|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D** | **Model not trainable – Getting Resource exhausted exception** | **Explanation:** Initially we considered batch size equal to total number of images in the train data set.  **Decision:** Reduce the batch size to half the total number of images in the train data set. |
| **2** | **Conv3D** | **Model not trainable – Getting Resource exhausted exception** | **Explanation:** Reduce the batch size to half the total number of images in the train data set.  **Decision:** Reduce the batch size to 64 |
| **3** | **Conv3D** | **Validation Accuracy: 0.18** | **Explanation:** Used dropouts after convolutional and FC layers.  **Decision: R**emove dropouts after Convolutional layer and retain after FC layer. Also use batch normalization after every convolutional layer. |
| **3** | **Conv3D** | **Validation Accuracy: 0.54** | **Explanation:** Remove dropouts after Convolutional layer and retain after FC layer. Also use batch normalization after every convolutional layer.  **Decision:** Use batch normalization and dropouts after every convolutional layer. Also, retain the dropouts in the FC layer |
| **4** | **Conv3D** | **Validation Accuracy: 0.56** | **Explanation:** Use batch normalization and dropouts after every convolutional layer. Also, retain the dropouts in the FC layer  **Decision:** Remove the dropouts after the convolutional layers and use L2 regularization in the FC layer. Retain the dropouts in FC. |
| **5** | **Conv3D** | **Validation Accuracy: 0.60** | **Explanation:** Tried with image size 120,160  **Decision:** Reduce the image size to 50,50 and the Accuracy increased to 0.70 |
| **6** | **All models** | **Validation Accuracy: 0.60** | **Explanation:** Tried with considering alternate frames from the 15th frame in the generator function.  **Decision:** Use every 3rd frame starting from the first frame. |
| **6** | **ConvLSTM** | **Validation Accuracy: 0.875**  **Training Accuracy: 0.877** | **# training sequences = 663**  **# validation sequences = 100**  **# epochs = 50**  **# batch size = 64**  **Total params: 6,710,309**  **Trainable params: 6,710,117**  **Non-trainable params: 192** |
| **7** | **ConvGRU** | **Validation Accuracy: 0.70**  **Training Accuracy: 0.83** | **# training sequences = 663**  **# validation sequences = 100**  **# epochs = 50**  **# batch size = 64**  **Total params:** 3,760,933 |
| **8** | **Conv3D** | **Validation Accuracy: 0.73**  **Training Accuracy: 0.96** | **# training sequences = 663**  **# validation sequences = 100**  **# epochs = 50**  **# batch size = 64**  **Total params:** 263,397 |
|  |  |  |  |
| **Final Model** | **ConvLSTM** | **Validation Accuracy: 0.875**  **Training Accuracy: 0.877** | Here I am choosing the Epoch 49 model as the validation accuracy and training accuracy are close indicating a good, trained model. Since training accuracy and validation accuracy are nearby this model is not overfitting.  Hence, we are choosing ConvLSTM to be the final model. |

For other models like ConvGRU and Conv3D, there is lot of difference between training accuracy and validation accuracy which indicates overfitting, hence we are not choosing those models.