**Gesture Recognition Assignment**

Problem Statement:-

As a data scientist at a home electronics company that manufactures state-of-the-art smart televisions. We 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.

* 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.

Observations:-

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment Number | Model | Result | Decision + Explanation |
| 1 | Conv3D | Epochs: 20  Train accuracy: 0.2  Validation accuracy: 0.41 | Adding more layers, as it’s clear that the model cannot learn. |
| 2 | Conv3D | Epochs: 20  Train accuracy: 0.61  Validation accuracy: 0.75 | While the model the accuracies have increased w.r.t the prior model, the number of parameters have gone up significantly. Reducing the number of parameters. |
| 3 | Conv3D | Epochs: 20  Train accuracy: 0.97  Validation accuracy: 0.83 | The model seems to be overfitting, reducing the number of parameters even further. |
| 4 | Conv3D | Epochs: 20  Train accuracy: 0.17  Validation accuracy: 0.5 | Reducing the number of parameters has had an adverse effect and now the model is underfitting, Increasing the number of epochs from 20 to 40 and removing the Dropout layers from the model. |
| 5 | Conv3D | Epochs: 40  Train accuracy: 1  Validation accuracy: 0.91 | The model is clearly overfitting, Reducing the number of parameters and re-introducing Dropout layers. |
| 6 | Conv3D | Epochs: 40  Train accuracy: 0.99  Validation accuracy: 1 | The model is definitely overfitting. Increasing the number of Dropout layers and replacing Flatten layer with GlobalAveragePooling3D |
| 7 | Conv3D | Epochs: 40  Train accuracy: 0.99  Validation accuracy: 0.91 | The model still seems to be overfitting, trying a different architecture |
| 8 | TimeDistributed Conv2D + LSTM | Epochs: 20  Train accuracy: 0.5  Validation accuracy: 0.66 | The accuracy is sub-par, most likely due to the limited number of epochs, however due to time restraints switching to a different architecture, increasing the number of epochs from 20 to 40 |
| 9 | TimeDistributed Conv2D + GRU | Epochs: 40  Train accuracy: 0.96  Validation accuracy: 0.81 | The model works well, however the difference between the train and validation accuracy is significant, adding some Dropout layers |
| 10 | TimeDistributed Conv2D + GRU | Epochs: 40  Train accuracy: 0.86  Validation accuracy: 0.58 | Adding Dropout layers has negatively impacted the model. Trying a different model architecture, increasing the number of epochs to 50 |
| 11 | TimeDistributed Conv2D | Epochs: 50  Train accuracy: 0.99  Validation accuracy: 0.91 | The model seems to be overfitting, introducing Dropout layers to deal with overfitting |
| 12 | TimeDistributed Conv2D | Epochs: 50  Train accuracy: 0.93  Validation accuracy: 0.91 | The accuracy scores for the model are great. Selecting model 12 as the final model |

Conclusion:-

We selected Model 12 (TimeDistributed Conv2D) as our final model as it achieved a training accuracy of 0.93 and validation accuracy of 0.91, which suited our requirements.