**Conv3D:**  
We started our experimentation with following config: batch\_size = 64, image\_height=160, image\_width=160, channels=3, num\_classes=5, total\_frames=30, frame\_samples=30, num\_epochs = 20,  
Activation= relu.

The architecture had 4 conv3D layers (each followed by max\_pooling layer) using 3\*3 filters with batch normalization and two dense layers and final SoftMax layer as output layer. We chose this setup to ensure that the model is not too simplistic and too complicated either. Idea was to modify the architecture based on how the model underfit/overfit(s) the data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exp No** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D** | **System threw error as resource got exhausted.** | **Crop and resize the images optimally, try to fit on data with above mentioned initial set of configs** |
| **2** | **Conv3D** | **Overfit: train=90%, val=20%(Acc)** | **Reduce the size of the batch from 64 to 32** |
| **3** | **Conv3D** | **Overfit: train=80%, val=50%(Acc)**  **Compute speed improved** | **crop images to remove unwanted excess pixels and rescale them(120\*120)** |
| **4** |  | **Overfit: train=90%, val=50%(Acc)**  **Compute speed improved further** | **Reduce frame\_samples to correct overfit prob** |
| **5** |  | **Overfit: train=88%, val=46%(Acc)** | **Further reduced frame\_samples as accuracy did not improve.** |
| **6** | **Conv3D** | **No significant improvement in Acc. Overfit problem.**  **Training speed increased significantly.** | **Experimented with different frame\_samples(16 to 30). Finally stuck to 16 frames as training accuracy was not affected much and remained almost the same.** |
| **7** | **Conv3D** | **Overfit: train=94%, val=80%(Acc)** | **Removed one conv layer(4->3) to simplify the model. Used dropout at conv and dense layers so as to address overfit problem.**  **Increased no of epochs to 50** |
| **8** | **Conv3D** | **Train\_acc improved to 99% not much improvement in val\_acc**  **Overfit: train=99%, val=77%(Acc)** | **Increased frame\_samples to 30 to introduce more data so as to avoid overfitting.** |
| **9** | **Conv3D** | **Accuracy was improved, but model still overfit**  **train=99%, val=86%(Acc)** | **Removed one more dense layer(2->1) to simplify the model.**  **Used dropout only at dense layer to address overfit.** |
| **10** | **Conv3D** | **Overfit: train=98%, val=89%(Acc)** | **As train accuracy was close to 99.8, reduced frame\_samples back to 16 to avoid the model from byhearting entire dataset.**  **Increased dropout ratio at dense layer.** |
| **11** | **Conv3D** | **Overfit: train=93%, val=81%(Acc)** | **Added dropout to one of the conv layer** |
|  | **Conv3D** | **Understood that the problem was with learning rate.** | **After several attempts found that after certain no of epochs, model would stagnate i,e Val\_loss would either stop or start increasing, model would overfit drastically.**  **Model would not converge** |
| **12** | **Conv3D** | **The model started to converge slowly**  **train=89%, val=82%(Acc)** | **Further reduced learning rate to lr=0.00009 and set factor=0.01, patience=5.**  **Learning was slow as lr was very small** |
| **13** | **Conv3D** | **The accuracies were oscillating as below:**  **train=86%, val=76%(Acc)**  **train=89%, val=76%(Acc)** | **Tried changing the lr so as to let the model converge slowly and achieve better accuracies.** |
| **. 14** | **Conv3D** | **The best accuracies achieved were**  **train=91%, val=87%(Acc)**  **train=90.12%, val=90%(Acc)** | **On further increasing the num\_epochs close to 60, With lr=0.00009, factor=0.01, patience=5.**  **The model performed better.** |
| **15** | **Conv3D** | **train=98%, val=93%(Acc)**  **train=98%, val=94%(Acc)** | **Final config: with lr=0.009** **factor=0.01, patience=5 and num\_epochs = 62. Model achieved the best accuracy.** |