|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D(52M trainable parameters)** | **Threw OOM error in my local system of gpu 4gb** | **I was trying to fit a model with 52 million trainable parameters which is huge so tried using nimble box** |
| **2** | **Conv3D(52M trainable parameters)** | **Accuracy: 0.24**  **(batch size 10)** | **Tried the same model with 52 million trainable parameters in nimble box ,there also I encountered lot of issues so I reduced the batch size 10 so it started working but accuracy was bit less it was not moving more than 24% then I tried running the same model with google colab** |
| **3** | **Conv3D(52M trainable parameters)** | **Accuracy: 0.98**  **(batch size 50)** | **I tried running the same model with google colab since it has more gpu was able to execute with batch size 50** |
| **4** | **Conv3D** | **Accuracy: 0.32** | **In order to decrease the number of parameters following things are done:**   1. **Decreased the number of filters in each CON3d layer**   filters = [8,16,32,64]  **for corresponding CON3D layer**   1. **This would have impacted accuracy so we increased the CON3D layers with maxpooling so that dimension reduce** 2. **At the end added dense layer with** [1000, 500, 5] **neurons respectively.** 3. Added drop outs |
| **5** | **Conv3D** | **Accuracy: 0.53**  **(with trainable parameters=**  2,835,749 **651,109)**  **Batch size=50** | **For the same model above tried reducing the dense layer with 256 and 128 neurons respectively and applied batch normalization in between layers, after doing this I could see low training accuracy :50% but high validation accuracy 98% so thought removing dropout (reguliazer) would fix this** |
| **6** | **Conv3D** | **Accuracy: 0.98**  **(with trainable parameters=**  **651,109)** | **I removed the dropouts from above model and finally got the best Model** |
| **7** | **ConvGRU** | **Accuracy:0.98**  **But unstable got very less accuracy for few epochs** | **Added one time distributed CON2D layer with 64 filters of size (3,3) and maxpooling the output is flattened and feed to GRU with 10 gates and finally dense layer with softmax. Even though the model gave good accuracy for few epochs its very unstable** |
| **8** | **ConvGRU** | **Accuracy<0.40** | **Tried few other ConvGRU like add few other layers of conv2d networks ,increasing dense layers, etc, but accuracy didn’t cross 0.4**  **So thought of trying transferable learning Conv2d with GRU** |
| **9** | **transferable learning Conv2d with GRU** | **Accuracy <0.1** | **Tried by freezing and defreezing the parameters of trained models: vggnet16,resnet50,vggnet19, InceptionV3,etc**  **And also tried adding dense layers at the end** |
| **Final Model** | **Conv3D** | **Accuracy: 0.98**  **(with trainable parameters=**  **651,109)** | **It has 4 layers of CON3D with** filters = [8,16,32,64]  **respective for each layer, maxpooling,batch normalization in between the layers of con3d layers,dense layers and softmax at the end.**  **This is the best model since it has less number of trainable parameters and giving high accuracy** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |