# Model Approaches:

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment Number** | **Model** | **Result** | **Decision + Explanation** |
| **1** | **Conv3D**  Batch Size – 40  Epocs – 15  Image\_size – 160 x 160 | ResourceExhaustedError: Graph execution error: OOM when allocating tensor with shape[40,18,160,160,64] and type float on /job:localhost/replica:0/task:0/device:GPU:0 by allocator GPU\_0\_bfc [[{{node sequential/activation/Relu-0-1-TransposeNCDHWToNDHWC-LayoutOptimizer}}]] |  |
| **2** | **Conv3D**  Batch Size – 8  Epochs – 30  Image Size – 160x160  Using first 18 frames of a image | ategorical\_accuracy: 0.9849  val\_loss: 0.8964  val\_categorical\_accuracy: 0.7500  Trainable params: 29,100,165 | **Slightly over fitting** |
| **3** | **Conv3D**  Batch Size – 8  Epochs – 30  Image Size – 160x160  Using alternate 18 frames of a image | Trainable params: 29,100,165  categorical\_accuracy: 0.9925  val\_loss: 0.4789  val\_categorical\_accuracy: 0.9000 | **Slightly overfitting w.r.t to training and validation scores** |
| **4** | **Conv3D**  Batch Size – 8  Epochs – 30  Image Size – 160x160  Using alternate 18 frames of a image **with data agumentation** | categorical\_accuracy: 0.9465  val\_loss: 0.3798  val\_categorical\_accuracy: 0.9000  Trainable params: 29,100,165 | **After implementing the data agumentation, the overfitting is reduced**  **@12th Epoc** |
| **5** | **Conv3D**  Batch Size – 8  Epochs – 25  Image Size – 160x160  Using alternate 18 frames of a image  Filter Size reduced to 2,2,2 with data agumentation applied | * categorical\_accuracy: 0.9713 * val\_loss: 0.4010 * val\_categorical\_accuracy: 0.9100 * Trainable params: 27,073,093 | **There is no overfitting or underfitting, 16th EPOC has given the better model with minimal validation loss and also the training parameters are reduced** |
| **6** | **Conv2D+LSTM:**  Data agumentation enabled, with alternate frame size and Early stopping enabled | categorical\_accuracy: 0.7036  val\_loss: 1.9061  val\_categorical\_accuracy: 0.3200  Trainable params: 7,527,429 | **With the applied condition, the model seems over fitting.** |
| **7** | **Conv2D +GRU** | Trainable params: 5,885,061  categorical\_accuracy: 0.8974  val\_loss: 0.4079  val\_categorical\_accuracy: 0.8700 | **Model is behaving similar to Model4 above only difference here is the no. of training parameters are less** |
| **8** | **Conv3D:**  Will all frames used as input, with input image size 160 x 160 | ResourceExhaustedError: OOM when allocating tensor with shape[16,64,30,160,160] and type float on /job:localhost/replica:0/task:0/device:GPU:0 by allocator GPU\_0\_bfc | **………………** |
| **9** | **Conv3D:**  Image size – 100x100  Batch size – 8  Alternate images - 18 |  | **Overfitting model** |
| **10** | **Conv3D**  LearningRate Chaged to 0.01  Image Size 160x160  Batch\_size - 8 | categorical\_accuracy: 0.1885  val\_loss: 1.6942  val\_categorical\_accuracy: 0.2100 | **UnderFitting Model** |

# Final Result:

1. Model 7 🡪 Conv2D with GRU 🡪 this is selected because of following reasons
   1. Good accuracy score when compared to all other models except(Model4)
   2. Minimal Loss
   3. Less Training Parameters
   4. And when compared to Conv3D models, the resource utilization is less in Conv2D+GRU

# Observations

## Model -1 :

### Input

* Batch\_size – 40 and image size – 160 x160
* And list of image frames used is 18

### Result

* We got ResourceExhaustedError (OOM Exception)

## Model -2:

### Input

* Batch Size reduced to 8, maintained the image size as is(160x160)
* Here we use the first 18 frames as input

### Result

* The model resulted in over fitting with accuracy scores (Training – 0.98, Val – 0.75)

## Model -3:

### Input

* Batch Size maintained to 8, maintained the image size as is(160x160)
* Using alternate 18 frames of a image in the input

### Result

* Slightly overfitting w.r.t to training and validation scores 0.99 and 0.90 respectively

## Model -4:

### Input

* Batch Size maintained to 8, maintained the image size as is(160x160)
* Using alternate 18 frames of a image in the input
* Data augmentation applied(set to True)

### Result

* This model performance is better than Model -3 above
* After implementing the data agumentation, the overfitting is reduced and the accuracy scores are better now (training – 0.9635 and val -0.9000)
* We can see reduction in validation loss also when compared to Model-3

## Model -5:

Note : Speicifically this model output varied drastically when I execute it multiple times, the outcomes I faces was:

1. Model Overfitting
2. Model ended-up saying Out of Memory
3. As I m using early stop mechanism with patience set to 5, the model is not improving w.r.t to validation loss, so it stopped training with 5-7 echos only( so results in write-up and notebook may vary)

### Input

* Batch Size maintained to 8, maintained the image size as is(160x160)
* Using alternate 18 frames of a image in the input
* Data augmentation applied(set to True)
* Filter size reduced to 2,2,2

### Result(Above table result)

* Model is pretty decent with no Overfitting or under fitting
* Training parameters are reduced when compared to above models
* But this modes as mentioned above is unstable time-to-time so the observations may vary

## Model -6(Conv2D + LSTM):

### Input:

* Data augmentation applied(set to True)
* Using alternate 18 frames of a image in the input
* No Change in batch size and image size

### Result:

* Model is overfitting

## Model -7:

### Input

* Batch Size maintained to 8, maintained the image size as is(160x160)
* Using alternate 18 frames of a image in the input
* Data augmentation applied(set to True)

### Result

* This model performance is Good with training – 0.8974 and val -0.8700 with validation loss 0.4)
* The no. of trainable parameters here is significally less than the Conv3D Model 4 above
* This would be my best selected model for this assignment.

## Model -8:

### Input

* Batch Size maintained to 8, maintained the image size as is(160x160)
* Using all 30 frames of a image in the input
* Data augmentation applied(set to True)

### Result

* We got ResourceExhaustedError (OOM Exception)

## Model -9:

### Input

* Batch Size maintained to 8, image size changed to 100x100
* Using alternate 18 frames of a image in the input
* Data augmentation applied(set to True)

### Result

* This model performance is Good with training – 0.8974 and val -0.8700 with validation loss 0.4)
* The no. of trainable parameters here is significally less than the Conv3D Model 4 above
* This would be my best selected model for this assignment.

## Model -10:

### Input

* Batch Size maintained to 8, image size changed to 160x160
* Using alternate 18 frames of a image in the input
* Data augmentation applied(set to True)
* LR rate changed to 0.01 from 0.001

### Result

* Under fitting model.