**Gesture Recognition - Neural Network**

**Problem Statement:**

We need 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. The following table consists of the experiments done to build a model to predict the gestures from the given data set.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1 & 2** | **Conv3D** | **Model 1 threw out of memory error for batch size 128. So, we decided to stick with batch size 32 and 64.**  **Model 2 Throws Generator error** | **Batch size to be used as 32. We will also try with 64 for couple of models.**  **Image resolution would be used at 80\*80, we would also try 120\*120 for couple of models to check if it improves accuracy.** |
| **3-> First model in notebook, Built on top of 1 & 2** | **Conv3D** | **No of Params =** 1,733,509  **Epochs = 20**  **Training Accuracy = 0.86**  **Validation Accuracy = 0.45** | **The Validation accuracy weas concluded good but it was varying wildly through different epochs.**  **We would test if we could get same results if we change the batch size to 64. We will also increase the epochs to see if the validation accuracy stabilizes.** |
| **4 -> Model 2** | **Conv3D** | **No of Params =** 1,733,509  **Epochs = 30**  **Training Accuracy = 0.99**  **Validation Accuracy = 0.84** | **We find the validation accuracy is improved and is stable with the increased number of epochs.**  **Now we can experiment if we can get better results by improving the image resolution. Previously we tried with the 80\*80 image resolution and only 13 images from the set we considered. Now we will use 120\*120 image resolution and we will use 20 images from each video frames.** |
| **5 -> Model 3** | **Conv3D** | **No of Params =** 7,467,909  **Epochs = 20**  **Training Accuracy = 0.99**  **Validation Accuracy = 0.89** | **In this model we can see the model is overfitting as the validation accuracy is dropped drastically.**  **We will now experiment if increasing the no of epochs help to increase the validation accuracy as we saw in the first two models. Or we will need to go for augmenting to create more training data to avoid overfitting.** |
| **6 -> Model 4** | **Conv3D** | **No of Params =** 7,467,909  **Epochs = 40**  **Training Accuracy = 0.83**  **Validation Accuracy = 0.75** | **Here we see the validation accuracy is increased compared to the previous model, but it is not able to increase above the threshold of 0.80 where the training accuracy have reached 0.95. This concludes no of epochs increased do not help much in improving the model which is overfitting.**  **Now we will try to implement the CNN + LSTM model and will compare the results with the second model which had highest validation accuracy.** |
| **7 -> Model 5** | **CNN+LSTM** | **No of Params =** 6,714,117  **Epochs = 20**  **Training Accuracy = 0.98**  **Validation Accuracy = 0.81** | **This model is having overall less training and validation accuracy, but we see steady trend in increasing the training and validation accuracy. Adding more epochs can help here to further improve the accuracy and r help in overfitting.** |
| **8 -> Model 6 -> Final Model** | **CNN+LSTM** | **No of Params =** 6,714,117  **Epochs = 40**  **Training Accuracy = 0.77**  **Validation Accuracy = 0.74** | **On further training this model did reach good training as well as validation accuracy and seems inline with no overfitting.** |