The write-up of our experiment using generators with & without augmented data and our analysis.

Experiment Number	Model	Result	Decision + Explanation
1	**Conv3D**	Total Params: **687813** Training Accuracy: **0.5309** Validation Accuracy: **0.1700**	- As we see from the experiments "image resolution" and number of frames in sequence have more impact on training time than batch_size. - can consider the Batch Size around 15-40 - resolution (120,120) of images
2	**Conv3D**	Total Params: **1736389** Training Accuracy: **0.6035** Validation Accuracy: **0.1500**	- As we see from the experiments "image resolution" and number of frames in sequence have more impact on training time than batch_size. - can consider the Batch Size around 15-40 - resolution (160, 160) of images
3	**Conv3D**	Total Params: **1,117,06** Training Accuracy: **0.9414** Validation Accuracy: **0.2300**	- Model is clearly overfitting. - Adding a dropout layer in the next model will help to overcome overfitting.
4	**Conv3D**	Total Params: **3,638,981** Training Accuracy: **0.7426** Validation Accuracy: **0.2300**	- We can see val_loss did not improve from 1.73641 so early stopping stops the epoch automatically!! - Last Epoch stopped on 11/25!! - Next we can try to reduce the filter size and image resolution and see if we get better results. Moreover since we see minor oscillations in loss.
5	**Conv3D**	Total Params: **2,556,533** Training Accuracy: **0.7637** Validation Accuracy: **0.2100**	- Model is clearly overfitting. - Adding dropouts has further reduced validation accuracy as the model doesn't seem to generalize well. - All the experimental models above have more than 1 million parameters. - We can try to reduce the model size to create & train a low memory footprint model and see the better performance.
6	**CNN-LSTM**	Total Params: **1,657,445** Training Accuracy: **0.9604** Validation Accuracy: **0.8300**	- For CNN LSTM model we get a best validation accuracy of 83% - Also the number of parameters is less compared to other models. - LSTM tackles gradient vanishing by ignoring useless data/information in the network.

7	**Conv3D with Data Augmentation**	Total Params: **2,556,533** Training Accuracy: **0.6487** Validation Accuracy: **0.2200**	- Model is clearly overfitting. - Adding more layers, reduced network parameters can help us to tackle overfitting to some extent.
8	**CNN-LSTM GRU with Data Augmentation**	Total Params: **2,573,925** Training Accuracy: **0.9165** Validation Accuracy: **0.6800**	- Model performance is improved & considerable. - However, the number of parameters is quite high compared to CNN-LSTM. - We see that overfitting is considerably high when we do more augmentation. However there is not much improvement on accuracy
9	**Transfer Learning with GRU**	Total Params: **3,693,253** Training Accuracy: **0.9964** Validation Accuracy: **0.9500**	- Model performance is Awesome. - However, the number of parameters is quite high compared to CNN-LSTM. - We see that overfitting is considerably high when we do more augmentation. However there is not much improvement on accuracy

Conclusion:

Based on our Experiment using generator with and without augmented data and our analysis:

We can see LSTM networks combat the RNN's vanishing gradients or long-term dependence issue. Gradient vanishing refers to the loss of information in a neural network as connections recur over a longer period. In simple words, LSTM tackles gradient vanishing by ignoring useless data/information in the network.

Here is the details result of CNN-LSTM Model:

- Total Params: **1,657,445**
- Training Accuracy: **0.9604**
- Validation Accuracy: **0.8300**

Hence, we considered the CNN-LSTM model is the best model, with below given weight for model testing.

The best weights of CNN-LSTM: model-00020-0.14990-0.96003-0.64083-0.83000.h5 (20 MB).