Experiment Number : 1

Details :

Batch Size : 64
Image Size : 120X120
Frames : All

Number of Layers : 3 Conv+pool; 2 Dense; 2
Dropouts : 0.2 after 2nd Conv+pooling, 0.2 after 3rd Conv+pooling

epochs : 20

Model :

Conv3D

Result :

Throws OOM error

Decision + Explanation :

Since the GPU couldn’t handle the batch size of 64 with 120X120 image size

Reduce the image size : 90X90

Experiment Number : 2

Details :

Batch Size : 64
Image Size : 90X90
Frames : All

Number of Layers : 3 Conv+pool; 2 Dense;
Dropouts :0.2 after 2nd Conv+pooling, 0.2 after 3rd Conv+pooling

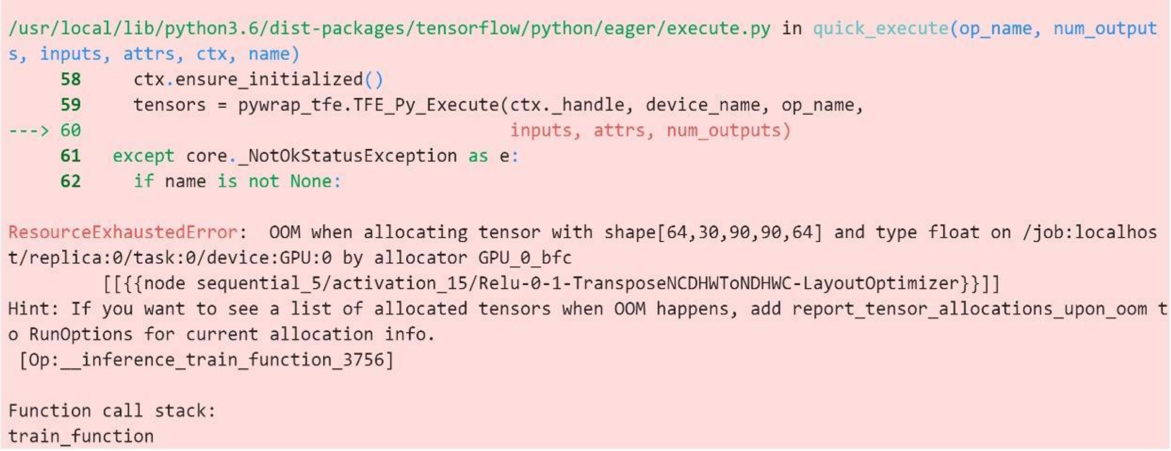
epochs : 20

Model :

Conv3D

Result :

Throws OOM error



```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/execute.py in quick_execute(op_name, num_output
s, inputs, attrs, ctx, name)
     58     ctx.ensure_initialized()
     59     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
---> 60                                         inputs, attrs, num_outputs)
     61   except core._NotOkStatusException as e:
     62     if name is not None:

ResourceExhaustedError:  OOM when allocating tensor with shape[64,30,90,90,64] and type float on /job:localhos
t/replica:0/task:0/device:GPU:0 by allocator GPU_0_bfc
	 [[{{node sequential_5/activation_15/Relu-0-1-TransposeNCDHWToNDHWC-LayoutOptimizer}}]]
Hint: If you want to see a list of allocated tensors when OOM happens, add report_tensor_allocations_upon_oom t
o RunOptions for current allocation info.
 [Op:__inference_train_function_3756]

Function call stack:
train_function
```

Decision + Explanation :

Since the GPU still couldn’t handle the batch size of 64 with 90X90 image size,

Reduced Batch size : 32
Reduced Number of Frames : 24

Experiment Number : 3

Details :

Batch Size : 32
Image Size : 90X90
Frames : 24

Number of Layers : 3 Conv+pool; 2 Dense
Dropouts : 0.2 after 2nd Conv+poolng, 0.2 after 3rd Conv+pooling

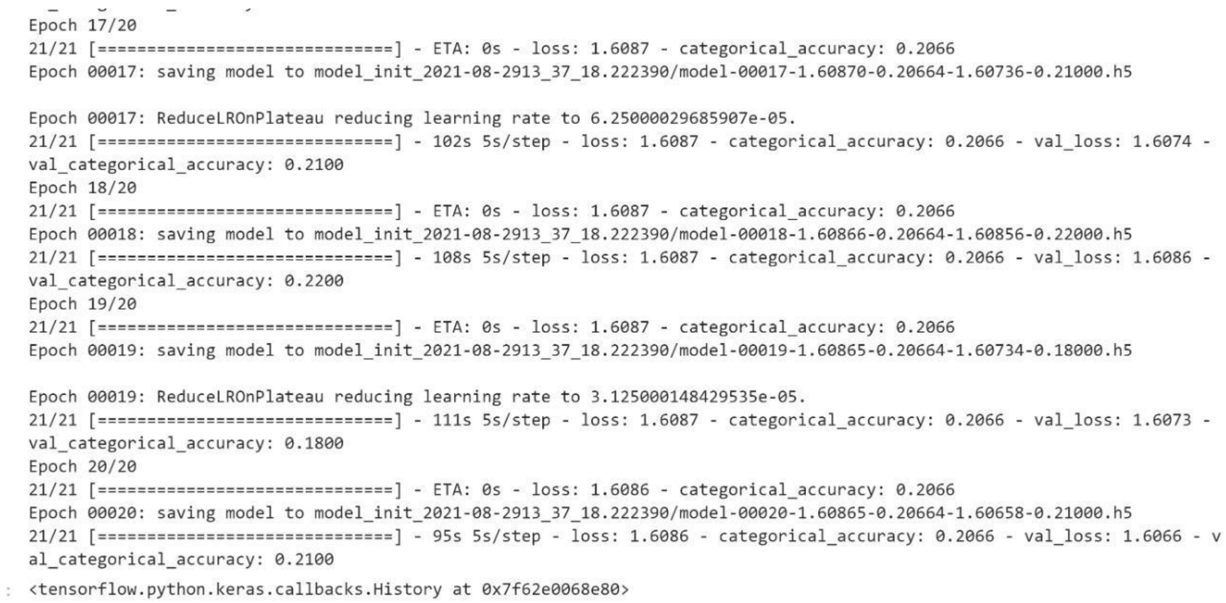
epochs : 20

Model :

Conv3D

Result :

Train Accuracy: 20.6%
Val Accuracy: 21.0%



```
Epoch 17/20
21/21 [==============================] - ETA: 0s - loss: 1.6087 - categorical_accuracy: 0.2066
Epoch 00017: saving model to model_init_2021-08-2913_37_18.222390/model-00017-1.60870-0.20664-1.60736-0.21000.h5

Epoch 00017: ReduceLROnPlateau reducing learning rate to 6.25000029685907e-05.
21/21 [==============================] - 102s 5s/step - loss: 1.6087 - categorical_accuracy: 0.2066 - val_loss: 1.6074 -
val_categorical_accuracy: 0.2100
Epoch 18/20
21/21 [==============================] - ETA: 0s - loss: 1.6087 - categorical_accuracy: 0.2066
Epoch 00018: saving model to model_init_2021-08-2913_37_18.222390/model-00018-1.60866-0.20664-1.60856-0.22000.h5
21/21 [==============================] - 108s 5s/step - loss: 1.6087 - categorical_accuracy: 0.2066 - val_loss: 1.6086 -
val_categorical_accuracy: 0.2200
Epoch 19/20
21/21 [==============================] - ETA: 0s - loss: 1.6087 - categorical_accuracy: 0.2066
Epoch 00019: saving model to model_init_2021-08-2913_37_18.222390/model-00019-1.60865-0.20664-1.60734-0.18000.h5

Epoch 00019: ReduceLROnPlateau reducing learning rate to 3.125000148429535e-05.
21/21 [==============================] - 111s 5s/step - loss: 1.6087 - categorical_accuracy: 0.2066 - val_loss: 1.6073 -
val_categorical_accuracy: 0.1800
Epoch 20/20
21/21 [==============================] - ETA: 0s - loss: 1.6086 - categorical_accuracy: 0.2066
Epoch 00020: saving model to model_init_2021-08-2913_37_18.222390/model-00020-1.60865-0.20664-1.60658-0.21000.h5
21/21 [==============================] - 95s 5s/step - loss: 1.6086 - categorical_accuracy: 0.2066 - val_loss: 1.6066 - v
al_categorical_accuracy: 0.2100
: <tensorflow.python.keras.callbacks.History at 0x7f62e0068e80>
```

Decision + Explanation :

The model was trainable but the TRAIN AND VALIDATION accuracy is very low(~20%).
This implies that the model is not trained adequately to distinguish the 5 classes accurately.

Decision :
Add one more conv3d+pooling layer with 20% dropout for better training of the model.
To make the model less complex we considered only alternate frames from each video (15 frames per video).

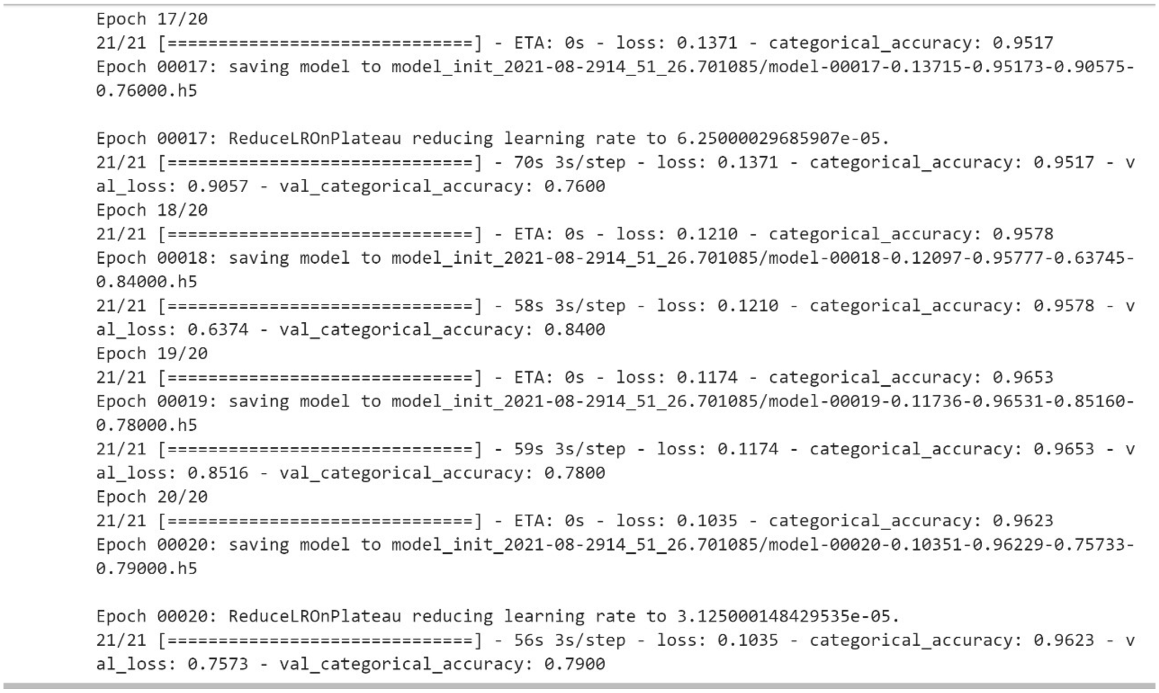
Experiment Number : 4

Details:
Batch Size : 32
Image Size : 90X90
Frames : 15

Number of Layers : 4 Conv+pool; 2 Dense
Dropouts (0.2) : after 2nd Conv+pooling, 3rd Conv+pooling and 4th Conv+pooling layers
epochs : 20

Model :
Conv3D

Result :
Training Accuracy: 96.23%
Val Accuracy: 79.0%



```
Epoch 17/20
21/21 [==============================] - ETA: 0s - loss: 0.1371 - categorical_accuracy: 0.9517
Epoch 00017: saving model to model_init_2021-08-2914_51_26.701085/model-00017-0.13715-0.95173-0.90575-
0.76000.h5

Epoch 00017: ReduceLROnPlateau reducing learning rate to 6.25000029685907e-05.
21/21 [==============================] - 70s 3s/step - loss: 0.1371 - categorical_accuracy: 0.9517 - v
al_loss: 0.9057 - val_categorical_accuracy: 0.7600
Epoch 18/20
21/21 [==============================] - ETA: 0s - loss: 0.1210 - categorical_accuracy: 0.9578
Epoch 00018: saving model to model_init_2021-08-2914_51_26.701085/model-00018-0.12097-0.95777-0.63745-
0.84000.h5
21/21 [==============================] - 58s 3s/step - loss: 0.1210 - categorical_accuracy: 0.9578 - v
al_loss: 0.6374 - val_categorical_accuracy: 0.8400
Epoch 19/20
21/21 [==============================] - ETA: 0s - loss: 0.1174 - categorical_accuracy: 0.9653
Epoch 00019: saving model to model_init_2021-08-2914_51_26.701085/model-00019-0.11736-0.96531-0.85160-
0.78000.h5
21/21 [==============================] - 59s 3s/step - loss: 0.1174 - categorical_accuracy: 0.9653 - v
al_loss: 0.8516 - val_categorical_accuracy: 0.7800
Epoch 20/20
21/21 [==============================] - ETA: 0s - loss: 0.1035 - categorical_accuracy: 0.9623
Epoch 00020: saving model to model_init_2021-08-2914_51_26.701085/model-00020-0.10351-0.96229-0.75733-
0.79000.h5

Epoch 00020: ReduceLROnPlateau reducing learning rate to 3.125000148429535e-05.
21/21 [==============================] - 56s 3s/step - loss: 0.1035 - categorical_accuracy: 0.9623 - v
al_loss: 0.7573 - val_categorical_accuracy: 0.7900
```

Decision + Explanation :
The model’s overall accuracy turns out to be good. However, the model seems to have overfit as there is a significant gap between the train and validation accuracy.

Decision :
Adding a dropout of 50% after the first dense layer.
Increasing the number of epochs to 30.

Experiment Number : 5

Details:

Batch size : 32
Image Size : 90X90
Frames : 15

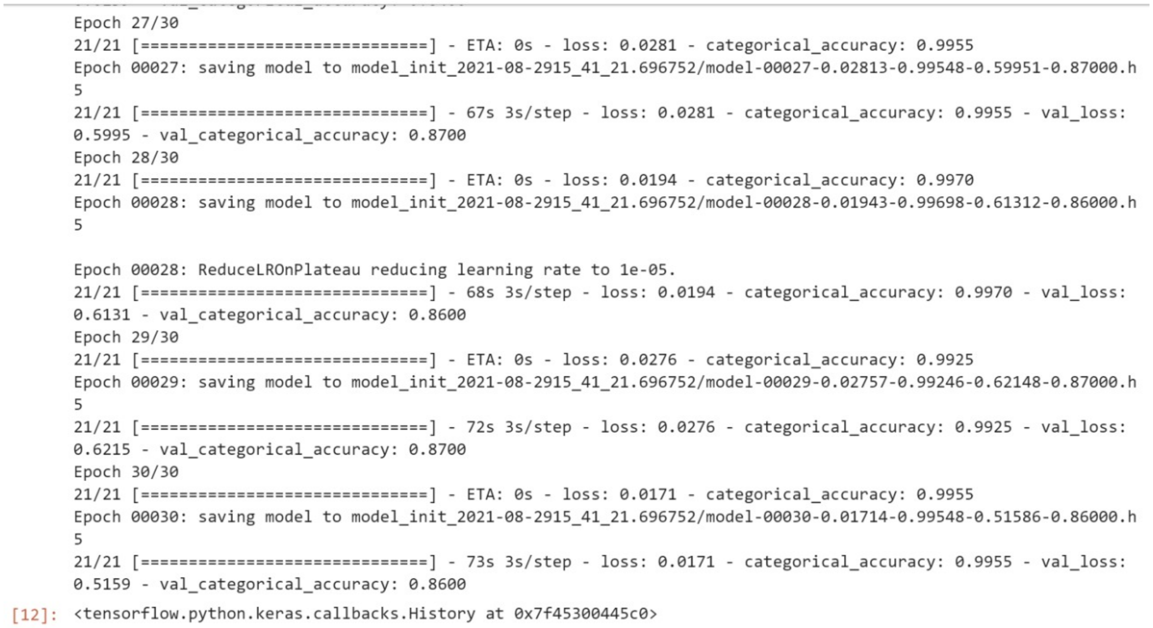
Number of Layers : 4 Conv+pool; 2 Dense
Dropouts (0.2) : after 2nd Conv+pooling, 3rd Conv+pooling and 4th Conv+pooling layers.
1 dropout of 0.5 after 1st dense layer

epochs : 30

Model :
Conv3D

Result :

Train accuracy : 99.54%
Val Accuracy : 86.00%



```
Epoch 27/30
21/21 [==============================] - ETA: 0s - loss: 0.0281 - categorical_accuracy: 0.9955
Epoch 00027: saving model to model_init_2021-08-2915_41_21.696752/model-00027-0.02813-0.99548-0.59951-0.87000.h
5
21/21 [==============================] - 67s 3s/step - loss: 0.0281 - categorical_accuracy: 0.9955 - val_loss:
0.5995 - val_categorical_accuracy: 0.8700
Epoch 28/30
21/21 [==============================] - ETA: 0s - loss: 0.0194 - categorical_accuracy: 0.9970
Epoch 00028: saving model to model_init_2021-08-2915_41_21.696752/model-00028-0.01943-0.99698-0.61312-0.86000.h
5

Epoch 00028: ReduceLROnPlateau reducing learning rate to 1e-05.
21/21 [==============================] - 68s 3s/step - loss: 0.0194 - categorical_accuracy: 0.9970 - val_loss:
0.6131 - val_categorical_accuracy: 0.8600
Epoch 29/30
21/21 [==============================] - ETA: 0s - loss: 0.0276 - categorical_accuracy: 0.9925
Epoch 00029: saving model to model_init_2021-08-2915_41_21.696752/model-00029-0.02757-0.99246-0.62148-0.87000.h
5
21/21 [==============================] - 72s 3s/step - loss: 0.0276 - categorical_accuracy: 0.9925 - val_loss:
0.6215 - val_categorical_accuracy: 0.8700
Epoch 30/30
21/21 [==============================] - ETA: 0s - loss: 0.0171 - categorical_accuracy: 0.9955
Epoch 00030: saving model to model_init_2021-08-2915_41_21.696752/model-00030-0.01714-0.99548-0.51586-0.86000.h
5
21/21 [==============================] - 73s 3s/step - loss: 0.0171 - categorical_accuracy: 0.9955 - val_loss:
0.5159 - val_categorical_accuracy: 0.8600
[12]: <tensorflow.python.keras.callbacks.History at 0x7f45300445c0>
```

Decision + Explanation :
Increase the 0.2 dropouts to 0.25 as the model still is slightly overfit.

Experiment Number : 6

Details:

Batch size : 32
Image Size : 90X90
Frames : 15

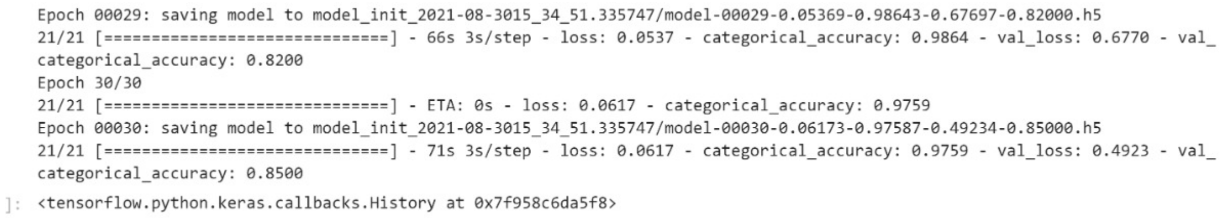
Number of Layers : 4 Conv+pool; 2 Dense
Dropouts (0.25) : after 2nd Conv+pooling, 3rd Conv+pooling and 4th Conv+pooling layer.
1 dropout (0.5) after the 1st dense layer.

epochs : 30

Model :
Conv3D

Result :

Train accuracy : 97.59%
Val Accuracy : 85.00%



```
    Epoch 00029: saving model to model_init_2021-08-3015_34_51.335747/model-00029-0.05369-0.98643-0.67697-0.82000.h5
    21/21 [==============================] - 66s 3s/step - loss: 0.0537 - categorical_accuracy: 0.9864 - val_loss: 0.6770 - val_
    categorical_accuracy: 0.8200
    Epoch 30/30
    21/21 [==============================] - ETA: 0s - loss: 0.0617 - categorical_accuracy: 0.9759
    Epoch 00030: saving model to model_init_2021-08-3015_34_51.335747/model-00030-0.06173-0.97587-0.49234-0.85000.h5
    21/21 [==============================] - 71s 3s/step - loss: 0.0617 - categorical_accuracy: 0.9759 - val_loss: 0.4923 - val_
    categorical_accuracy: 0.8500
]:  <tensorflow.python.keras.callbacks.History at 0x7f958c6da5f8>
```

Decision + Explanation :
Increase the dropout after the last conv3D and pooling layer to 0.4 as the model is still slightly overfit.

Experiment Number : 7

Details:

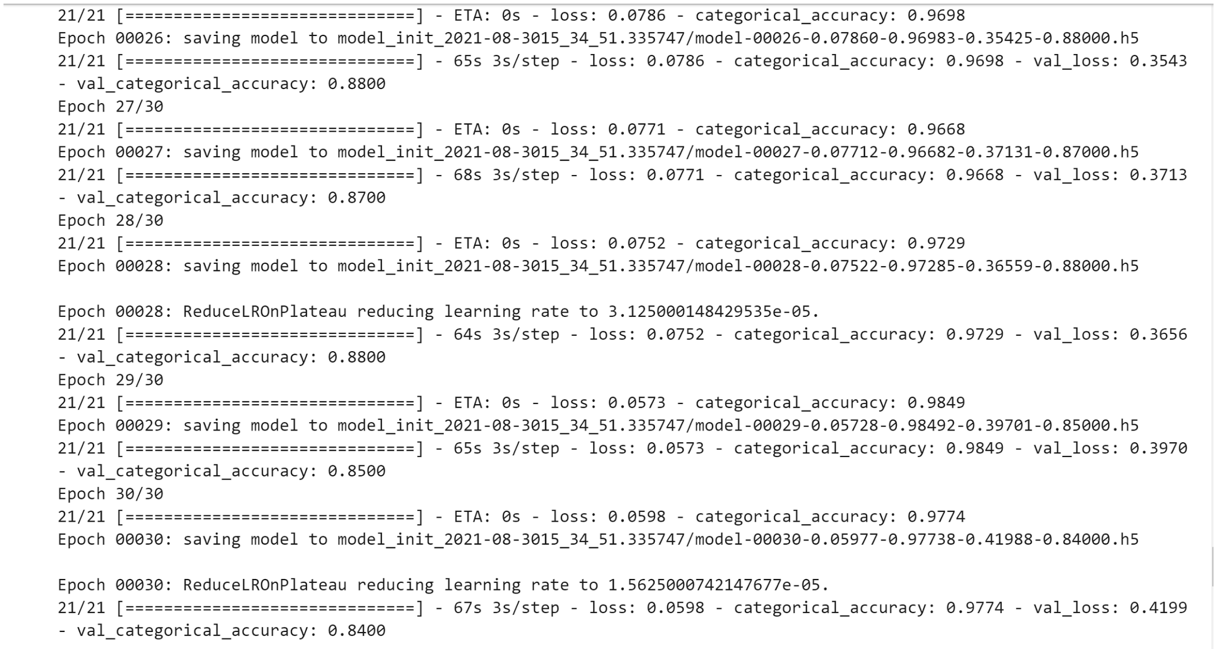
Batch size : 32
Image Size : 90X90
Frames : 15

Number of Layers : 4 Conv+pool; 2 Dense
Dropouts (0.25) : After 2nd and 3rd Conv+pooling.
1 dropout (0.4) and 4th Conv+pooling layer.
1 dropout (0.5) after 1st dense layer

epochs : 30

Model :
Conv3D

Result :
Train accuracy: 97.74%
Val accuracy : 84.00%



```
21/21 [==============================] - ETA: 0s - loss: 0.0786 - categorical_accuracy: 0.9698
Epoch 00026: saving model to model_init_2021-08-3015_34_51.335747/model-00026-0.07860-0.96983-0.35425-0.88000.h5
21/21 [==============================] - 65s 3s/step - loss: 0.0786 - categorical_accuracy: 0.9698 - val_loss: 0.3543
- val_categorical_accuracy: 0.8800
Epoch 27/30
21/21 [==============================] - ETA: 0s - loss: 0.0771 - categorical_accuracy: 0.9668
Epoch 00027: saving model to model_init_2021-08-3015_34_51.335747/model-00027-0.07712-0.96682-0.37131-0.87000.h5
21/21 [==============================] - 68s 3s/step - loss: 0.0771 - categorical_accuracy: 0.9668 - val_loss: 0.3713
- val_categorical_accuracy: 0.8700
Epoch 28/30
21/21 [==============================] - ETA: 0s - loss: 0.0752 - categorical_accuracy: 0.9729
Epoch 00028: saving model to model_init_2021-08-3015_34_51.335747/model-00028-0.07522-0.97285-0.36559-0.88000.h5

Epoch 00028: ReduceLROnPlateau reducing learning rate to 3.125000148429535e-05.
21/21 [==============================] - 64s 3s/step - loss: 0.0752 - categorical_accuracy: 0.9729 - val_loss: 0.3656
- val_categorical_accuracy: 0.8800
Epoch 29/30
21/21 [==============================] - ETA: 0s - loss: 0.0573 - categorical_accuracy: 0.9849
Epoch 00029: saving model to model_init_2021-08-3015_34_51.335747/model-00029-0.05728-0.98492-0.39701-0.85000.h5
21/21 [==============================] - 65s 3s/step - loss: 0.0573 - categorical_accuracy: 0.9849 - val_loss: 0.3970
- val_categorical_accuracy: 0.8500
Epoch 30/30
21/21 [==============================] - ETA: 0s - loss: 0.0598 - categorical_accuracy: 0.9774
Epoch 00030: saving model to model_init_2021-08-3015_34_51.335747/model-00030-0.05977-0.97738-0.41988-0.84000.h5

Epoch 00030: ReduceLROnPlateau reducing learning rate to 1.5625000742147677e-05.
21/21 [==============================] - 67s 3s/step - loss: 0.0598 - categorical_accuracy: 0.9774 - val_loss: 0.4199
- val_categorical_accuracy: 0.8400
```

Decision + Explanation :
Increase 0.25 dropouts to 0.3 as the model is still slightly overfit.

Experiment Number : 8

Details:
Batch size : 32
Image Size : 90X90
Frames : 15

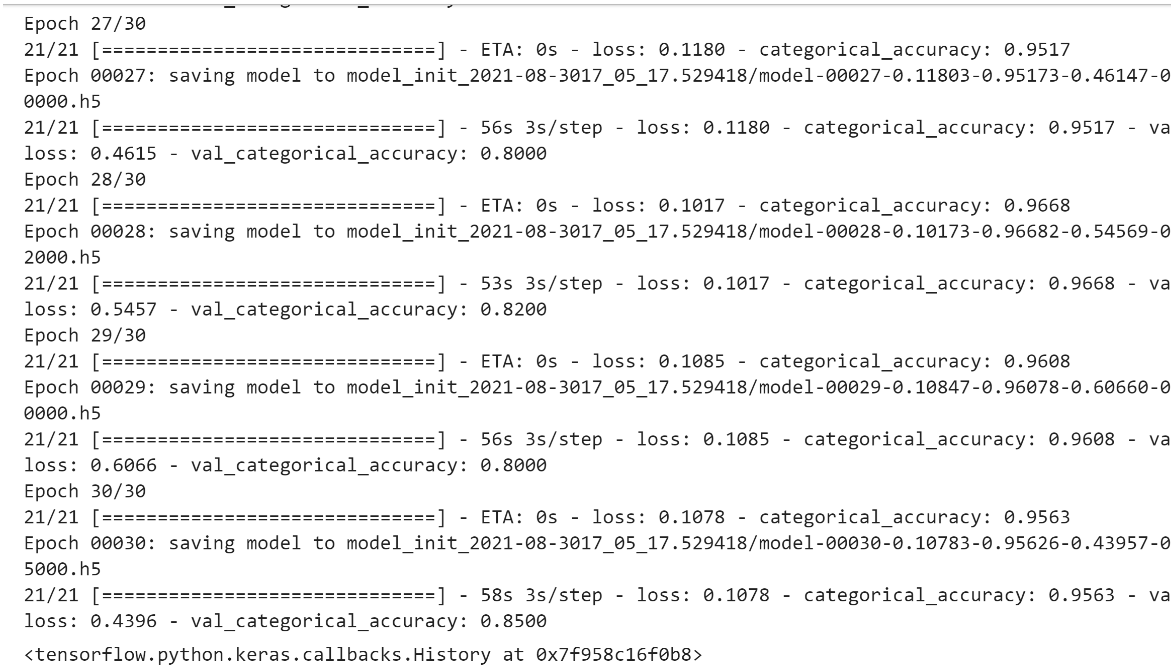
Number of Layers : 4 Conv+pool; 2 Dense

Dropouts (0.3) : after 2nd and 3rd Conv+pooling .
1 dropout (0.4) after the 4th Conv layer.
1 dropout (0.5) after 1st dense layer

epochs : 30

Model :
Conv3D

Result :
Train Accuracy : 95.63%
Val Accuracy : 85.00%



```
Epoch 27/30
21/21 [==============================] - ETA: 0s - loss: 0.1180 - categorical_accuracy: 0.9517
Epoch 00027: saving model to model_init_2021-08-3017_05_17.529418/model-00027-0.11803-0.95173-0.46147-0
0000.h5
21/21 [==============================] - 56s 3s/step - loss: 0.1180 - categorical_accuracy: 0.9517 - va
loss: 0.4615 - val_categorical_accuracy: 0.8000
Epoch 28/30
21/21 [==============================] - ETA: 0s - loss: 0.1017 - categorical_accuracy: 0.9668
Epoch 00028: saving model to model_init_2021-08-3017_05_17.529418/model-00028-0.10173-0.96682-0.54569-0
2000.h5
21/21 [==============================] - 53s 3s/step - loss: 0.1017 - categorical_accuracy: 0.9668 - va
loss: 0.5457 - val_categorical_accuracy: 0.8200
Epoch 29/30
21/21 [==============================] - ETA: 0s - loss: 0.1085 - categorical_accuracy: 0.9608
Epoch 00029: saving model to model_init_2021-08-3017_05_17.529418/model-00029-0.10847-0.96078-0.60660-0
0000.h5
21/21 [==============================] - 56s 3s/step - loss: 0.1085 - categorical_accuracy: 0.9608 - va
loss: 0.6066 - val_categorical_accuracy: 0.8000
Epoch 30/30
21/21 [==============================] - ETA: 0s - loss: 0.1078 - categorical_accuracy: 0.9563
Epoch 00030: saving model to model_init_2021-08-3017_05_17.529418/model-00030-0.10783-0.95626-0.43957-0
5000.h5
21/21 [==============================] - 58s 3s/step - loss: 0.1078 - categorical_accuracy: 0.9563 - va
loss: 0.4396 - val_categorical_accuracy: 0.8500
<tensorflow.python.keras.callbacks.History at 0x7f958c16f0b8>
```

Decision + Explanation :
Add a dropout of 0.3 after the first conv layer also as the model still slightly overfits.

Experiment Number : 9

Details:
Batch size : 32

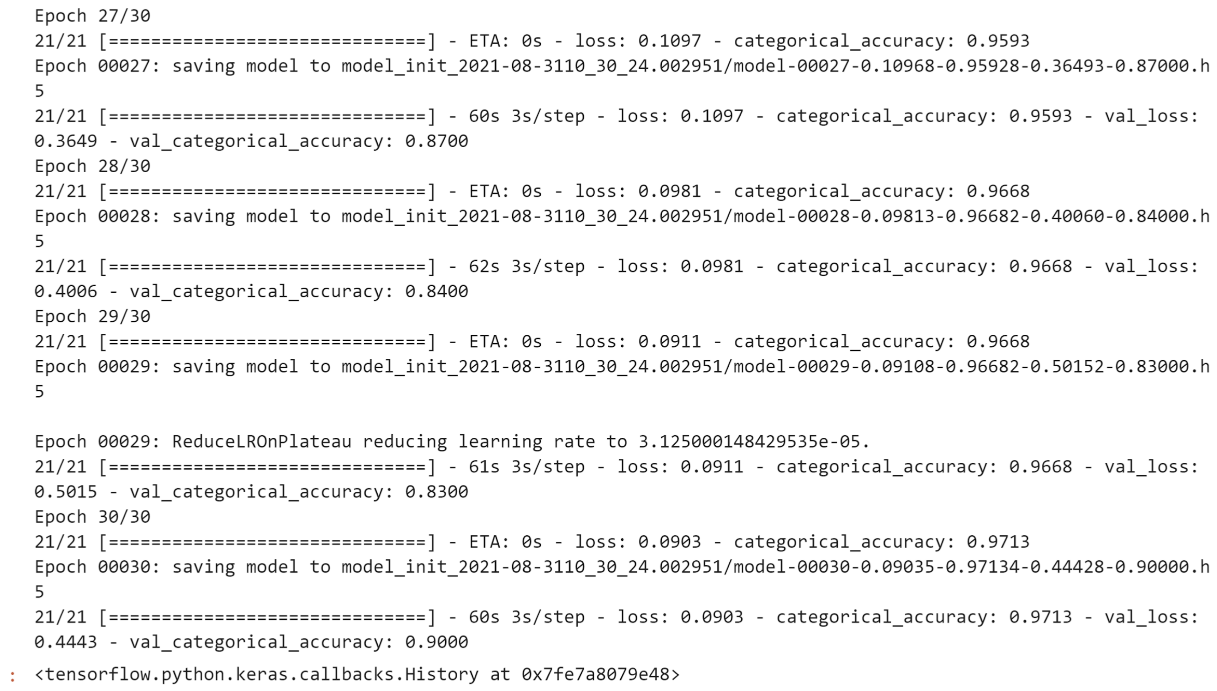
Image Size : 90X90
Frames : 15

Number of Layers : 4 Conv+pool; 2 Dense;
Dropouts (0.3) : after 1s Conv+Poolingt , 2nd Conv+pooling and 3rd Conv+pooling layers .
1 dropout (0.4) after the 4th Conv layer.
1 dropout (0.5) after 1^{st} dense layer

epochs : 30

Model :
Conv3D

Result :
Train Accuracy : 97.13%
Val Accuracy : 90.00%



```
   Epoch 27/30
   21/21 [==============================] - ETA: 0s - loss: 0.1097 - categorical_accuracy: 0.9593
   Epoch 00027: saving model to model_init_2021-08-3110_30_24.002951/model-00027-0.10968-0.95928-0.36493-0.87000.h
   5
   21/21 [==============================] - 60s 3s/step - loss: 0.1097 - categorical_accuracy: 0.9593 - val_loss:
   0.3649 - val_categorical_accuracy: 0.8700
   Epoch 28/30
   21/21 [==============================] - ETA: 0s - loss: 0.0981 - categorical_accuracy: 0.9668
   Epoch 00028: saving model to model_init_2021-08-3110_30_24.002951/model-00028-0.09813-0.96682-0.40060-0.84000.h
   5
   21/21 [==============================] - 62s 3s/step - loss: 0.0981 - categorical_accuracy: 0.9668 - val_loss:
   0.4006 - val_categorical_accuracy: 0.8400
   Epoch 29/30
   21/21 [==============================] - ETA: 0s - loss: 0.0911 - categorical_accuracy: 0.9668
   Epoch 00029: saving model to model_init_2021-08-3110_30_24.002951/model-00029-0.09108-0.96682-0.50152-0.83000.h
   5

   Epoch 00029: ReduceLROnPlateau reducing learning rate to 3.125000148429535e-05.
   21/21 [==============================] - 61s 3s/step - loss: 0.0911 - categorical_accuracy: 0.9668 - val_loss:
   0.5015 - val_categorical_accuracy: 0.8300
   Epoch 30/30
   21/21 [==============================] - ETA: 0s - loss: 0.0903 - categorical_accuracy: 0.9713
   Epoch 00030: saving model to model_init_2021-08-3110_30_24.002951/model-00030-0.09035-0.97134-0.44428-0.90000.h
   5
   21/21 [==============================] - 60s 3s/step - loss: 0.0903 - categorical_accuracy: 0.9713 - val_loss:
   0.4443 - val_categorical_accuracy: 0.9000
:  <tensorflow.python.keras.callbacks.History at 0x7fe7a8079e48>
```

Decision + Explanation :

The gap between training and validation accuracy is now considerably less.

Since we have added adequate and consistent dropouts to the model and have a better model than the previous models, let’s now add some regularization to keep the training accuracy in check so as to avoid any latent overfitting.

Decision:

Added L2 regularization parameter of 0.001 in 2nd, 3rd and 4th convolution layers. We chose L2 regularization over L1 as L2 can be used to learn complex data patterns and computationally efficient.

Reducing the learning rate to 0.0001 just to train the model at a much slower pace with more number of epochs (50).

Experiment Number : 10

Details:
Batch size : 32
Image Size : 90X90
Frames : 15

Number of Layers : 4 Conv+pool; 2 Dense
Dropouts (0.3) : after 1st Conv+pooling, 2nd Conv+pooling and 3rd Conv+pooling layers.
1 dropout (0.4) after the 4th Conv layer.
1 dropout (0.5) after the 1st dense layer.

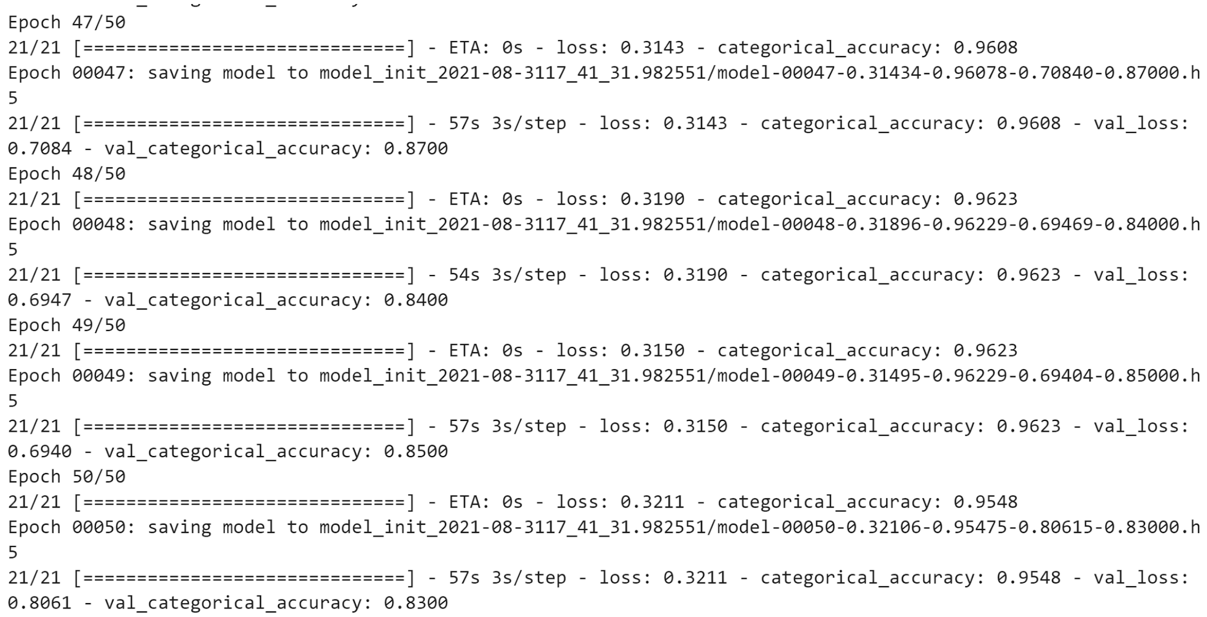
Added L2 regularization parameter of 0.001 in 2nd, 3rd and 4th convolution layers

Learning Rate : 0.0001

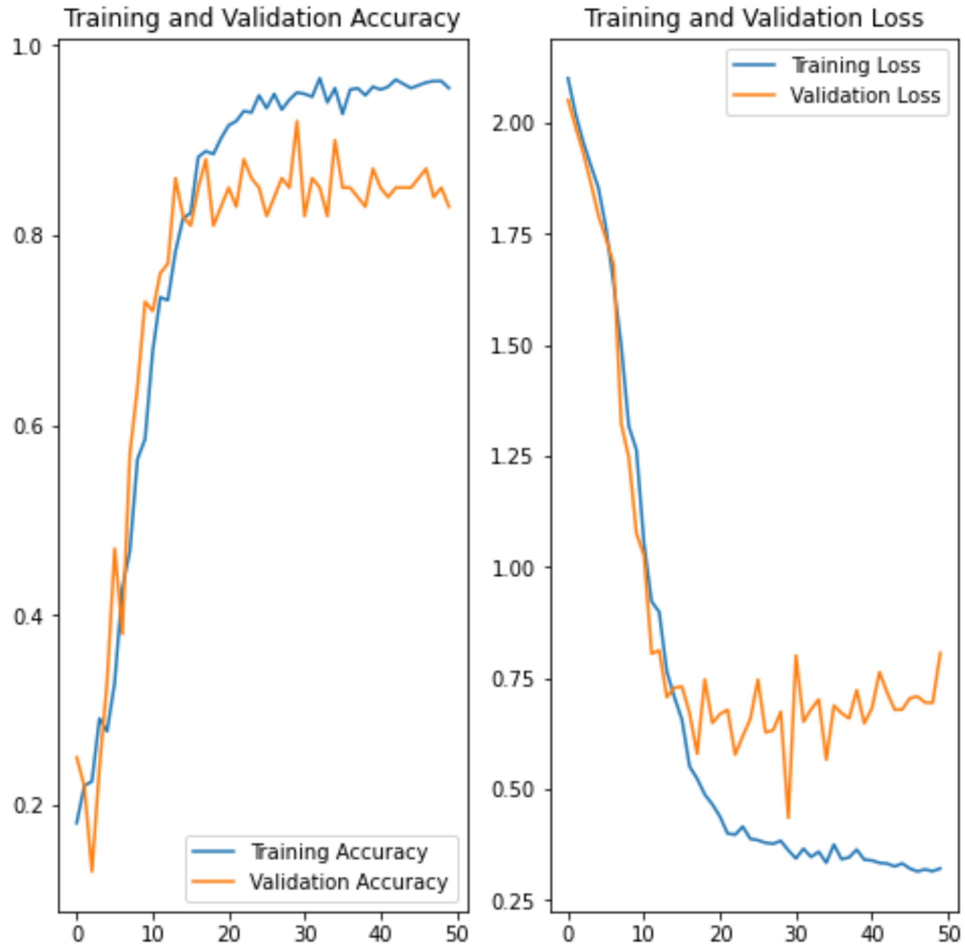
epochs : 50

Model :
Conv3D

Result :
Train Accuracy : 95.13%
Val Accuracy : 83.00%



```
Epoch 47/50
21/21 [==============================] - ETA: 0s - loss: 0.3143 - categorical_accuracy: 0.9608
Epoch 00047: saving model to model_init_2021-08-3117_41_31.982551/model-00047-0.31434-0.96078-0.70840-0.87000.h
5
21/21 [==============================] - 57s 3s/step - loss: 0.3143 - categorical_accuracy: 0.9608 - val_loss:
0.7084 - val_categorical_accuracy: 0.8700
Epoch 48/50
21/21 [==============================] - ETA: 0s - loss: 0.3190 - categorical_accuracy: 0.9623
Epoch 00048: saving model to model_init_2021-08-3117_41_31.982551/model-00048-0.31896-0.96229-0.69469-0.84000.h
5
21/21 [==============================] - 54s 3s/step - loss: 0.3190 - categorical_accuracy: 0.9623 - val_loss:
0.6947 - val_categorical_accuracy: 0.8400
Epoch 49/50
21/21 [==============================] - ETA: 0s - loss: 0.3150 - categorical_accuracy: 0.9623
Epoch 00049: saving model to model_init_2021-08-3117_41_31.982551/model-00049-0.31495-0.96229-0.69404-0.85000.h
5
21/21 [==============================] - 57s 3s/step - loss: 0.3150 - categorical_accuracy: 0.9623 - val_loss:
0.6940 - val_categorical_accuracy: 0.8500
Epoch 50/50
21/21 [==============================] - ETA: 0s - loss: 0.3211 - categorical_accuracy: 0.9548
Epoch 00050: saving model to model_init_2021-08-3117_41_31.982551/model-00050-0.32106-0.95475-0.80615-0.83000.h
5
21/21 [==============================] - 57s 3s/step - loss: 0.3211 - categorical_accuracy: 0.9548 - val_loss:
0.8061 - val_categorical_accuracy: 0.8300
```



Decision + Explanation :

The gap between training and validation accuracy has again increased as the model seems to overfit after 30 epochs.

Also we can slightly increase the regularization term to 0.005 in the 4th Conv+pooling layer and remove the regularization term in the 2nd layer to ignore the features at a later stage rather then in the initial layers.

Decision:

Reduce Epochs to 30.
Regularization : 0.001 in 3rd conv+pooling Layer and 0.005 in 4th Conv+pooling layer.

Experiment Number : 11

Details:
Batch size : 32
Image Size : 90X90
Frames : 15

Number of Layers : 4 Conv+pool; 2 Dense
Dropouts (0.3) : after 1st Conv+pooling, 2nd Conv+pooling and 3rd Conv+pooling layers.
1 dropout (0.4) after the 4th Conv layer.
1 dropout (0.5) after the 1st dense layer.

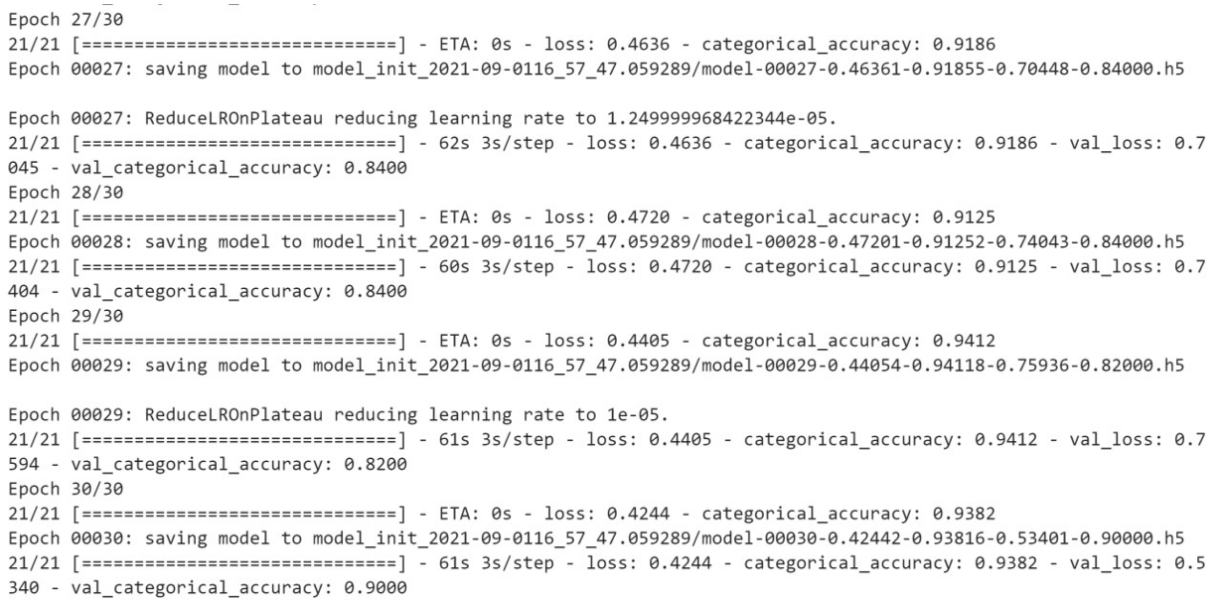
Added L2 regularization parameter of 0.001 in the 3rd Conv+pooling layer
Added L2 regularization parameter of 0.005 in the 4th convolution layers

Learning Rate : 0.0001

epochs : 30

Model :
Conv3D

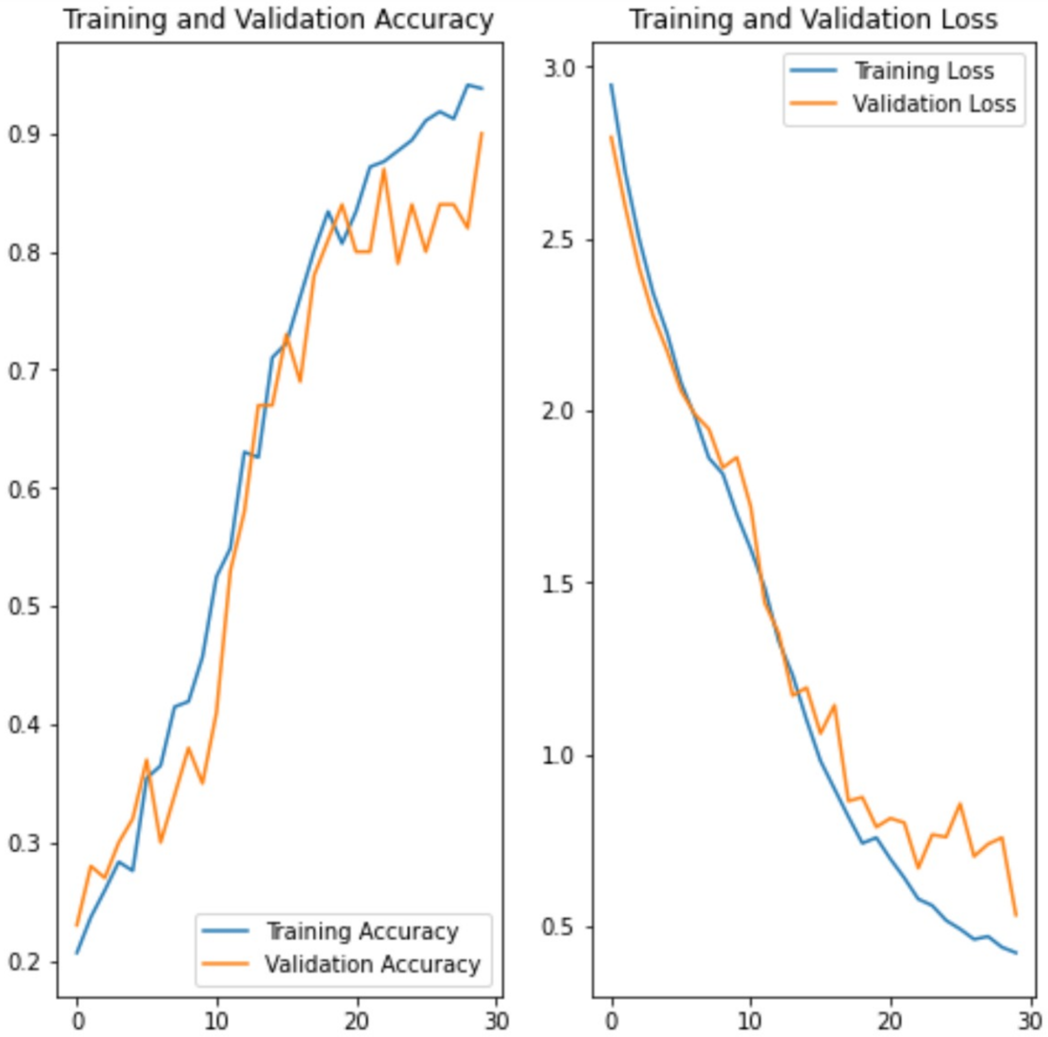
Result :
Train Accuracy : 93.82%
Val Accuracy : 90.00%



```
Epoch 27/30
21/21 [==============================] - ETA: 0s - loss: 0.4636 - categorical_accuracy: 0.9186
Epoch 00027: saving model to model_init_2021-09-0116_57_47.059289/model-00027-0.46361-0.91855-0.70448-0.84000.h5

Epoch 00027: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-05.
21/21 [==============================] - 62s 3s/step - loss: 0.4636 - categorical_accuracy: 0.9186 - val_loss: 0.7
045 - val_categorical_accuracy: 0.8400
Epoch 28/30
21/21 [==============================] - ETA: 0s - loss: 0.4720 - categorical_accuracy: 0.9125
Epoch 00028: saving model to model_init_2021-09-0116_57_47.059289/model-00028-0.47201-0.91252-0.74043-0.84000.h5
21/21 [==============================] - 60s 3s/step - loss: 0.4720 - categorical_accuracy: 0.9125 - val_loss: 0.7
404 - val_categorical_accuracy: 0.8400
Epoch 29/30
21/21 [==============================] - ETA: 0s - loss: 0.4405 - categorical_accuracy: 0.9412
Epoch 00029: saving model to model_init_2021-09-0116_57_47.059289/model-00029-0.44054-0.94118-0.75936-0.82000.h5

Epoch 00029: ReduceLROnPlateau reducing learning rate to 1e-05.
21/21 [==============================] - 61s 3s/step - loss: 0.4405 - categorical_accuracy: 0.9412 - val_loss: 0.7
594 - val_categorical_accuracy: 0.8200
Epoch 30/30
21/21 [==============================] - ETA: 0s - loss: 0.4244 - categorical_accuracy: 0.9382
Epoch 00030: saving model to model_init_2021-09-0116_57_47.059289/model-00030-0.42442-0.93816-0.53401-0.90000.h5
21/21 [==============================] - 61s 3s/step - loss: 0.4244 - categorical_accuracy: 0.9382 - val_loss: 0.5
340 - val_categorical_accuracy: 0.9000
```



Decision + Explanation :

We’ll consider this as our final model as the train and test accuracies are close and promising.
The train loss = 0.4244 and validation loss = 0.5340