Problem Statement

Developing a gesture recognition model which can detect hand gestures of users through a webcam on a smart TV and perform operations corresponding to each gesture.

Experiment Results

Experiment #	Model	Model Info/Hyperparams	Results	Inference
1	**Conv3D**	Batch Size: 64 Epochs: 40 4 CNN (conv3d) layers Flatten layer 2 Dense Layers 20% dropout	**Training Acc: 90.95%** **Validation Accuracy: 64%**	Model overfits on training data, flatten layer is followed by huge weight matrix as it flattens the data into 1D
2	**Conv3D**	Batch Size: 64 Epochs: 40 2 CNN (conv3d) layers GlobalAveragePooling layer 2 Dense Layers	**Training Acc: 72.85%** **Validation Accuracy: 58%**	Using globalaveragepooling reduces the total number of paramters, but the model fails to learn the pattern on the data, both training and validation accuracy is relatively low
3	**Time Distributed Conv2D+ConvLSTM2D**	Batch Size: 64 Epochs: 40 3 time-distributed Conv2D layers ConvLSTM2D layer GlobalAveragePooling layer Dense Layers with dropout (10%)	**Error**	Throws OOM error Changed batch size to 40
4	**Time Distributed Conv2D+ConvLSTM2D**	Batch Size: 40 Epochs: 40 3 time-distributed Conv2D layers ConvLSTM2D layer GlobalAveragePooling layer Dense Layers with dropout(10%)	**Training Acc: 89.14%** **Validation Acc: 86%**	Overall, the model performs much better than conv3D models, however, there is still room for improvement
5 [BEST MODEL]	**Time Distributed Conv2D+ConvLSTM2D**	Batch Size: 40 Epochs: 40 2 time-distributed Conv2D layers ConvLSTM2D layer GlobalAveragePooling layer Dense Layers with dropout(5%)	**Training Acc: 93.21%** **Validation Acc: 90%**	This is the best model in the experiment

FINAL MODEL INFERENCE

Time Distributed Conv2D+ConvLSTM2D gives the best results on the data.

The best model gives an overall validation accuracy of 90% and validation loss of 0.3433.

The size of the model is kept very light with a total of 30,821 parameters out of which 30,581 are trainable parameters

The model architecture is as follows:

- 2 time distributed Conv2D layers
    - 16 filters, (3,3) kernel size
    - 32 filters, (3,3) kernel size
    - Batch Normalisation after each layer
- ConvLSTM2D Layer: Similar to an LSTM layer, but combines gating of LSTM with 2D convolutions.
- Batch Normalisation
- Time Distributed Dense layer with 64 neurons
- Batch Normalization
- GlobalAveragePooling2D layer,
- Dense layer with 64 neurons,
- 5% dropout
- Dense layer with 128 neurons,
- Output Layer