Neural Network Project

Gesture Recognition

# Problem Statement

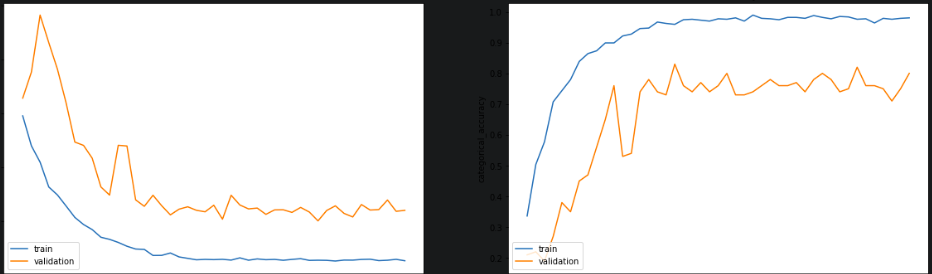
Imagine you are working as a data scientist at a home electronics company which manufactures state of the art **smart televisions**. You want to develop a cool feature in the smart-TV that can **recognize five different gestures** performed by the user which will help users control the TV without using a remote.

|  |  |
| --- | --- |
| **Gesture** | **Action** |
| **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 |

# Model Results

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment Number | Model | Result | Decision + Explanation |
| 1 | CNN2D + GRU  Adam Optimizer  LR: 0.0001  Batch\_size: 10  Epochs: 20 | Training Accuracy: 99.85 %  Validation Accuracy: 73.00% Training Loss: 0.067 | Time Distributed CNN2D + GRU model has been used. The model seems to be overfitting. Training the model with more epochs should probably improve the model to more extent. |
| 2 | CNN2D + LSTM  Adam Optimizer  LR: 0.0001  Batch\_size: 10  Epochs: 20 | Training Accuracy: 98.79%  Validation Accuracy: 73.00% Training Loss: 0.09 | Time Distributed CNN2D + LSTM model has been used. The model seem to be overfitting, it has good train accuracy and bad validation accuracy with least learning loss. Training the model with more epochs should probably improve the model to more extent. |
| 3 | CNN3D  Adam Optimizer  LR: 0.0001  Batch\_size: 10  Epochs: 20 | Training Accuracy: 95.32%  Validation Accuracy: 71.00% Training Loss: 0.1909% | CNN3D model has been used. The model does seem to be overfitting, it has high difference of train accuracy and validation accuracy with a bit high learning loss. Adding of Regularizers and Removal of a layer should probably improve the model to some extent. |
| 4 | CNN3D + l2 Regularizer  Adam Optimizer  LR: 0.0001  Batch\_size: 10  Epochs: 20 | Training Accuracy: 92.76%  Validation Accuracy: 69.00% Training Loss: 4.6951 | CNN3D model has been used with l2 Regularizer to reduce the model overfitting. The model still does seem to be overfitting, it has good train accuracy and bad validation accuracy compared to CNN3D Model. But the training loss seems to have been increased rapidly to a greater extent of 4%. |
| 5 | CNN2D + LSTM (45 Epochs)  SGD Optimizer  LR: 0.01  Batch\_size: 10  Epochs: 45 | Training Accuracy: 98.04%  Validation Accuracy: 80% Training Loss: 0.1356 | Time Distributed CNN2D + LSTM model has been used. Replacing the optimizer from Adam to SGD with the default learning rate of 0.01. The model does seem to have bit higher training loss than rest of the models, but the difference in training accuracy and validation accuracy is comparatively very less. |
| 6 | CNN2D + LSTM (45 Epochs)  Adam Optimizer  LR: 0.0001  Batch\_size: 10  Epochs: 45 | Training Accuracy: 99.40%  Validation Accuracy: 82% Training Loss: 0.0869 | Time Distributed CNN2D + LSTM model has been used. The model seems to have no major improvement than the first model with 20 epochs. |
| 7 | CNN2D + GRU (45 Epochs)  Adam Optimizer  LR: 0.0001  Batch\_size: 10  Epochs: 45 | Training Accuracy: 100%  Validation Accuracy: 76% Training Loss: 0.003 | Time Distributed CNN2D + GRU model has been used with increased epochs to see if we would have any increase in validation accuracy, but there has been no as such. The model does overfit, it has good train accuracy and validation accuracy with improved training loss than the first CNN2D + GRU Model. But there is not much significant improvement in the model even after training it for more epochs. |
| Final Model | CNN2D + LSTM (45 Epochs)  SGD Optimizer  LR: 0.01  Batch\_size: 10  Epochs: 45 | Training Accuracy: 98.04%  Validation Accuracy: 80% Training Loss: 0.1356 | CNN2D + LSTM with SGD Optimizer gives better accuracy results than compared to other models. This can be further improved by increasing epochs and tuning its hyperparameters. |

# Final Model Graph



# Further Suggestions for Improvements

The model can be further improved by using **Pre-Trained Models**, **Tuning the Hyperparameters** and **Exploring more ways to feed more information to models from the Data**. These methods can improve the accuracy of the model and the validation accuracy with less training loss as compared to the above models.