**Model 1 – Conv3D**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exp No** | **Model** | **Dataset** | **Result** | **Decision + Explanation** |
| 1 | Conv3D | 360\*360 images  6 frames/video  45 training  15 validation | Throws OOM error | Cropping and resizing the images to 224\*224 |
| 2 | Conv3D | 224\*224 images  6 frames/video  45 training  15 validation | Total params: 12,625,269  Training Accuracy: 0.31  Training time: 8s | Increase number of layers. Adding a Conv3D (32), Activation and BatchNormalization Layer |
| 3 | Conv3D | 224\*224 images  6 frames/video  45 training  15 validation | Total params: 50,489,557  Training Accuracy: 0.4  Training time: 12s | Increase the number of frames per video. |
| 4 | Conv3D | 224\*224 images  10 frames/video  45 training  15 validation | Total params: 100,956,373  Throws OOM error  Training Accuracy: 0.4  Training time: 12s | ResourceExhaustedError OOM when allocating tensor. Therefore, decreased the batch size to 8. Each batch will have 8\*10 images. However, adding more layers.  Conv3D (64)\*2, Conv3d(128)\*2  Increasing epochs to 10 |
| 5 | Conv3D | 224\*224 images  10 frames/video  45 training  15 validation  Batch size 8  Epochs 10 | Total params:  49,798,549  Training Accuracy: 0.77  Validation Accuracy remains at 0.2 | Overfitting on a small sample of training data. In the next step will use the entire training data. |
| 6 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 10 | Total params:  49,798,549  Accuracy: 0.31  Validation Accuracy: 0.21  Training time: Approx 40 minutes | The accuracy does not seem to improve after successive epochs. Model is heavily overfitting as well. Add dropouts in the first 2 convolution layer. Remove the 64 and 128 layers of Conv3D as accuracy is not improving. |
| 7 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 10 | Total params:  48,645,213  Accuracy: 0.31  Validation Accuracy: 0.19  Training time: Approx 40 minutes | Decreased the dense connections as validation accuracy is abysmal. Change learning rate. Run for more epochs. |
| 8 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25 | Total params:  48,645,213  Accuracy: 0.88  Validation Accuracy: 0.46  Training time: Approx 60 minutes | Missed to add a drop out of 0.5 in the final dense layer. Adding and running again |
| 9 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25 | Total params:  48,645,213  Accuracy: 0.27  Validation Accuracy: 0.20  Training time: Approx 60 minutes | Drop out is too steep and performance is widely affected. Removing 64 and 128 Conv3D layers to verify if it is even required. Changing Dense Layer to 128 and removing drop out |
| 10 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25  LR 0.001  factor= 0.125, patience=3, min\_lr=0.0001 | Total params:  11,982,045  loss: 1.6729 –  Training accuracy: 0.9140  val\_loss: 7.3101  Validation Accuracy: 0.6000  Training time: Approx 40 minutes | Best accuracy seen so far. Still has signs of overfitting. Will add a dropout of 0.20 in Dense layer connection. |
| 11 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25  LR 0.001  factor= 0.125, patience=3, min\_lr=0.0001 | Total params:  11,982,045  loss: 1.6729 –  Training accuracy: 0.87  val\_loss: 7.3101  Validation Accuracy: 0.54  Training time: Approx 40 minutes | Both training and test accuracy dropped. Will add another layer (64) and slightly increase drop out to 0.25. Increase LR to 0.002. |
| 12 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25  LR 0.002  factor= 0.125, patience=3, min\_lr=0.0001 | Total params:  24,092,509  loss: 4.2102  Training accuracy: 0.8929  val\_loss: 41.3044 – Validation Accuracy: 0.8000 | Good improvement in validation accuracy. Can add another layer (128) and slightly increase drop out to 0.3. LR stays at 0.002. Decrease decay from 0.125 to 0.0625. |
| 12 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25  LR 0.002  factor= 0.0625, patience=3, min\_lr=0.0001 | Total params:  24,092,509  loss: 4.2102  Training accuracy: 0.8929  val\_loss: 41.3044 – Validation Accuracy: 0.8000 | Good improvement in validation accuracy. Increasing batch size to 16. |
| 13 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25  LR 0.002  factor= 0.0625, patience=3, min\_lr=0.0001 | Resource Exhausted OOM error | Revert to a batch size of 8. Increase number of epochs to 30. |
| 14 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25  LR 0.002  factor= 0.0625, patience=3, min\_lr=0.0001 | Total params:  24,092,509  loss: 4.2102  Training accuracy: 0.8929  val\_loss: 41.3044 – Validation Accuracy: 0.8000 | As validation is mostly greater than training, the dropouts might be too high. Reverting to 0.2 and 0.25 dropouts that gave good results initially |
| 15 | Conv3D | 224\*224 images  10 frames/video  All training  All validation  Batch size 8  Epochs 25  LR 0.002  factor= 0.0625, patience=3, min\_lr=0.0001 | Total params:  24,092,509  Loss: 6.33072  Training Accuracy: 0.80392  Validation Loss: 6.29401  Validation Accuracy: 0.74000 | Too many parameters. Will try reducing parameters by increasing strides and decreasing the dense 128 layer to a dense layer of 32 |
| 16 | Conv3D | 120\*120 images  10 frames/video  All training  All validation  Batch size 16  Epochs 30  LR 0.002  factor= 0.0625, patience=3, min\_lr=min\_lr=0.00001 | Total params: 3,542,485  Trainable params: 3,540,501  Stopped after 10 epochs on seeing exceedingly small improvement | Progress was too slow. Learning did not increase beyond 0.36 accuracy for training after 10 epochs. Increasing learning rate to 0.002. |
| 17 | Conv3D | 120\*120 images  10 frames/video  All training  All validation  Batch size 16  Epochs 30  LR 0.01  factor= 0.01, patience=3, min\_lr=0.00001 | Total params: 3,542,485  Trainable params: 3,540,501  loss: 0.8047  Training Accuracy: 0.7059  val\_loss: 3.1594 Validation Accuracy: 0.6800 | Decent model given low parameters. |
|  |  |  |  |  |
| **Final Model** | **Conv3D** | **120\*120 images**  **10 frames/video**  **All training**  **All validation**  **Batch size 16**  **Epochs 30**  **LR 0.01**  **factor= 0.01, patience=3, min\_lr=0.00001** | **Total params: 3,542,485**  **Trainable params: 3,540,501**  **loss: 0.8047**  **Training Accuracy: 0.7059**  **val\_loss: 3.1594 Validation Accuracy: 0.6800** | **Prefer this decent model (3M parameters) over the good model with 24M parameters** |

**Model 2 – Conv2D + RNN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exp No** | **Model** | **Dataset** | **Result** | **Decision + Explanation** |
| 1 | ImageNet + GRU | 224\*224 images  10 frames/video  45 training  15 validation | Total Params:  33,224,837  Trainable Params:  9,637,125  Training Accuracy: 0.49  Testing Accuracy: 0.2 | Ran only for 3 epochs. Need to run more to understand if training accuracy increases to overfiting while validation accuracy should remain same (Ablation run) |
| 2 | ImageNet + GRU | 224\*224 images  10 frames/video  45 training  15 validation  Epochs 20 | Total Params:  33,224,837  Trainable Params:  9,637,125  Training Accuracy: 0.8  Testing Accuracy: 0.2 | The resnet s parameter are set as non trainable. Will change these to trainable to verify if there is any improvement |
| 3 | ImageNet + GRU | 224\*224 images  10 frames/video  45 training  15 validation  Epochs 20 | Total Params:  33,224,837  Trainable Params:  33,171,717  OOM when allocating tensor | OOM occurs. To avoid this will configure all as non trainable but last 5-10 layers. Picking 8 randomly |
| 4 | ImageNet + GRU | 224\*224 images  10 frames/video  45 training  15 validation  Epochs 25 | Total Params:  33,224,837  Trainable Params:  13,052,677  Training Accuracy: 0.73  Testing Accuracy: 0.2 | Increase no of trainable layers in ResNet to 16. Increase LR to 0.002 |
| 4 | ImageNet + GRU | 224\*224 images  10 frames/video  45 training  15 validation  Epochs 25  Adam LR = 0.002 | Total Params:  33,224,837  Trainable Params:  15,158,533  Training Accuracy: 0.57  Testing Accuracy: 0.2 | Made performance worse. Will stay with last 8 trainable layers. Increase the number of GRU layers. Have two 32, two 64 and 2 128. Followed by a dense 128 layer |
| 5 | ImageNet + GRU | 224\*224 images  10 frames/video  45 training  15 validation  Epochs 25  Adam LR = 0.002 | Total params: 62,189,573  Trainable params: 42,017,413  Training Accuracy: 0.71  Testing Accuracy: 0.2 | Made performance worse. Will stay with last 8 trainable layers. Increase the number of GRU layers. Have two 32, two 64 and 2 128. Followed by a dense 128 layer |
| 6 | ImageNet + GRU | 224\*224 images  10 frames/video  All data  Epochs 25  Adam LR = 0.002 | Total params: 62,189,573  Trainable params: 42,017,413  Training Accuracy: 0.84  Testing Accuracy: 0.2 | Highly overfitting. Increasing dropout in in between layers to 0.3 and final dense layer to 0.5. Increase LR to 0.005. Keeping only last 4-8 layers of resnet as trainable due to long training time. Increased dropouts. Decreased number of layers to just two 128s and a dense 64 |
| 7 | ImageNet + GRU | 224\*224 images  10 frames/video  All data  Epochs 25  Adam LR = 0.002 | Total params: 62,279,685  Trainable params: 42,107,525  Training Accuracy: 0.71  Testing Accuracy: 0.23 | Still highly overfitting. Decreasing layers to size 32. |
| 8 | ImageNet + GRU | 224\*224 images  10 frames/video  All data  Epochs 25  Adam LR = 0.002 | Total params: 62,279,685  Trainable params: 42,107,525  Training Accuracy: 0.71  Testing Accuracy: 0.23 | Still highly overfitting. Changing V2 of ResNet50 |
| 9 | ImageNet + GRU | 224\*224 images  10 frames/video  All data  Epochs 25  Adam LR = 0.003 | Total params: 33,210,437  Trainable params: 10,700,357  Training Accuracy: 0.63  Testing Accuracy: 0.23 | Still highly overfitting. Decreasing dropout but adding regularizer of 0.03 in dense layer |
| 10 | ImageNet + GRU | 120\*120 images  11 frames/video  All data  Epochs 25  Adam LR = 0.001 | OOM while allocating tensor | Decreasing batch size to 0.4. |
| 11 | ImageNet + GRU | 224\*224 images  11 frames/video  All data  Epochs 25  Adam LR = 0.003  Batch size 4  Regularizer 0.03 | Total params: 33,210,437  Trainable params: 10,700,357  Training Accuracy : 0.96  Validation Accuracy: 0.3 | Slightly better validation accuracy. Increasing regularizer to 0.5. Increasing dropout in final dense layer to 0.5 |
| 12 | ImageNet + GRU | 224\*224 images  10 frames/video  All data  Epochs 25  Adam LR = 0.003  Batch size 4  Regularizer 0.03 | Total params: 33,210,437  Trainable params: 10,700,357  Training Accuracy : 0.68  Validation Accuracy: 0.28 | No luck. Try to change Resnet entirely as it seems to only overfit. Trying a smaller VGG16. Remove all excessive layers and keep just the basic. Decreasing image size |
| 13 | ImageNet + GRU (VGGNet) | 120\*120images  10+1 frames/video  All data  Epochs 20  Adam LR = 0.001  Batch size 10  Regularizer 0.03 | Total params: 15,021,509  Trainable params: 306,821  Loss : 0.57095  Training Accuracy 0.79954  Val Loss: 0.87  Validation Accuracy : 0.64 | Finalizing this as it is simple and portable |
| **Final Model** | **ImageNet + GRU (VGGNet)** | **120\*120images**  **10+1 frames/video**  **All data**  **Epochs 20**  **Adam LR = 0.001**  **Batch size 10**  **Regularizer 0.03** | **Total params: 15,021,509**  **Trainable params: 306,821**  **Loss : 0.57095**  **Training Accuracy 0.79954**  **Val Loss: 0.87**  **Validation Accuracy : 0.64** | **Finalizing this as the GRU model** |