|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D** | **Performing model training with different batch size to user largest possible batch size for more generalized model. Throws Resource exhausted error when batch size=128** | **Selecting the batch size=64** |
| **2** | **Conv3D** | **Val acc around 57%, training accuracy is 1.** | **Clear indication of overfitting. Introducing 25% drop out in last conv layer, reduced the dense layer filters from 128 to 64** |
| **3** | **Conv3d** | **Validation and training accuracy stands at 42% and 95% respectively** | **Model is still suffering from overfitting. Reducing the no of frames from 30 to 20 to reduce overfit.** |
| **4** | **Conv3d** | **Val accuracy 41% and training accuracy is 98%** | **Overfitting is still prevailing. It is evident that CNN (Conv3d) models are unable to get high accuracy while classifying validation data. Hence, we will check the same with Con2d+RNN architecture. Also, switching back the no of frames to 30** |
| **5** | **Conv2d+RNN** | **with batch size=64, we got GPU memory warning during training. Also, val accuracy is very less (29.68%) where training accuracy is 87.5%.** | **Updating the batch size to 32 to resolve memory issue.** |
| **6** | **Conv2d+RNN (with batch size 32)** | **62.5 % of val accuracy achieved, training accuracy is 89.28%** | **Validation accuracy increased than before. Change the optimizer from SGD to RMSProp.** |
| **7** | **Conv2d+RNN (with batch size 32, optimizer = RMSProp)** | **val accuracy increased to 70% and 96% training accuracy achieved, overfitting also reduced to some extent.** | **Change the optimizer to Adam** |
| **8** | **Conv2d+RNN (with batch size 32, optimizer Adam)** | **Marginal improvement of val accuracy (71.8%) achieved, training accuracy stand at 94%. Overfitting is slightly reduced.** | **Change the architecture to**  **ConvLSTM2D layers if it can give higher validation accuracy. Change the optimizer back to SGD.** |
| **9** | **ConvLSTM2D**  **with batch size 32, optimizer=SGD** | **Very poor performance observed, both val and training accuracy stands at 20%. This is possibly due very less number of conv layers (2) and no of fitters (8) in both conv and dense layer.** | **Unable to use a complex structure for model training like other architectures, due to memory issue as ConvLSTM2D layers are computationally expensive as both input and recurrent transformations are convolutional. Hence, changing the architecture to Conv2d and LSTM, stacked individually.** |
| **10** | **Conv2d+LSTM, batch size=32, Optimizer=SGD**  **LSTM layer has 64 filters** | **63% val accuracy achieved and training accuracy is 84%** | **Overfitting slight reduced, but val accuracy reduced that Conv2d+RNN architecture, Change the no of LSTM filters to 128** |
| **11** | **Conv2d+LSTM, batch size=32, Optimizer=SGD**  **LSTM layer has 128 filters** | **68.75% val accuracy achieved, training accuracy is 91.9%.** | **To increase accuracy, adding another conv2d layer (4th) with 256 filters** |
| **12** | **Conv2d+LSTM, batch size=32, Optimizer=SGD**  **LSTM layer has 128 filters, additional conv2d (4th) layer with 256 filters** | **67%val accuracy, 91.8% training accuracy** | **As no improvement from previous model is observed removing the conv2d (4th) layer with 256 filters.**  **Using optimizer RMSprop** |
| **13** | **Conv2d+LSTM, batch size=32, Optimizer=RMSprop**  **LSTM layer has 128 filters,** | **68.75% val accuracy and training accuracy is 98.5%** | **Marginal improvement from previous model. Using optimizer Adam.** |
| **14** | **Conv2d+LSTM, batch size=32, Optimizer=Adam**  **LSTM layer has 128 filters** | **78% val accuracy,**  **98% training accuracy** | **Higher val accuracy. Overfitting is reduced to some extent.**  **However, after observing the validation accuracy vs epoch graphs of all CONV2d + LSTM models built so far, it seems that the model performance may increase if no of epoch is higher. Currently for all fits, no of epoch is 30. Changing it to 50.** |
| **15** | **Conv2d+LSTM, batch size=32, Optimizer=Adam**  **LSTM layer has 128 filters, no of epoch = 50** | **79.6% val accuracy,**  **99% training accuracy** | **Conv2d + LSTM models are performing better consistently when epoch numbers are high (35+). The no of epoch vs validation loss indicates that with higher number of epochs, the val loss may reduce further.**  **Change the architecture to conv2d+GRU, changing the epoch to 40, as GRU converges faster.** |
| **16** | **Conv2d+GRU, batch size=32, Optimizer=Adam**  **LSTM layer has 128 filters, no of epoch = 40** | **89% val accuracy while training accuracy is 99%** | **After using GRU instead of LSTM, overfitting has been greatly reduced. Also, the loss and accuracy graphs are indicating that GRU is converging faster than LSTM (as it requires less no of epochs to reduce val loss)** |

Based on the above observation, we are considering Conv2d+GRU to be the best model resulting 89% validation accuracy. It has also been observed that optimisers RMSProp and Adam is doing better than SGD in terms of valuation accuracy while Adam is giving the slightly better result than RMSProp. The .h5 model file which results in highest validation accuracy is "model-00038-0.01989-0.99256-0.41072-0.89062.h5"