Motion Gesture Recognition – Deep Learning

(Kuntamukkala Pavan Kumar, Arjun Amla, Priyanka Kumari)

DS C46 Batch

Problem Statement:

-We need to develop a feature in the smart-TV that can recognise five different gestures (Thumbs Up, Thumbs Down, Swipe Left, Swipe Right, Stop) performed by the user which will help users control the TV without using a remote.
-There are 663 Gesture folders in Training Folder and 100 Gesture Folders in Validation Folder.
-In each folder we have 30 frames/ images.
-During Training of a model, we took only 15 frames (alternative) out of 30 frames.
-The Image size of train dataset contains 160 x 120 and 360 x 360. We resized them to image of size 80 x 80. The reason for opting such size is computational and memory constraints on our side. Batch size, we took it as 32.
-Number of epoch runs we took as 20, 30, 50 depending on the model loss improvement and its improvement in the accuracy metric.
-The experiments done in building a gesture model are tabulated as follows:

Experiment Number	Model Name	Description of Model	Hyper Parameters Tuned/ Changes Made to the Model	Result (For Last Epoch Run)	Comments
Model 1	Conv3D	Input Layer: 8 Units Hidden Layer 1: 64 Units MaxPooling3D Dense Layer: 256 units Output Layer (Dense connection) : 5 Units Activation: Relu	Optimizer: Adam Learning Rate: 0.01 Loss: Categorical Cross Entropy Epochs: 20	Training Categorical Accuracy: 20.39% Validation Categorical Accuracy: 17.19% Total Number of Parameters: 19,676,885	Model is performing poor. We'll try adding more Hidden Layers
Model 2	Conv3D	Input Layer: 32 Units Hidden Layer 1: 64 Units Maxpooling3D Hidden Layer 2 Units: 128 Maxpooling3D	Epochs: 40	Training Categorical Accuracy: 26.49% Validation Categorical Accuracy: 17.97%	Number of Parameters decreased, but no improvement in models performance. Lets add Dropout layer of value 0.2

		Hidden Layer 3 Units: 256 Maxpooling3D Hidden Layer 4 Units: 512 Maxpooling3D Flatten Layer Dense Layer Units: 256 Output Dense Layer Units: 5		Total Number of Parameters: 7,658,245	
Model 3	Conv3D	Input Layer: 32 Units Hidden Layer 1: 64 Units Maxpooling3D Hidden Layer 2 Units: 128 Maxpooling3D Hidden Layer 3 Units: 256 Maxpooling3D Hidden Layer 4 Units: 512 Maxpooling3D Flatten Layer Dense Layer Units: 512 Output Dense Layer Units: 5	Dropout Layer added with value 0.2 Epochs : 40	Training Categorical Accuracy: 29.46% Validation Categorical Accuracy: 14.06% Total Number of Parameters: 13,558,021	Number of Parameters increased, since we increase number of units in dense layer to 512. But there is no improvement in accuracies of the model. Lets add GlobalAveragePooling3D instead of Flatten Layer
Model 4	Conv3D	Input Layer: 32 Units Hidden Layer 1: 64 Units Maxpooling3D Hidden Layer 2 Units: 128 Maxpooling3D Hidden Layer 3 Units: 256 Maxpooling3D Hidden Layer 4 Units: 512 Maxpooling3D GlobalAverage Pooling3D Layer Dense Layer Units: 512	GlobalAveragePooling3D is added	Training Categorical Accuracy: 49.11% Validation Categorical Accuracy: 29.69% Total Number of Parameters: 2,023,685	Number of Parameters reduced due to replacement of Flatten Layer with GlobalAveragePooling3D. Even though there is an improvement in both the accuracies, they are not at all sufficient as per their values.

		Output Dense Layer Units: 5			
Model 5	Time Distributed Conv2D + GRU	Time Distributed Conv2D: 32 Units MaxPooling2D Time Distributed Conv2D: 64 Units MaxPooling2D GlobalAverage Pooling2D Dense Layer: 64 Units GRU Layer: 64 Units Output Dense Layer: 5 Units	Epochs : 50	Training Categorical Accuracy: 91.52% Validation Categorical Accuracy: 53.12% Total Number of Parameters: 49,733	We can see that there is an overfitting problem in the model. Lets add some dropouts
Model 6	Time Distributed Conv2D + GRU	Time Distributed Conv2D: 32 Units MaxPooling2D Time Distributed Conv2D: 64 Units MaxPooling2D GlobalAverage Pooling2D Dense Layer: 64 Units GRU Layer: 64 Units Output Dense Layer: 5 Units	Dropout Layers added with value 0.2 Epochs : 50s	Training Categorical Accuracy: 90.92% Validation Categorical Accuracy: 78.12% Total Number of Parameters: 49,733	The model now seem to be performing decent. The number of parameters are also less when compared to previous models.
Model 7	Time Distributed ConvLSTM2D	Time Distributed Conv2D Input Layer: 8 Units Time Distributed Hidden Layer 1: 16 Units ConvLSTM2D : 8 Units Time Distributed	Epochs: 50	Training Categorical Accuracy: 62.35% Validation Categorical Accuracy: 44.53% Total Number of Parameters:	Even though the number of parameters are less, Models performance is not satisfactory.

		Dense Layer: 64 Units GlobalAverage Pooling2D Dense Layer : 64 Units Output Layer Dense: 5		13,781	
Model 8	Transfer Learning using MobileNet V2 + GRU	Time Distributed MobileNetV2 Layer Time Distributed MaxPooling2D Time Distributed Global Average Pooling2D GRU : 64 Units Dense Layer: 64 Units Output Dense Layer: 5 Units	Epochs: 20	Training Categorical Accuracy: 44.05% Validation Categorical Accuracy: 12.5% Total Number of Parameters: 2,526,021	Models performance is not good by using MobileNetV2 layer. Lets use earlier version MobileNet layer i.e., MobileNet version 1
Model 9	Transfer Learning using MobileNet + GRU	Time Distributed MobileNet Layer Time Distributed MaxPooling2D Time Distributed Global Average Pooling2D GRU : 128 Units Dense Layer: 128 Units Output Dense Layer: 5 Units	Epochs: 20	Training Categorical Accuracy: 84.08% Validation Categorical Accuracy: 58.59% Total Number of Parameters: 3,693,253	MobileNet version 1 is performing better in terms of Training Accuracy compared to MobileNetV2. But its Validation accuracy is not satisfactory. The Model seems to be suffering from over fitting problem. Lets add some dropouts.
Model 10	Transfer Learning using MobileNet + GRU	Time Distributed MobileNet Layer	Drop Out Layers added with value: 0.2 Epochs: 30	Training Categorical Accuracy: 94.20%	There is an improvement in the Training accuracy. But the overfitting problem still persists, which is evident from

		Time Distributed MaxPooling2D Time Distributed Global Average Pooling2D GRU : 128 Units Dense Layer: 128 Units Output Dense Layer: 5 Units		Validation Categorical Accuracy: 61.72% Total Number of Parameters: 3,693,253	significant difference of training and validation accuracies. Lets increase the dropout layer values to 0.5
Model 11	Transfer Learning using MobileNet + GRU	Time Distributed MobileNet Layer Time Distributed MaxPooling2D Time Distributed Global Average Pooling2D GRU : 128 Units Dense Layer: 128 Units Output Dense Layer: 5 Units	Dropout Layers added with value : 0.5 Epochs: 30	Training Categorical Accuracy: 68.30% Validation Categorical Accuracy: 57.03% Total Number of Parameters: 3,693,253	Even though the overfitting problem is solved, there is reduction in Training accuracy. When compared to GRU Model 6, it is not satisfactory.
Final Model	**Time Distributed Conv2D + GRU**	**Time Distributed Conv2D: 32 Units MaxPooling2D Time Distributed Conv2D: 64 Units MaxPooling2D GlobalAveragePooling2D Dense Layer: 64 Units GRU Layer: 64 Units**	**Dropout Layers added with value 0.2 Epochs : 50s**	**Training Categorical Accuracy: 90.92% Validation Categorical Accuracy: 78.12% Total Number of Parameters: 49,733**	**The model now seem to be performing decent. The number of parameters are also less when compared to previous models.**

		Output Dense Layer: 5 Units			

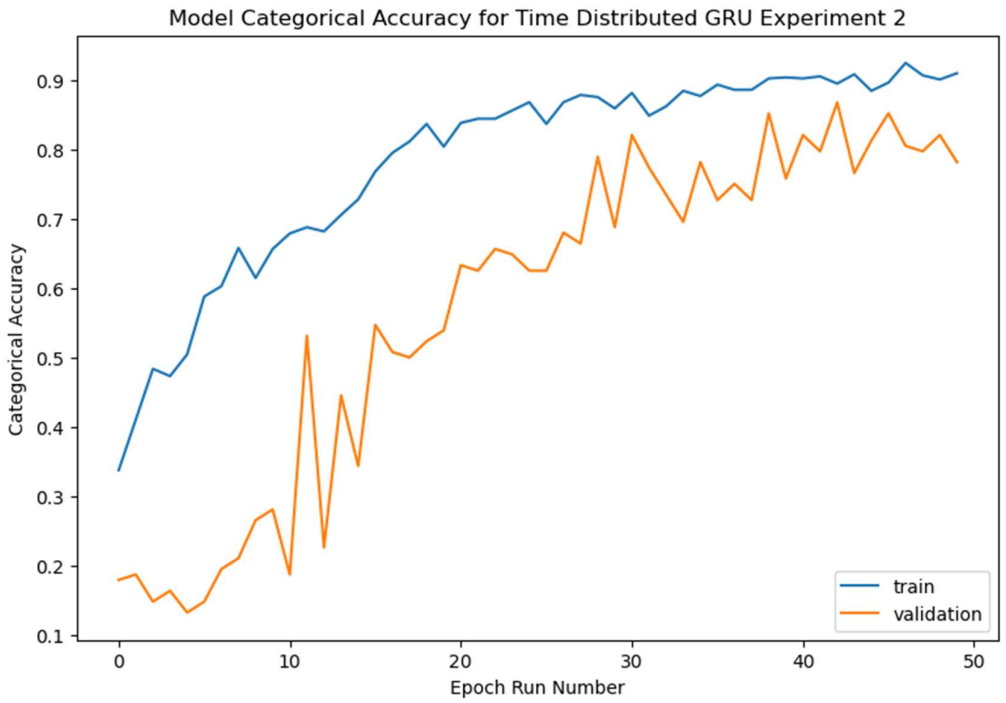
Conclusion:

Model Number 6 (i.e., Time Distributed Conv2D + GRU model with dropout layers = 0.2 attached) , is the best and final model for Motion Gesture Recognition.

Since the performance metrics of the final model's last epoch run (Epoch Run 50) are as follows:

Training Categorical Accuarcy: 90.92%

Validation Categorical Accuracy: 78.12%.



-From the above graph, we can see that for the last 10 epoch runs (epoch run 40-50) the Training and Validation Accuracies are stabilised. We'll pick one of the .h5 file in the stated epoch runs range.

-We are picking Epoch Run 46 "`model-00046-0.27637-0.89583-0.37275-0.85156.h5`" as our .h5 file.
-Since it has Training Accuracy: 89.58%, Validation Accuracy: 85.15%.
-The trade off between these two accuracies are better for the above .h5 file when compare d to other .h5 files.