Following are a list of models/experiments tried out:

* As there are different image sizes, we set the best possible size of images to be 120x120 (as the smaller images’ smallest dimension is 120) reducing the loss due to shrinkage of the image. Also, the generator normalizes the images at each of the 3 channels (for RGB cases)
* We have reduced the frame rate of the videos from 30 fps to 20 fps for faster processing. These 20 frame images are taken at regular intervals using the np.linspace function. The batch size is experimented to 10-30 across various experiments.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exp.No#** | **Model** | **Model Details** | **Output** | **Result** |
| 1 | Model\_01 | Classic Convolutional 3D model with 3 (Conv3D layer+Activation+Pooling) + 1 Dense Layer on an image size 120x120. epochs=20; batch\_size=20. Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 38,56,069  Trainable Parameters: 38,55,941 | Accuracy: 96.98  Validation Accuracy: 72.15  Validation loss: 0.7215 | The model is slightly overfitting, and the validation loss can also be reduced. |
| 2 | Model\_02 | Classic Convolutional 3D model with 5 (Conv3D layer+Activation+Pooling) + 2 Dense Layers on an image size 120x120. epochs=20; batch\_size=30. Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 9,81,189  Trainable Parameters: 9,80,933 | Accuracy: 92.31  Validation Accuracy: 86.00  Validation loss: 0.4759 | Improvement in the Validation Loss. And the overfitting problem seems to have come down. |
| 3 | Model\_03 | Classic Convolutional 3D model with 5 (Conv3D layer+Activation+Pooling) + 5 Dense Layers.  image size 120x120; epochs=20; batch\_size=20.  Reduced frame rate to 20 frames (alternate frames). Adjusted the Learning Rate Plateau factor.  Model Parameters: 11,69,093  Trainable Parameters: 11,69,093 | Accuracy: 100.00  Validation Accuracy: 92.00  Validation loss: 0.2748 | Further improvement in the validation loss. Accuracy has also improved. This could be a better model to move ahead. |
| 4 | Model\_04 | Classic Convolutional 3D model with 5 (Conv3D layer+Activation+Pooling) + 5 Dense Layers.  image size 100x100; epochs=20; batch\_size=20.  Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 11,69,093  Trainable Parameters: 11,69,093 | Accuracy: 92.91  Validation Accuracy: 86.00  Validation loss: 0.4529 | Reduction of image size has not helped improvement of the model. As the accuracy, validation accuracy and validation loss have degraded. |
| 5 | Model\_05 | Classic Convolutional 3D model with 5 (Conv3D layer+Activation+Pooling) + 5 Dense Layers.  image size 120x120 – Gray-scaled Images ; epochs=20; batch\_size=30.  Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 21,01,381  Trainable Parameters: 21,01,381 | Accuracy: 91.55  Validation Accuracy: 85.00  Validation loss: 0.4138 | The validation loss seems very much inline with the training loss and has a negligible difference. The accuracy difference has also come down. This can also be a better model. |
| 6 | Model\_06 | Classic Convolutional 3D model with 5 (Conv3D layer+Activation+Pooling) + 5 Dense Layers.  image size 120x120 – Gray-scaled Images ; epochs=20; batch\_size=10.  Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 21,01,381  Trainable Parameters: 21,01,381 | Accuracy: 99.40  Validation Accuracy: 83.00  Validation loss: 0.5240 | The model started to overfit as the accuracy, loss differences across training and validation data have increased |
| 7 | Model\_07 | CNN + RNN model. A time distributed CNN with 5 layers followed by a LSTM(64). Running on image size 120x120; epochs=20; batch\_size=30, Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 10,05,541  Trainable Parameters: 10,04,549 | Accuracy: 83.41  Validation Accuracy: 36.00  Validation loss: 1.7382 | The model doesn’t show a good accuracy levels and the validation accuracy has further gone down.. Also, the validation loss is high. |
| 8 | Model\_08 | CNN + RNN model. A time distributed CNN with 5 layers followed by a GRU(64). Running on image size 120x120; epochs=20; batch\_size=30, Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 8,54,117  Trainable Parameters: 8,53,125 | Accuracy: 79.03  Validation Accuracy: 42.00  Validation loss: 1.5774 | The model’s accuracy has gone down along with the parameters count. The models’ validation accuracy has improved.. but the validation loss is still very high. |
| 9 | Model\_09 | CNN-RNN Model – CNN-Transfer Learning of MobileNet. RNN Model – a LSTM(128)… Running on image size 120x120; epochs=20; batch\_size=30, Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 38,23,941  Trainable Parameters: 5,93,029 | Accuracy: 100.00  Validation Accuracy: 79.00  Validation loss: 0.5678 | There is a difference between the validation and training accuracy (overfitting). Also, the validation loss is high. |
| 10 | Model\_10 | CNN-RNN Model – CNN-Transfer Learning of MobileNet. RNN Model – a GRU(128)… Running on image size 120x120; epochs=20; batch\_size=30, Reduced frame rate to 20 frames (alternate frames).  Model Parameters: 36,76,741  Trainable Parameters: 4,45,829 | Accuracy: 100.00  Validation Accuracy: 79.00  Validation loss: 0.5915 | There is a difference between the validation and training accuracy (overfitting). Also, the validation loss is high. |

**Overall Result:** The model\_03 is the best model from the experimented models.