First, a random model is built and then it is improved by tweaking with the parameters and hyperparameters. We have a sequence of images (30) which represent a video. Though we have only considered only around half the images as it helps to reduce memory consumption. Below table represents every model built on the dataset with their results also what we infer from it as well as decision taken.

Experiment Number	Model	Result	Decision + Explanation
1	**Conv3D**	**Training acc: 97.74%** **Val acc: 86%**	Slight overfit. (Initial model) it is because dropouts are not used. Checking with batch size = 128 in the next model
2	**Conv3D**	**Training acc: 65%** **Val acc: 25%**	Heavy Overfitting obtained. Batch size = 128 is not optimal for validation set. Using Batch Normalization and Dropouts to reduce overfitting and also reducing batch size
3	**Conv3D**	**Training acc: 99.55%** **Val acc: 85%**	Reduce batch size to 64 Reduce no of neurons in last hidden layer to reduce parameters. Overfitting is reduced. Further reducing batch size to see if there is any improvement in val accuracy
4	**Conv3D**	**Training acc: 99.4%** **Val acc: 92%**	Reduce batch size to 32 Rest same as model 3. Batch size = 32 is optimal for both train and validation data. We got very good validation accuracy and least overfitting
5	**Conv3D**	**Training acc: 99.7%** **Val acc: 88%**	Adding one more hidden layer at last to model 4 to increase accuracy. Using dropouts to tackle overfitting. But val accuracy drops
6	**Conv2D**	**Training acc: 64.86 %** **Val acc: 54%**	Converting images to grayscale format as parameters will be reduced and model size shrinks. This reduces large amount of information which makes it difficult to train though it has less memory consumption
7	**Conv2D**	**Training acc: 83.11 %** **Val acc: 65%**	Using Dropouts to reduce overfitting. Increase last hidden layer neurons to increase accuracy. But still overfit exists this might be due to loss of

			information in input. Next, we try CNN-RNN stack model
8	**ResNet50 + GRU**	**Training acc: 52.64 %** **Val acc: 56%**	Model is not able to learn on training dataset itself. It is because of less no of parameters and depth. Increase last hidden layer neurons to increase accuracy in the next model
9	**ResNet50 + GRU**	**Training acc: 47.66 %** **Val acc: 43%**	The model doesn't fit well on our dataset after increasing complexity. So, we discard this model as ResNet50 pre-trained model doesn't seem to suit our dataset. In next model we will try ResNet50V2 pre-trained model and see if there is any improvement.
10	**ResNet50V2 + GRU**	**Training acc: 100 %** **Val acc: 87%**	Increased depth of ResNet50V2 helped achieve higher accuracy than previous CNN-RNN models. But there is slight overfitting it may be because ResNet50V2 is too complex for our dataset and it is difficult to generalize the data for it
11	**ResNet50V2 + LSTM**	**Training acc: 95.48 %** **Val acc: 84%**	Trying LSTM instead of GRU to check if accuracy increases. LSTM has surely helped reducing overfitting but validation accuracy is not that great like model 4.
Final Model	**Model 4**	**Training acc: 99.4%** **Val acc: 92%**	This model has the least overfit with less parameters relatively (less than 1 million)