		Dim: 120x120
		Augmentation = False
		LR = 0.001
		LSTM(128)
			Train Accuracy: 30.56%
			Validation Accuracy: 37.50 %
				Improvement from last model. Enhancing image dataset using Image filters: Edge enhancement, gaussian blur, image detailing, sharpening and brightness. Apply these filters on randomly basis on datasets.

7	TimeDistributed Conv2D with LSTM(128)	Batch size: 32
		Epochs: 30
		Dim: 120x120
		Augmentation = False
		LR = 0.001
		LSTM(128)
			Train Accuracy: 23.56%
			Validation Accuracy: 50.00 %
				Accuracy went down. Adding TimeDistributed layers with Conv2D. Adjust LSTM's trainable parameter to 64. Resize dimention to 64x64

8	Conv3D with LSTM(64)	Batch size: 32
		Epochs: 40
		Dim: 64x64
		Augmentation = False
		LR = 0.001
		LSTM(64)
			Train Accuracy: 25.40%
			Validation Accuracy: 25 %
				Accuracy went down. Adding GRU layer instead of LSTM layer.

9	Conv3D with GRU	Batch size: 32
		Epochs: 40
		Dim: 64x64
		Augmentation = False
		LR = 0.001
		GRU(64)
			Train Accuracy: 31.40%
			Validation Accuracy: 25 %
				No much improvement. Use GlobalAveragePooling2D instead of GRU

10	Transfer Learning(ImageNet with TimeDistributed and GRU)	Batch size: 32 Epochs: 20 Dim: 120x120 Augmentation = False LR = 0.001	Train Accuracy: 98.32% Validation Accuracy: 93.75%	This is the best model so far we got. Trainable parameters are 3,669,317. The model size is also small (44.5 MB).

Conclusion:

The Model build using transfer learning (ImageNet), time distributed and GRU (Experiment No: 11) gives better result and with very less number of parameters than other models.