# **Gesture Recognition Assignment**

## **Problem Statement:**

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 TV without using the remote.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D** | **Train Accuracy = 0.98**  **Validation Accuracy = 0.53** | **It’s a base model and overfitting. Increasing Batch Size to 64.** |
| **2** | **Conv3D** | **Train Accuracy = 0.62**  **Validation Accuracy = 0.49**  **(At epoch 9, both the training and validation overlap at accuracy of 0.74)** | **Even though the overfitting reduced but accuracy took a drastic toll. Also at epoch 9 there was good accuracy but the loss was high. Adding some more layers to the model.** |
| **3** | **Conv3D** | **Train Accuracy = 0.21**  **Validation Accuracy = 0.16** | **Adding layers didn’t increase accuracy rather brought it down. Reducing batch size to 32 and increasing epochs to 20** |
| **4** | **Conv3D** | **Train Accuracy = 0.34**  **Validation Accuracy = 0.23** | **Accuracy increased slightly so is overfitting. Decreasing learning rate to 0.0002 and increasing batch size to 64** |
| **5** | **Conv3D** | **Train Accuracy = 0.36**  **Validation Accuracy = 0.28** | **Accuracy again increased slightly. Adding batch normalization layers and Reducing batch size to 16.** |
| **6** | **Conv3D** | **Train Accuracy = 0.93**  **Validation Accuracy = 0.46** | **Accuracy increased but overfitting so adding dropout layers with value 0.2.** |
| **7** | **Conv3D** | **Train Accuracy = 0.89**  **Validation Accuracy = 0.54** | **Slightly reduced the overfitting so tried increasing the dropout value to 0.5.** |
| **8** | **Conv3D** | **Train Accuracy = 0.94**  **Validation Accuracy = 0.63** | **Overfitting is still persisting but accuracy is good so played around with increasing batch size to 25 and epoch to 30.** |
| **9** | **Conv3D** | **Train Accuracy = 0.93**  **Validation Accuracy = 0.61** | **No major change as compared to previous one so adding Global Average Layer in place of Flatten layer and increasing epoch to 40.** |
| **10** | **Conv3D** | **Train Accuracy = 0.93**  **Validation Accuracy = 0.88**  **(This accuracy achieved at epoch 33)** | **This is a by far the most stable and robust model. Both the training and validation accuracy are considerably closer and the validation loss is very less. Lets use GRU layers to see if they compete with this.** |
| **11** | **Time Distributed+ Conv2D + GRU** | **Train Accuracy = 0.95**  **Validation Accuracy = 0.56** | **Tried a slightly different architecture and model overfitted. Adding Dropout layer with value 0.5 and keeping batch size = 25 and epoch = 35.** |
| **12** | **Time Distributed+ GRU** | **Train Accuracy = 0.84**  **Validation Accuracy = 0.61** | **Model still overfitting. Adding Global Average Pooling layer and dense layer instead of GRU and keeping batch size of 25 and epoch 40** |
| **13** | **Time Distributed+ Dense** | **Train Accuracy = 0.93**  **Validation Accuracy = 0.82** | **The model performed very well and trying ConvLSTM2D now.** |
| **14** | **Time Distributed + Conv2DLSTM** | **Train Accuracy = 0.89**  **Validation Accuracy = 0.85**  **(At epoch 31, this accuracy was obtained.)** | **The model is perfect with the minimal validation loss and model is robust as accuracies look fine and are considerably close and is chosen as final model.** |
| **Final Model** | **Time Distributed + Conv2DLSTM** | **Train Accuracy = 0.89**  **Validation Accuracy = 0.85** | **The model is robust at batch size of 25, epoch of 31 with Global Average Pooling Layer and Dense Layer.** |

# **Conclusion:**

The model obtained at 31st epoch of Experiment 14 is found to be the most stable and robust. Also the model is very lightweight of just 254 kB.