Images resizing is done to 120\*120. Batch size is changed in the model mentioned in the description. Input shape is (30,120,120,3) throughout. Mostly Adam optimizer is used.

First, tried 6 different Conv3D models and the best one is selected as our final model. After that we used 4 CNN+RNN architecture. One contains LSTM, one contains GRU, other two contain transfer learning. Out of all the 10 models, Model 6 (Conv3D) model is selected as best model since it gives highest train and validation accuracy.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| 1 | Conv3D | Train\_Accuracy = 22.3%  Val\_Accuracy = 21.0%  Epochs = 20 | 2 Conv3D(32 and 64) and 2 dense layers (128 and 256) used along with the dropouts, Batch Normalization and Pooling.  Batch size = 8  The model doesn’t overfit but the accuracy is low |
| 2 | Conv3D | Train\_Accuracy = 21.4%  Val\_Accuracy = 39.0%  Epochs = 20 | Increased the batch size to see if the results change.  Batch Size = 16  The model is underfitting and the accuracy is still low |
| 3 | Conv3D | Train\_Accuracy = 58.5%  Val\_Accuracy = 25.0%  Epochs = 20 | Added a Conv3D layer (256), reduced one dense layer (128) and increased batch size  Batch size = 32  The train accuracy increased but the model is overfitting |
| 4 | Conv3D | Train\_Accuracy =81.7%  Val\_Accuracy = 25%  Epochs = 20 | Increased batch size on the same architecture  Batch size = 64  The train accuracy further increased but the model is still overfitting |
| 5 | Conv3D | Train\_Accuracy = 93.7%  Val\_Accuracy = 24.0%  Epochs = 20 | Removed dense layer (128), added dense layer (512) instead and decreased batch size  Batch size = 32  The train accuracy further increased but the model is overfitting |
| 6 | Conv3D | Train\_Accuracy = 93.9%  Val\_Accuracy = 89.0%  Epochs = 20 | Decreased batch size further and increased the pooling size and kernel size (from (2,2,2) to (3,3,3)) to address the overfitting issue  Batch size = 8  **This model seems fine since it gives the best results** |
| 7 | Conv2D and LSTM | Train\_Accuracy = 94.9%  Val\_Accuracy = 38.0%  Epochs = 20 | Used 4 Conv2D layer (16,32,64 and 128) and a LSTM layer (64) along with the time distributed layer, Batch Normalization, Dropouts and Max Poling layer  Batch size = 32  The training accuracy is high but the validation is low and the model is overfitting |
| 8 | Conv2D and GRU | Train\_Accuracy = 91.4%  Val\_Accuracy = 24.0%  Epcohs = 20 | Using GRU instead of LSTM so that it has less parameters. Tweaked layers of above architecture by keeping 3 Conv2D layer (16,32,64) and a GRU layer (64) along with the time distributed layer, Batch Normalization, Dropouts and Max Poling layer  Batch size = 64  The training accuracy is high but the model is still overfitting |
| 9 | Using Transfer Learning (ResNet50) | Train\_Accuracy = 56.6%  Val\_Accuracy = 49.0%  Epochs = 20 | To address the above issues, used ResNet50 and added a GRU layer(64).  Batch size = 64  The model overfitting issue is relatively resolved but the accuracy is really low |
| 10 | Using Transfer Learning (MobileNet) | Train\_Accuracy = 88.4%  Val\_Accuracy = 79.0%  Epochs = 20 | Used MobileNet and added a GRU layer (64) and a Dense layer (64)  Batch size = 64  Model overfitting is relatively low but still not resolved and the accuracy is lesser than selected Conv3D model |
| **Final Model ( Model 6 )** | **Conv3D** | **Train\_Accuracy = 93.9%**  **Val\_Accuracy = 89.0%**  **Epochs = 20** | **Good training and validation accuracy. The model contains optimum parameters.** |