## Basic Model (Conv 3d)

Building a basic Conv3d model with 2 layers and performing experiments

Model: "sequential\_2"  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Layer (type) Output Shape Param #   
=================================================================  
 conv3d\_6 (Conv3D) (None, 15, 120, 120, 32) 2624   
   
 batch\_normalization\_6 (Batc (None, 15, 120, 120, 32) 128   
 hNormalization)   
   
 max\_pooling3d\_6 (MaxPooling (None, 7, 60, 60, 32) 0   
 3D)   
   
 conv3d\_7 (Conv3D) (None, 7, 60, 60, 64) 55360   
   
 batch\_normalization\_7 (Batc (None, 7, 60, 60, 64) 256   
 hNormalization)   
   
 max\_pooling3d\_7 (MaxPooling (None, 3, 30, 30, 64) 0   
 3D)   
   
 flatten (Flatten) (None, 172800) 0   
   
 dense\_2 (Dense) (None, 256) 44237056   
   
 dropout\_1 (Dropout) (None, 256) 0   
   
 dense\_3 (Dense) (None, 5) 1285   
   
=================================================================  
 Total params: 44,296,709  
 Trainable params: 44,296,517  
 Non-trainable params: 192  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Experiments:

* Batch size 662 : no memory
* Batch size 100 : no memory
* Batch size-50 , Frames per video -30, epochs-5 : no memory
* Batch size-30, Frames per video -30, epochs -5
  + loss: 1.4345 - categorical\_accuracy: 0.6335 - val\_loss: 285.0408 - val\_categorical\_accuracy: 0.2200
* Batch size-10, Frames per video -15, epochs -5
  + loss: 1.5565 - categorical\_accuracy: 0.2624 - val\_loss: 4.9574 - val\_categorical\_accuracy: 0.3400 - lr: 0.0010
* Batch size-50, Frames per video -15, epochs -5
  + loss: 1.5311 - categorical\_accuracy: 0.4600 - val\_loss: 32.0031 - val\_categorical\_accuracy: 0.2600 - lr: 0.0010
* Batch size-75, Frames per video -15, epochs -5
  + loss: 1.9469 - categorical\_accuracy: 0.6667 - val\_loss: 12.4178 - val\_categorical\_accuracy: 0.1800 - lr: 0.0010

Observation:

Drop in accuracy by reducing the number of frames per video

Increase in accuracy by increasing the batch size

## Adding more layers to above model Convo3D

As the above model is not able to capture the complexity, we have added more layers to the model

Model: "sequential\_3"  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Layer (type) Output Shape Param #   
=================================================================  
 conv3d\_8 (Conv3D) (None, 15, 120, 120, 64) 5248   
   
 batch\_normalization\_8 (Batc (None, 15, 120, 120, 64) 256   
 hNormalization)   
   
 max\_pooling3d\_8 (MaxPooling (None, 7, 60, 60, 64) 0   
 3D)   
   
 conv3d\_9 (Conv3D) (None, 7, 60, 60, 128) 221312   
   
 batch\_normalization\_9 (Batc (None, 7, 60, 60, 128) 512   
 hNormalization)   
   
 max\_pooling3d\_9 (MaxPooling (None, 3, 30, 30, 128) 0   
 3D)   
   
 conv3d\_10 (Conv3D) (None, 3, 30, 30, 256) 884992   
   
 batch\_normalization\_10 (Bat (None, 3, 30, 30, 256) 1024   
 chNormalization)   
   
 max\_pooling3d\_10 (MaxPoolin (None, 1, 15, 15, 256) 0   
 g3D)   
   
 global\_average\_pooling3d\_1 (None, 256) 0   
 (GlobalAveragePooling3D)   
   
 dense\_4 (Dense) (None, 512) 131584   
   
 dropout\_2 (Dropout) (None, 512) 0   
   
 dense\_5 (Dense) (None, 5) 2565   
   
=================================================================  
 Total params: 1,247,493  
 Trainable params: 1,246,597  
 Non-trainable params: 896  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Experiments

* Batch size - 30, Frames per video- 15, epochs -5
  + loss: 0.6112 - categorical\_accuracy: 0.7602 - val\_loss: 2.8803 - val\_categorical\_accuracy: 0.1800 - lr: 0.0010

Observation

* Accuracy got increased , but also resulted in overfitting

## Regularization to above model Conv3D

As we can see the overfitting of the model (difference between training and validation accuracy) , we tried to increase dropout and regularization

Model: "sequential\_2"  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Layer (type) Output Shape Param #   
=================================================================  
 conv3d\_3 (Conv3D) (None, 15, 120, 120, 64) 5248   
   
 batch\_normalization\_3 (Batc (None, 15, 120, 120, 64) 256   
 hNormalization)   
   
 max\_pooling3d\_3 (MaxPooling (None, 7, 60, 60, 64) 0   
 3D)   
   
 dropout\_4 (Dropout) (None, 7, 60, 60, 64) 0   
   
 conv3d\_4 (Conv3D) (None, 7, 60, 60, 128) 221312   
   
 batch\_normalization\_4 (Batc (None, 7, 60, 60, 128) 512   
 hNormalization)   
   
 max\_pooling3d\_4 (MaxPooling (None, 3, 30, 30, 128) 0   
 3D)   
   
 dropout\_5 (Dropout) (None, 3, 30, 30, 128) 0   
   
 conv3d\_5 (Conv3D) (None, 3, 30, 30, 256) 884992   
   
 batch\_normalization\_5 (Batc (None, 3, 30, 30, 256) 1024   
 hNormalization)   
   
 max\_pooling3d\_5 (MaxPooling (None, 1, 15, 15, 256) 0   
 3D)   
   
 dropout\_6 (Dropout) (None, 1, 15, 15, 256) 0   
   
 global\_average\_pooling3d\_1 (None, 256) 0   
 (GlobalAveragePooling3D)   
   
 dense\_2 (Dense) (None, 512) 131584   
   
 dropout\_7 (Dropout) (None, 512) 0   
   
 dense\_3 (Dense) (None, 5) 2565   
   
=================================================================  
 Total params: 1,247,493  
 Trainable params: 1,246,597  
 Non-trainable params: 896  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Experiments:

* Batch size- 30: Frames per video- 15, epoch 5
  + loss: 1.5038 - categorical\_accuracy: 0.6063 - val\_loss: 3.8467 - val\_categorical\_accuracy: 0.1600 - lr: 0.0010
* Batch size- 10 Frames per video- 30, epoch 5
  + loss: 1.4031 - categorical\_accuracy: 0.5551 - val\_loss: 4.6684 - val\_categorical\_accuracy: 0.1600 - lr: 0.0010

Observation:

* Regularization also reduced the accuracy of the model

## New Model Comprising CNN + LSTM

Overall accuracy of the Conv3d is not great so we will explore the combination of CNN along with LSTM

Applied Transfer learning from Resnet50 and freezes the params for Resnet50

Layer (type) Output Shape Param #   
=================================================================  
 input\_5 (InputLayer) [(None, 30, 120, 120, 3) 0   
 ]   
   
 TimeDistributed\_CNN (TimeDi (None, 30, 2048) 23587712   
 stributed)   
   
 LSTM\_Layer (LSTM) (None, 128) 1114624   
   
 Output\_Layer (Dense) (None, 5) 645   
   
=================================================================  
 Total params: 24,702,981  
 Trainable params: 1,115,269  
 Non-trainable params: 23,587,712  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Experiment:

* Batch size-10 , Frames per video =30, epoch -5
  + loss: 1.4378 - categorical\_accuracy: 0.3529 - val\_loss: 1.4212 - val\_categorical\_accuracy: 0.3500 - lr: 0.0010
* Batch size-30 , Frames per video =30, epoch -5
  + loss: 1.5242 - categorical\_accuracy: 0.3062 - val\_loss: 1.4008 - val\_categorical\_accuracy: 0.3300

Observation:

* Poor accuracy due to less depth

## Using Bi-directional LSTM

To capture more information about the video we are using bi directional LSTM

Layer (type) Output Shape Param #   
=================================================================  
 input\_1 (InputLayer) [(None, 30, 120, 120, 3) 0   
 ]   
   
 TimeDistributed\_CNN (TimeDi (None, 30, 256) 24112256   
 stributed)   
   
 Bidirectional\_LSTM\_Layer (B (None, 512) 1050624   
 idirectional)   
   
 Output\_Layer (Dense) (None, 5) 2565   
   
=================================================================  
 Total params: 25,165,445  
 Trainable params: 1,577,733  
 Non-trainable params: 23,587,712  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Experiment:

* Batch Size - 30, frames per video - 30 , epoch 5
  + loss: 1.9887 - categorical\_accuracy: 0.6591 - val\_loss: 5.0402 - val\_categorical\_accuracy: 0.2300 - lr: 0.001

Observation:

* Accuracy got improved significantly but resulted in overfitting

## Allowing change of Resnet params with Bidirectional model

Layer (type) Output Shape Param #   
=================================================================  
 input\_1 (InputLayer) [(None, 30, 120, 120, 3) 0   
 ]   
   
 TimeDistributed\_CNN (TimeDi (None, 30, 512) 24636800   
 stributed)   
   
 Bidirectional\_LSTM\_Layer (B (None, 512) 1574912   
 idirectional)   
   
 Output\_Layer (Dense) (None, 5) 2565   
   
=================================================================  
 Total params: 26,214,277  
 Trainable params: 2,626,565  
 Non-trainable params: 23,587,712  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Experiment:

* Batch size- 30, frames per video-30, epoch 5
  + loss: 2.4814 - categorical\_accuracy: 0.6501 - val\_loss: 2.7880 - val\_categorical\_accuracy: 0.3300 - lr: 0.0010

Observation:

* No much improvement in the accuacy but minor reduction of over fitting

## Adding 2 GRU in place of LSTM

Layer (type) Output Shape Param #   
=================================================================  
 input\_1 (InputLayer) [(None, 30, 120, 120, 3) 0   
 ]   
   
 TimeDistributed\_CNN (TimeDi (None, 30, 512) 24636800   
 stributed)   
   
 Bidirectional\_GRU\_Layer\_1 ( (None, 30, 512) 1182720   
 Bidirectional)   
   
 Bidirectional\_GRU\_Layer\_2 ( (None, 256) 493056   
 Bidirectional)   
   
 Output\_Layer (Dense) (None, 5) 1285   
   
=================================================================  
 Total params: 26,313,861  
 Trainable params: 2,726,149  
 Non-trainable params: 23,587,712  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Experiment:

* Batch size- 30, frames per vide0-30, epoch- 5
  + loss: 3.5103 - categorical\_accuracy: 0.7149 - val\_loss: 4.0950 - val\_categorical\_accuracy: 0.2300 - lr: 0.0010

Observation:

* Accuracy got improved but resulted in overfitting

### Final Model (Regularization of above)

Layer (type) Output Shape Param #   
=================================================================  
 input\_3 (InputLayer) [(None, 30, 120, 120, 3) 0   
 ]   
   
 TimeDistributed\_CNN (TimeDi (None, 30, 512) 24636800   
 stributed)   
   
 Bidirectional\_GRU\_Layer\_1 ( (None, 30, 1024) 3151872   
 Bidirectional)   
   
 Bidirectional\_GRU\_Layer\_2 ( (None, 432) 1609632   
 Bidirectional)   
   
 Output\_Layer (Dense) (None, 5) 2165   
   
=================================================================  
 Total params: 29,400,469  
 Trainable params: 5,812,757  
 Non-trainable params: 23,587,712  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Experiment:

* Batch size -30, frame per video 30, epoch 5
  + loss: 118.2519 - categorical\_accuracy: 0.5339 - val\_loss: 110.9952 - val\_categorical\_accuracy: 0.4300 - lr: 1.0000e-04

Observation:

* Regularization resulted in reduced accuracy at the same time resulted in reduction of overfitting