Introduction

We have conducted multiple experiments by playing around with the generator, batch size, model architecture, their parameters & hyperparameters.

Below is the tabulation of the list of:

1. models built and experiments conducted
2. performance metrics for each of the models
3. decision behind building each model and, interpretation of their results
4. selecting the final model for complete training & conclusion

Model architecture for each of the model is presented in the code, under its respective summary.

Generator

Generator no.	Image size (width x height)	No. of frames	Cropping (along width & height equally on either sides)	Normalization	Training sequences	Validation sequences	Batch size	Training Batches	Calculation of training data points using batches	Validation Batches	Calculation of validation data points using batches
Generator-1	260 x 240	15	20% & 20%	Min-Max	132	21	10	14	(13 x 10) + (1 x 2)	3	(2 x 10) + (1 x 1)
Generator-2	100 x 100	18	20% & 20%	Min-Max	100	25	25	4	(4 x 25) + (0 x 0)	1	(1 x 25) + (0 x 0)
Generator-3	100 x 100	18	20%, 15% & 20%	Min-Max	133	59	15	9	(8 x 15) + (1 x 13)	4	(3 x 15) + (1 x 14)
Generator-4	120 x 120	18	20% & 20%	Min-Max	332	50	13	26	(26 x 13) + (1 x 7)	4	(3 x 13) + (1 x 11)
Generator-5	120 x 120	18	0% & 0%	Standard	332	50	8	42	(41 x 8) + (1 x 4)	7	(6 x 8) + (1 x 2)
Generator-6	120 x 120	18	0% & 0%	Min-Max	332	50	10	34	(33 x 10) + (1 x 2)	5	(10 x 5) + (0 x 0)
Generator-7	120 x 120	18	0% & 0%	Min-Max	663	100	10	67	(66 x 10) + (1 x 3)	10	(10 x 10) + (0 x 0)

Models & Experiments

Experiment no.	Model	BN	Dropout	TL trainable	Trainable parameters	Non-trainable parameters	Loss function	Optimizer	Learning rate (LR)	LR reduction factor	Number of Epochs
Experiment-1	Conv2D + LSTM	No	No	NA	100,589	0	Categorical CE	Adagrad	0.05	0.75	10
Experiment-2	Conv3D	Yes	50%	NA	3,667,539	240	Categorical CE	Adagrad	0.1	0.25	5
Experiment-3	Conv3D	Yes	25%, 50%	NA	147,829	112	Categorical CE	Adam	0.001	0.5	5
Experiment-4	Conv2D w. InceptionV3 + SimpleRNN	No	50%	No	135,205	21,802,784	Categorical CE	Adam	0.001	0.5	5
Experiment-5	Conv3D	No	No	NA	274,845	0	Categorical CE	Adam	0.001	0.5	5
Experiment-6	Conv2D w. InceptionV3 + GRU	No	50%	No	143,125	21,802,784	Categorical CE	Adam	0.001	0.5	10
Experiment-7	Conv2D w. InceptionV3 + GRU	No	50%	No	143,125	21,802,784	Categorical CE	Adagrad	0.001	0.5	10
Experiment-8	Conv2D w. InceptionV3 + GRU	Yes	50%	Yes	22,559,333	34,496	Categorical CE	Adagrad	0.002	0.5	20
Experiment-9	Conv3D	Yes	50%	NA	995,701	112	Categorical CE	Adam	0.001	0.5	5
Experiment-10	Conv2D w. VGG16 + GRU	No	50%	No	306,965	14,714,688	Categorical CE	Adam	0.001	0.5	5
Experiment-11	Conv2D w. InceptionV3 + GRU	No	50%	No	143,125	21,802,784	Categorical CE	Adagrad	0.001	0.5	15
Experiment-12	Conv2D w. InceptionV3 + GRU	Yes	40%, 25%	Yes	22,559,333	34,496	Categorical CE	Adagrad	0.1	0.5	10
Experiment-13	Conv2D w. InceptionV3 + GRU	Yes	40%, 25%	Yes	22,559,333	34,496	Categorical CE	Adagrad	0.05	0.75	10
Final Experiments											
Experiment-1	Conv2D w. InceptionV3 + SimpleRNN	Yes	40%, 25%	Yes	22,032,837	34,496	Categorical CE	Adagrad	0.05	0.75	10
Experiment-2	Conv2D w. InceptionV3 + GRU	Yes	40%, 25%	Yes	22,559,333	34,496	Categorical CE	Adagrad	0.05	0.75	10
Experiment-3	Conv2D w. InceptionV3 + LSTM	Yes	40%, 25%	Yes	22,822,437	34,496	Categorical CE	Adagrad	0.05	0.75	10
Experiment-4	Conv3D	No	No	NA	141,901	0	Categorical CE	Adagrad	0.05	0.75	10
Experiment-5	Conv3D	Yes	50%	NA	141,965	64	Categorical CE	Adagrad	0.05	0.75	10
Final Model											
Final	Conv2D w. InceptionV3 + LSTM	Yes	40%, 25%	Yes	22,822,437	34,496	Categorical CE	Adagrad	0.001	0.75	30

Performance Metrics

Experiment no.	Generator no.	Training Accuracy	Validation Accuracy	Training Log Loss	Validation Log Loss	Training Time (s)
Experiment-1	Generator-1	20.71%	33.33%	1.7062	2.1849	3982
Experiment-2	Generator-2	27.50%	20.00%	1.5414	1.5914	1091
Experiment-3	Generator-2	17%	20%	1.6145	1.6124	865
Experiment-4	Generator-3	22.76%	30.91%	1.6545	1.5965	3437
Experiment-5	Generator-3	30.95%	35.71%	1.6372	1.5171	2503
Experiment-6	Generator-4	70.59%	7.14%	0.7368	0.7059	691
Experiment-7	Generator-5	38.10%	36.54%	1.4467	1.5798	995
Experiment-8	Generator-5	68.22%	47.12%	0.9221	1.2054	2106
Experiment-9	Generator-5	23.64%	23.08%	1.6456	1.7369	504
Experiment-10	Generator-5	25.00%	24.00%	1.5838	1.6031	2350
Experiment-11	Generator-6	52.84%	48.00%	1.2678	1.4813	1478
Experiment-12	Generator-6	75.92%	65.00%	0.6839	1.1634	1161
Experiment-13	Generator-6	67.61%	62.00%	0.8858	1.272	392
Final Experiments						
Experiment-1	Generator-6	22.65%	32.00%	1.6374	1.6966	2819
Experiment-2	Generator-6	22.65%	36.00%	1.6350	1.5219	392
Experiment-3	Generator-6	33.53%	24.00%	1.4910	1.4663	390
Experiment-4	Generator-6	55.00%	50.00%	1.2004	1.2658	363
Experiment-5	Generator-6	27.94%	32.00%	1.4794	1.4623	383
Final Model						
Final	Generator-7	98.06%	93.00%	0.1465	0.2569	2237

***all performance metrics are recorded at the end of last epoch*

Interpretation & Decision

Experiment no.	Generator no.	Interpretation & Decision
Experiment-1	Generator-1	Used same percentage of cropping for both square & rectangular images. Used min-max normalization of image pixels to boost model performance and, training time. Also, there are no significant outliers, as our images are not noisy. Used Conv2D in tandem with LSTM, as it is the most complex RNN to have good accuracy. Validation accuracy is more than training accuracy, with validation loss > 2 and training loss > 1. Training time too high.
Experiment-2	Generator-2	Used Conv3D to process 3D images (Videos), due to suitability. Redcued number of epochs, as CNN is a better image processor. Reduced image size, increased frame count slightly (comp.), used batch normalization (BN) and dropouts, increased learning rate, reduced training data points, increased batch size (also causes more GPU usage) to reduce training time. Reduced learning rate reduction factor to compensate for the increased learning rate.Training accuracy improced & validation accuracy dropped. Both training & validation loss still > 1. Training time reduced significantly. Model underfitting.
Experiment-3	Generator-2	Used Conv3D. Reduced trainable parameters, added more dropouts, reduced learning rate, increased reduction factor (slightly) to make up for the parameters & dropouts. Switched from Adagrad to Adam optimizer, as latter is more suitable for dense (non-sparse) features than former. Training accuracy alone dropped, but lower than validation accuracy, due to many dropouts & BN. Both training & validation loss still > 1. Training time reduced.
Experiment-4	Generator-3	Used Conv2D with transfer learning (TL) for image processing, with pre-trained weights, in tandem with Vanilla RNN for faster processing and better results. Used Inception V3 architecture for transfer learning, as it is recommended by Keras for image processing. Used cropping proportionally for square & rectangular images. Increased training & validation data points, reduced batch size, reduced dropouts, removed BN to improve accuracy. Validation accuracy more than training accuracy, both poor till now, with their losses still > 1. Training time increased tremendously. Model still underfitting.
Experiment-5	Generator-3	Used Conv3D without BN & dropouts to improve accuracy. Both training & validation accuracy increased, both doing okay. However, validation accuracy is still more than that of training, with both their losses still > 1. Training time reduced significantly. Model still underfitting.
Experiment-6	Generator-4	Increased image size. Used same percentage of cropping for both square & rectangular images. Used similar model architecture to "Experiment-4", using GRU in place of Vanilla RNN, with 10 epochs instead of 5. Increased training & validation data points to 50% and, lowered the batch size (slightly) and, increased the epochs to 10 to have less model generalization and, better accuracy. Added 50% dropouts for faster training, without hampering the transfer learning. Training accuracy increased a great deal but, validation accuracy is very poor. Might be a generator issue. However, both training & validation loss < 1 also, training time is quite low. Still not considering this, as generator needs a fix. Model suddenly cannot overfit this much.
Experiment-7	Generator-5	Regarding generator issue, there was some error fetching the reamining data points after exhausting all mini-batches. No cropping done for both square & rectangular images, since some information getting lost. Used only intermediate frames in all experiments. Switched from min-max normalization to standardization of images, to check if any noise significant enough to affect the accuracy. Lowered the batch size little bit, with the same training & validation data points, for further reducing the model generalization. Same model used as in "Experiment-6". GRU with more epochs than initial CNN+RNN experiment means more learning and more accuracy. Switched from Adam to Adagrad optimizer to consider sparse features, to compensate for noise filtering using standardization. Moderate training & validation accuracy, both losses still > 1, training time quite low. Model still underfitting.
Experiment-8	Generator-5	Same model used as in "Experiment-6" however, increased epochs to 20 and, setting trainable parameters of transfer learning to true i.e., pre-trained weights can now be trained. Also, slightly increasing the learning rate and introducing batch normalization for faster convergence trading off with improved accuracy by including many trainable weights and more epochs. Both training & validation accuracy improved significantly, with training loss < 1 for the first time but, validation loss still > 1, training time increased significantly. However, model began to slightly overfit.

Experiment-9	Generator-5	Using Conv3D with BN & 50% dropouts and, Adam optimizer with slightly smaller learning rate & same reduction factor, to check if CNN performs any better without considering the sparse features on fine images. Added BN and reduced the epochs to 5 for reducing the training time, dropouts kept same. Model is now neither overfitting nor underfitting, but has poor accuracy in both sets, with both losses still > 1. Training time reduced significantly.
Experiment-10	Generator-5	Using Conv2D with VGG16 for transfer learning in tandem with GRU, without BN, keeping the dropouts same. Used VGG16 (non-trainable weights) to see if it helps improving accuracy without overfitting/underfitting the model. Since, VGG16 is one of the standard models available for training. Did not go for other higher standard models like ResNet due to lack of computational resources. Absolutely no improvement in model performance, with tremendous increase in training time.
Experiment-11	Generator-6	Switched back to min-max normalization, same batch size as initial experiment, adagrad optimizer, as the model performance is not boosted by more generalizability, noise filtering, treating the feature as sparse. Only leads to more training time. Also, replaced VGG16 back with InceptionV3 in previous model for the same reason. However, made the learning rate small, with same reduction factor. Both training & validation accuracy are fairly descent, without the model underfitting/overfitting. However, both their losses still > 1. Significant reduction in training time even after increasing the epochs.
Experiment-12	Generator-6	Set the InceptionV3 trainable parameters to true in the previous model. Introduced BN, added more dropouts, increased learning rate, with same reduction factor, reduced epochs for performance-training time trade-off. Both training & validation accuracy are good for limited data points. Training loss < 1 and, validation loss is close to 1 for the first time i.e., least so far, with even lower training time.
Experiment-13	Generator-6	Reduced learning rate and increased reduction factor slightly (compensation for learning rate) in previous model for even faster convergence. Model performance almost identical, with a slight dip compared to "Experiment-12". We consider model architecture in "Experiment-13", for final experiments, due to the lower gap between training & validation accuracy during the final epochs.

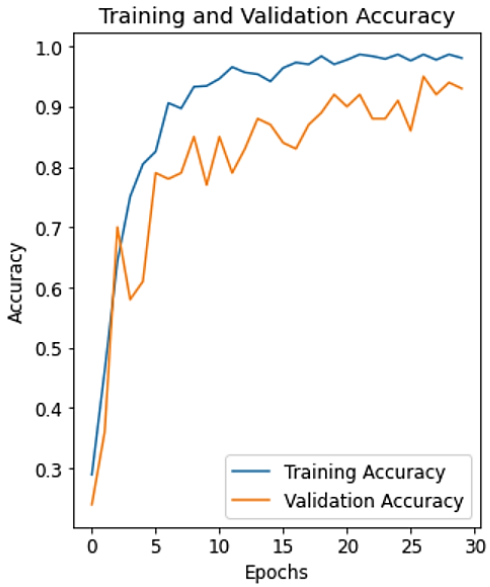
*Conclusion has been briefly discussed here along with other explanation

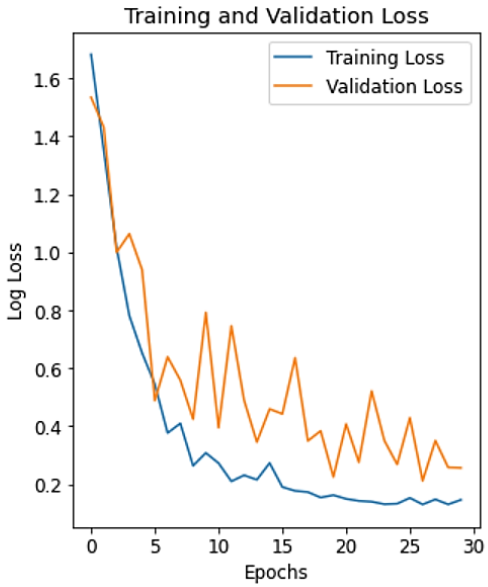
Final Experiments		
Experiment-1	Generator-6	Used the same model architecture, its parameters and, generator as in "Experiment-13" (finalized one) for "Final Experiments-1,2,3". Compared model performance between Vanilla RNN, GRU and LSTM, by keeping other parameters intact. Training accuracy constant in Vanilla RNN & GRU, with an increase in LSTM. Whereas, validation accuracy increasing from Vanilla RNN to GRU later, decreasing in LSTM (being the lowerst). Both training & validation loss > 1. However, training time for GRU & LSTM was much lower compared to Vanilla RNN. Model seems to underfit in all scenarios. Exactly same experiment was conducted in "Experiment-13" and "Final Experiment-2" but, the results was completely off. Also, Vanilla RNN is supposed to have the least training time, being the least complex net of the bunch. This can be explained by NN taking the maximum time to learn from the initial model (any chosen) then, only learn what is left out from the subsequent models. Accuracy gap between training & validation is somewhat similar for all Conv2D + RNN models however, validation accuracy is higher than its training accuracy in Conv2D with Vanilla RNN and GRU unlike the one with LSTM.
Experiment-2	Generator-6	
Experiment-3	Generator-6	

Experiment-4	Generator-6	Used two Conv3D NN's – with & without hyperparameters (BN & dropouts), keeping all other parameters from "Experiment-13" intact. Model without hyperparameters seems to have the best performance compared to any other models built during the final experiment and, the one with hyperparameters being the lowest. Moderate increase in the Conv3D model without hyperparameter tuning can be explained by the model applying based on what it has already learnt from the previous models. Model performance plummeting after this can be due to bad data points (important features missing), but unable to provide a clear explanation. We still consider the model architecture in "Final Experiment-3", to train our final model, as LSTM can handle the temporal (sequenced) data the best. Also, disagreement in accuracies can be explained due to very high variance while conducting multiple repeated experiments before as compared during the final ones. Accuracy gap between training & validation is nearly the same for both Conv3D however, validation accuracy is higher than its training accuracy in Conv3D without any hyperparameters unlike the one with hyperparameters.
Experiment-5	Generator-6	

Final Model		
Final	Generator-7	Used Conv2D with InceptionV3 transfer learning, with trainable weights, in tandem with LSTM for our final model, fitting all data points, with 30 epochs. Model & other parameters (including generator) same as "Experiment-13" but, with LSTM in place of GRU or, exactly same as "Final Experiment-3", with learning rate of 0.001 and lower-bound of 0.00001 for faster convergence. Note: "Generator-7" is exactly the same as "Generator-6" with all data points. **Final result:** Model accuracy is excellent, with **training accuracy** at **98%** and **validation accuracy** at **93%**. Also, having low to bare minimum gap between training & validation epochs at different epochs, during training. Same applies for the loss as well. Both training & validation loss is well < 1. Training time is acceptable for a model having good performance.

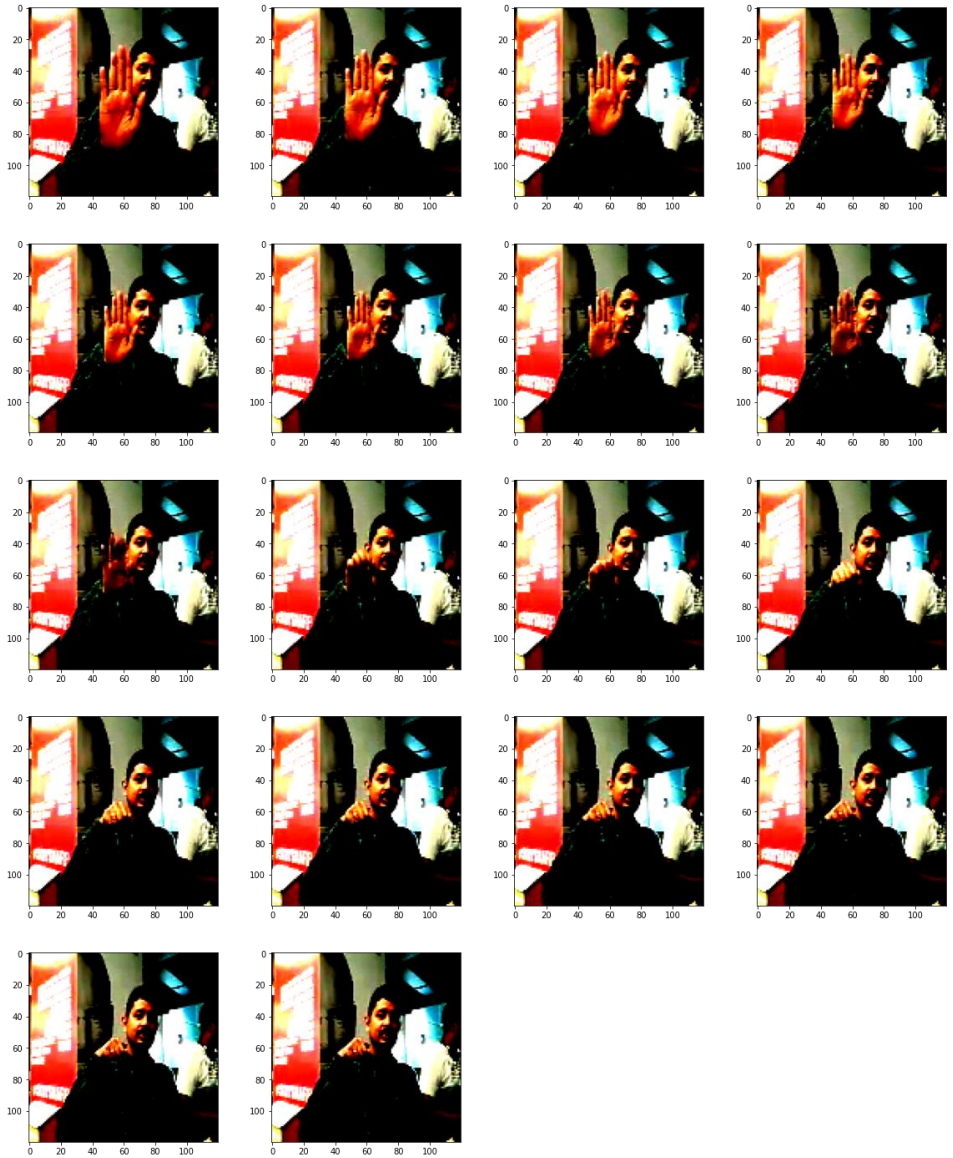
Performance Metrics for Final Model:



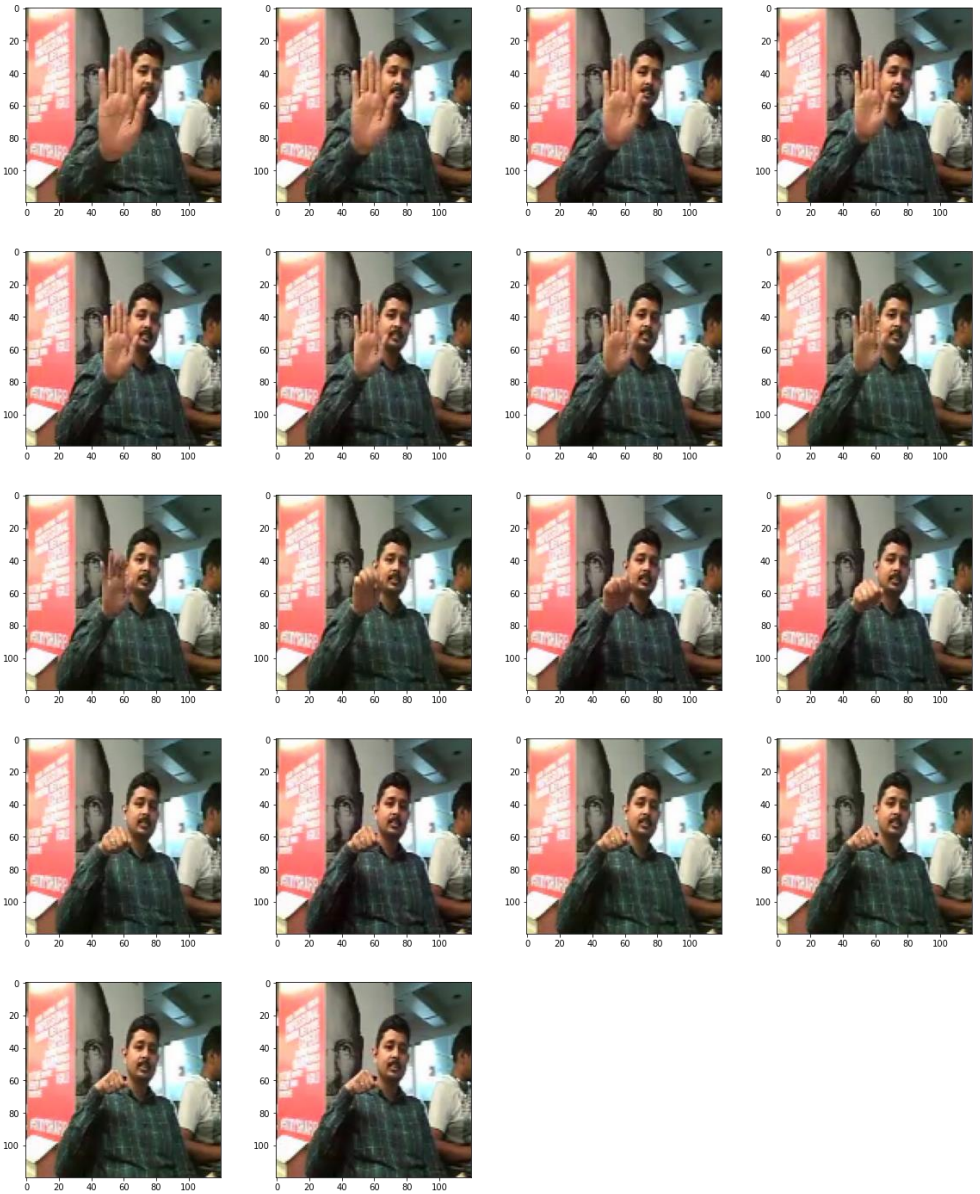


Appendix: Refer to the appendix section (after this) for insights on the model training & architectures for all the initial experiments conducted, along with the change in its learning rates. This section also contains a sample video in its frames with both min-max normalization & standardization and, how different generator modifications effect training.

Frames of a video after standardization of its pixel values



Frames of a video after Min-max Normalization of its pixel values



Generator

Generator-1, Generator-2, Generator-4

```
img_idx = [frames for frames in range(5,frames+5)] #list of images t
o be selected in sequence for each video, maximum information captur
ed during the intermediate frames

num_batches = math.floor(num_train_sequences/batch_size) #to calcula
te the number of batches

rem_num_batches = num_train_sequences - num_batches #to calculate th
e remaining number of batches

if image.shape[0] != image.shape[1]:
                      image = image[16:128,12:96] #crop images for 1
60 x 120
                    else:
                      image = image[36:288,36:288] #crop images for
360 x 360


                    image = imresize(image,(y,z)) #resize images

                    batch_data[folder,idx,:,:,0] = (image[:,:,0] - n
p.min(image[:,:,0]))/(np.max(image[:,:,0])- np.min(image[:,:,0])) #n
ormalise and feed in the image
                    batch_data[folder,idx,:,:,1] = (image[:,:,1] - n
p.min(image[:,:,1]))/(np.max(image[:,:,1])- np.min(image[:,:,1])) #n
ormalise and feed in the image
                    batch_data[folder,idx,:,:,2] = (image[:,:,2] - n
p.min(image[:,:,2]))/(np.max(image[:,:,2])- np.min(image[:,:,2])) #n
ormalise and feed in the image
```

Generator-3

```
img_idx = [frames for frames in range(5,frames+5)] #list of images t
o be selected in sequence for each video, maximum information captur
ed during the intermediate frames

num_batches = math.floor(len(t)/batch_size) #to calculate the number
of batches
        for batch in range(num_batches): # we iterate over the numbe
r of batches
            batch_data = np.zeros((batch_size,frames,width,height,ch
annels)) # frames is the number of images you use for each video, (w
idth,height) is the final size of the input images and 3 is the numb
er of channels RGB
            batch_labels = np.zeros((batch_size,5)) # batch_labels i
s the one hot representation of the output

rem_data_points = len(t) - num_batches*batch_size #to calculate the
reamining data points
        if (rem_data_points != 0): #checking if any data points rema
ining
          batch_data = np.zeros((rem_data_points,frames,width,height
,channels)) # frames is the number of images you use for each video,
(width,height) is the final size of the input images and 3 is the nu
mber of channels RGB
          batch_labels = np.zeros((rem_data_points,5)) #
batch_labels is the one hot representation of the output

if image.shape[0] != image.shape[1]:
                      image = image[16:128,9:102] #cropping 20%
along width & 15% along height equally on either sides for 160 x 120
images
                    else:
                      image = image[36:288,36:288] #cropping 20%
along width & height equally on either sides for 360 x 360 images
                    image = imresize(image,(width,height)) #resize
images


                    batch_data[folder,idx,:,:,channels-3] =
(image[:,:,channels-3] - np.min(image[:,:,channels-
3]))/(np.max(image[:,:,channels-3])- np.min(image[:,:,channels-3]))
#normalise using Min-Max, and feed in the image
                    batch_data[folder,idx,:,:,channels-2] =
(image[:,:,channels-2] - np.min(image[:,:,channels-
2]))/(np.max(image[:,:,channels-2])- np.min(image[:,:,channels-2]))
#normalise using Min-Max, and feed in the image
                    batch_data[folder,idx,:,:,channels-1] =
(image[:,:,channels-1] - np.min(image[:,:,channels-
1]))/(np.max(image[:,:,channels-1])- np.min(image[:,:,channels-1]))
#normalise using Min-Max, and feed in the image

imgs = os.listdir(source_path+'/'+ t[folder].split(';')[0]) # read
all the images in the folder
image = imread(source_path+'/'+
t[folder].strip().split(';')[0]+'/'+imgs[item]).astype(np.float32)
```

Generator-5

```
img_idx = [frames for frames in range(5,frames+5)] #list of images t
o be selected in sequence for each video, maximum information captur
ed during the intermediate frames

num_batches = math.floor(sequences/batch_size) #to calculate the num
ber of batches
       for batch in range(num_batches): # we iterate over the numbe
r of batches
           batch_data = np.zeros((batch_size,frames,width,height,ch
annels)) # frames is the number of images you use for each video, (w
idth,height) is the final size of the input images and 3 is the numb
er of channels RGB
           batch_labels = np.zeros((batch_size,5)) # batch_labels i
s the one hot representation of the output

if (sequences % batch_size != 0): #checking if any batches remaining
          rem = sequences - sequences % batch_size #to calculate the
remaining number of batches
          seq = sequences
          batch_size = rem
          for folder in range(rem,seq): # we iterate over the remain
ing number of batches
```

```
            batch_data = np.zeros((batch_size,frames,width,height,ch
annels)) # frames is the number of images you use for each video, (w
idth,height) is the final size of the input images and 3 is the numb
er of channels RGB
            batch_labels = np.zeros((batch_size,5)) # batch_labels i
s the one hot representation of the output

                    Mu = image.mean()
                    Std = image.std()

                    batch_data[folder,idx,:,:,0] = (image[:,:,0] - M
u)/Std  #Normalize
                    batch_data[folder,idx,:,:,1] = (image[:,:,1] - M
u)/Std  #normalise
                    batch_data[folder,idx,:,:,2] = (image[:,:,2] -
Mu)/Std  #normalise

imgs = os.listdir(source_path+'/'+ t[folder].split(';')[0]) # read
all the images in the folder
image = imread(source_path+'/'+
t[folder].strip().split(';')[0]+'/'+imgs[item]).astype(np.float32)
```

Generator-6

```
img_idx = [frames for frames in range(5,frames+5)] #list of images t
o be selected in sequence for each video, maximum information captur
ed during the intermediate frames

num batches = math.floor(sequences/batch size) #to calculate the num
ber of batches

        for batch in range(num_batches): # we iterate over the numbe
r of batches
            batch_data = np.zeros((batch_size,frames,width,height,ch
annels)) # frames is the number of images you use for each video, (w
idth,height) is the final size of the input images and 3 is the numb
er of channels RGB
            batch_labels = np.zeros((batch_size,5)) # batch_labels i
s the one hot representation of the output

for folder_oth in range(batch_size): # iterate over the batch_size
            imgs = os.listdir(source_path+'/'+ t[(folder_oth + (num_
batches*batch_size))%(len(folder_list))].split(';')[0]) # read all t
he images in the folder
            imgs.sort(reverse=True)

            for idx,item in enumerate(img_idx): #  Iterate over the
frames/images of a folder to read them in
                    image = imread(source_path+'/'+ t[(folder_oth +
(num_batches*batch_size))%(len(folder_list))].strip().split(';')[0]+
'/'+imgs[item]).astype(np.float32)
                    image = imresize(image, (width, height)) #resize
d images, cropping not carried out since, affects accuracy

                    batch_data[folder,idx,:,:,0] = (image[:,:,0] - n
p.min(image[:,:,0]))/(np.max(image[:,:,0])- np.min(image[:,:,0])) #n
ormalise and feed in the image
                    batch_data[folder,idx,:,:,1] = (image[:,:,1] - n
p.min(image[:,:,1]))/(np.max(image[:,:,1])- np.min(image[:,:,1])) #n
ormalise and feed in the image
                    batch_data[folder,idx,:,:,2] = (image[:,:,2] -
np.min(image[:,:,2]))/(np.max(image[:,:,2])- np.min(image[:,:,2]))
#normalise and feed in the image
```

Generator-7

Generator-6 with full dataset, instead of 50%.

Experiments

Experiment-1 Generator-1

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed_146 (TimeD  (None, 18, 120, 120, 8)  1184
 istributed)

 time_distributed_147 (TimeD  (None, 18, 59, 59, 8)    584
 istributed)

 time_distributed_148 (TimeD  (None, 18, 29, 29, 8)    0
 istributed)

 time_distributed_149 (TimeD  (None, 18, 29, 29, 16)   1168
 istributed)

 time_distributed_150 (TimeD  (None, 18, 14, 14, 16)   2320
 istributed)

 time_distributed_151 (TimeD  (None, 18, 7, 7, 16)     0
 istributed)

 time_distributed_152 (TimeD  (None, 18, 7, 7, 32)     4640
 istributed)

 time_distributed_153 (TimeD  (None, 18, 3, 3, 32)     0
 istributed)

 time_distributed_154 (TimeD  (None, 18, 288)          0
 istributed)

 lstm_14 (LSTM)              (None, 64)                90368

 dense_14 (Dense)            (None, 5)                 325

=================================================================
```

Model Training:

```
Epoch 1/10
14/14 [==============================] - ETA: 0s - loss: 1.7105 - ca
tegorical_accuracy: 0.2000 Source path =  /content/drive/MyDrive/Pro
ject_data/Project_data/val ; batch size = 10

Epoch 1: saving model to model_init_2022-05-0300_45_32.533735/model-
00001-1.71046-0.20000-2.09339-0.16667.h5
14/14 [==============================] - 985s 75s/step - loss: 1.710
5 - categorical_accuracy: 0.2000 - val_loss: 2.0934 - val_categorica
l_accuracy: 0.1667 - lr: 0.0500
Epoch 2/10
14/14 [==============================] - ETA: 0s - loss: 1.7119 - ca
tegorical_accuracy: 0.2071
Epoch 2: saving model to model_init_2022-05-0300_45_32.533735/model-
00002-1.71191-0.20714-2.34158-0.30000.h5
14/14 [==============================] - 721s 55s/step - loss: 1.711
9 - categorical_accuracy: 0.2071 - val_loss: 2.3416 - val_categorica
l_accuracy: 0.3000 - lr: 0.0500
Epoch 3/10
14/14 [==============================] - ETA: 0s - loss: 1.7346 - ca
tegorical_accuracy: 0.2429
Epoch 3: saving model to model_init_2022-05-0300_45_32.533735/model-
00003-1.73456-0.24286-2.11246-0.16667.h5

Epoch 3: ReduceLROnPlateau reducing learning rate to 0.0375000005587
93545.
14/14 [==============================] - 567s 44s/step - loss: 1.734
6 - categorical_accuracy: 0.2429 - val_loss: 2.1125 - val_categorica
l_accuracy: 0.1667 - lr: 0.0500
Epoch 4/10
14/14 [==============================] - ETA: 0s - loss: 1.6906 - ca
tegorical_accuracy: 0.1786
Epoch 4: saving model to model_init_2022-05-0300_45_32.533735/model-
00004-1.69060-0.17857-2.01779-0.20000.h5
14/14 [==============================] - 433s 33s/step - loss: 1.690
6 - categorical_accuracy: 0.1786 - val_loss: 2.0178 - val_categorica
l_accuracy: 0.2000 - lr: 0.0375
Epoch 5/10
14/14 [==============================] - ETA: 0s - loss: 1.7005 - ca
tegorical_accuracy: 0.2500
Epoch 5: saving model to model_init_2022-05-0300_45_32.533735/model-
00005-1.70047-0.25000-2.14493-0.40000.h5
14/14 [==============================] - 336s 26s/step - loss: 1.700
5 - categorical_accuracy: 0.2500 - val_loss: 2.1449 - val_categorica
l_accuracy: 0.4000 - lr: 0.0375
Epoch 6/10
14/14 [==============================] - ETA: 0s - loss: 1.6904 - ca
tegorical_accuracy: 0.2929
Epoch 6: saving model to model_init_2022-05-0300_45_32.533735/model-
00006-1.69035-0.29286-2.19389-0.33333.h5

Epoch 6: ReduceLROnPlateau reducing learning rate to 0.0281250011175
8709.
14/14 [==============================] - 271s 21s/step - loss: 1.690
4 - categorical_accuracy: 0.2929 - val_loss: 2.1939 - val_categorica
l_accuracy: 0.3333 - lr: 0.0375
Epoch 7/10
14/14 [==============================] - ETA: 0s - loss: 1.7450 - ca
tegorical_accuracy: 0.1929
Epoch 7: saving model to model_init_2022-05-0300_45_32.533735/model-
00007-1.74499-0.19286-2.09363-0.30000.h5
14/14 [==============================] - 208s 16s/step - loss: 1.745
0 - categorical_accuracy: 0.1929 - val_loss: 2.0936 - val_categorica
l_accuracy: 0.3000 - lr: 0.0281
Epoch 8/10
14/14 [==============================] - ETA: 0s - loss: 1.6960 - ca
tegorical_accuracy: 0.2357
Epoch 8: saving model to model_init_2022-05-0300_45_32.533735/model-
00008-1.69604-0.23571-1.92595-0.36667.h5
```

```
14/14 [==============================] - 157s 12s/step - loss: 1.696
0 - categorical_accuracy: 0.2357 - val_loss: 1.9259 - val_categorica
l_accuracy: 0.3667 - lr: 0.0281
Epoch 9/10
14/14 [==============================] - ETA: 0s - loss: 1.7244 - ca
tegorical_accuracy: 0.1714
Epoch 9: saving model to model_init_2022-05-0300_45_32.533735/model-
00009-1.72442-0.17143-2.17803-0.16667.h5
14/14 [==============================] - 98s 8s/step - loss: 1.7244
- categorical_accuracy: 0.1714 - val_loss: 2.1780 - val_categorical_
accuracy: 0.1667 - lr: 0.0281
Epoch 10/10
14/14 [==============================] - ETA: 0s - loss: 1.7062 - ca
tegorical_accuracy: 0.2071
Epoch 10: saving model to model_init_2022-05-0300_45_32.533735/model
-00010-1.70622-0.20714-2.18488-0.33333.h5

Epoch 10: ReduceLROnPlateau reducing learning rate to 0.021093750838
190317.
14/14 [==============================] - 118s 9s/step - loss: 1.7062
- categorical_accuracy: 0.2071 - val_loss: 2.1849 - val_categorical_
accuracy: 0.3333 - lr: 0.0281
Execution time for this model is 3981.8887910842896 seconds
```

Experiment-2 Generator-2

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv3d (Conv3D)             (None, 18, 100, 100, 8)   656

 activation (Activation)     (None, 18, 100, 100, 8)   0

 conv3d_1 (Conv3D)           (None, 18, 100, 100, 8)   1736

 activation_1 (Activation)   (None, 18, 100, 100, 8)   0

 max_pooling3d (MaxPooling3D  (None, 9, 50, 50, 8)     0
 )

 conv3d_2 (Conv3D)           (None, 9, 50, 50, 16)     3472

 activation_2 (Activation)   (None, 9, 50, 50, 16)     0

 conv3d_3 (Conv3D)           (None, 7, 48, 48, 16)     6928

 activation_3 (Activation)   (None, 7, 48, 48, 16)     0

 max_pooling3d_1 (MaxPooling  (None, 4, 24, 24, 16)    0
 3D)

 conv3d_4 (Conv3D)           (None, 2, 22, 22, 32)     13856

 activation_4 (Activation)   (None, 2, 22, 22, 32)     0

 batch_normalization (BatchN  (None, 2, 22, 22, 32)    128
 ormalization)

 max_pooling3d_2 (MaxPooling  (None, 1, 11, 11, 32)    0
 3D)

 flatten (Flatten)           (None, 3872)              0

 dense (Dense)               (None, 64)                247872

 activation_5 (Activation)   (None, 64)                0

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 5)                 325

 activation_6 (Activation)   (None, 5)                 0

=================================================================
```

Model Training:

```
Epoch 1/5
4/4 [==============================] - ETA: 0s - loss: 1.8361 - cate
gorical_accuracy: 0.2000 Source path =  /content/drive/MyDrive/Proje
ct_data/Project_data/val ; batch size = 10

Epoch 1: saving model to model_init_2022-05-0204_35_49.090151/model-
00001-1.83609-0.20000-1.61232-0.20000.h5
4/4 [==============================] - 238s 76s/step - loss: 1.8361
- categorical_accuracy: 0.2000 - val_loss: 1.6123 - val_categorical_
accuracy: 0.2000 - lr: 0.0010
Epoch 2/5
4/4 [==============================] - ETA: 0s - loss: 1.6460 - cate
gorical_accuracy: 0.2750
Epoch 2: saving model to model_init_2022-05-0204_35_49.090151/model-
00002-1.64601-0.27500-1.60963-0.10000.h5
4/4 [==============================] - 225s 75s/step - loss: 1.6460
- categorical_accuracy: 0.2750 - val_loss: 1.6096 - val_categorical_
accuracy: 0.1000 - lr: 0.0010
Epoch 3/5
4/4 [==============================] - ETA: 0s - loss: 1.5490 - cate
gorical_accuracy: 0.3250
Epoch 3: saving model to model_init_2022-05-0204_35_49.090151/model-
00003-1.54905-0.32500-1.58369-0.30000.h5
4/4 [==============================] - 181s 60s/step - loss: 1.5490
- categorical_accuracy: 0.3250 - val_loss: 1.5837 - val_categorical_
accuracy: 0.3000 - lr: 0.0010
Epoch 4/5
4/4 [==============================] - ETA: 0s - loss: 1.6472 - cate
gorical_accuracy: 0.2500
Epoch 4: saving model to model_init_2022-05-0204_35_49.090151/model-
00004-1.64723-0.25000-1.56220-0.40000.h5
4/4 [==============================] - 183s 61s/step - loss: 1.6472
- categorical_accuracy: 0.2500 - val_loss: 1.5622 - val_categorical_
accuracy: 0.4000 - lr: 0.0010
Epoch 5/5
```

```
4/4 [==============================] - ETA: 0s - loss: 1.5414 - cate
gorical_accuracy: 0.2750
Epoch 5: saving model to model_init_2022-05-0204_35_49.090151/model-
00005-1.54143-0.27500-1.59141-0.20000.h5
4/4 [==============================] - 167s 56s/step - loss: 1.5414
- categorical_accuracy: 0.2750 - val_loss: 1.5914 - val_categorical_
accuracy: 0.2000 - lr: 0.0010
Execution time for this model is 1091.3605103492737 seconds
```

Experiment-3 Generator-2

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv3d_8 (Conv3D)           (None, 18, 100, 100, 8)   656

 activation_11 (Activation)  (None, 18, 100, 100, 8)   0

 batch_normalization_6 (Batc  (None, 18, 100, 100, 8)  32
 hNormalization)

 max_pooling3d_8 (MaxPooling  (None, 9, 50, 50, 8)     0
 3D)

 conv3d_9 (Conv3D)           (None, 9, 50, 50, 16)     3472

 activation_12 (Activation)  (None, 9, 50, 50, 16)     0

 batch_normalization_7 (Batc  (None, 9, 50, 50, 16)    64
 hNormalization)

 max_pooling3d_9 (MaxPooling  (None, 4, 25, 25, 16)    0
 3D)

 conv3d_10 (Conv3D)          (None, 4, 25, 25, 32)     13856

 activation_13 (Activation)  (None, 4, 25, 25, 32)     0

 batch_normalization_8 (Batc  (None, 4, 25, 25, 32)    128
 hNormalization)

 max_pooling3d_10 (MaxPoolin  (None, 2, 12, 12, 32)    0
 g3D)

 conv3d_11 (Conv3D)          (None, 2, 12, 12, 64)     55360

 activation_14 (Activation)  (None, 2, 12, 12, 64)     0

 dropout_7 (Dropout)         (None, 2, 12, 12, 64)     0

 max_pooling3d_11 (MaxPoolin  (None, 1, 6, 6, 64)      0
 g3D)

 flatten_3 (Flatten)         (None, 2304)              0

 dense_9 (Dense)             (None, 32)                73760

 activation_15 (Activation)  (None, 32)                0

 dropout_8 (Dropout)         (None, 32)                0

 dense_10 (Dense)            (None, 16)                528

 activation_16 (Activation)  (None, 16)                0

 dropout_9 (Dropout)         (None, 16)                0

 dense_11 (Dense)            (None, 5)                 85

 activation_17 (Activation)  (None, 5)                 0

=================================================================
```

Model Training:

```
Epoch 1/5
4/4 [==============================] - ETA: 0s - loss: 3.7601 - cate
gorical_accuracy: 0.2500
Epoch 1: saving model to model_init_2022-04-3009_16_08.034975/model-
00001-3.76008-0.25000-1.61105-0.20000.h5
4/4 [==============================] - 188s 62s/step - loss: 3.7601
- categorical_accuracy: 0.2500 - val_loss: 1.6110 - val_categorical_
accuracy: 0.2000 - lr: 0.0010
Epoch 2/5
4/4 [==============================] - ETA: 0s - loss: 2.6640 - cate
gorical_accuracy: 0.1800
Epoch 2: saving model to model_init_2022-04-3009_16_08.034975/model-
00002-2.66396-0.18000-1.60980-0.12000.h5
4/4 [==============================] - 189s 45s/step - loss: 2.6640
- categorical_accuracy: 0.1800 - val_loss: 1.6098 - val_categorical_
accuracy: 0.1200 - lr: 0.0010
Epoch 3/5
4/4 [==============================] - ETA: 0s - loss: 1.9980 - cate
gorical_accuracy: 0.1200
Epoch 3: saving model to model_init_2022-04-3009_16_08.034975/model-
00003-1.99802-0.12000-1.60875-0.12000.h5
4/4 [==============================] - 166s 42s/step - loss: 1.9980
- categorical_accuracy: 0.1200 - val_loss: 1.6088 - val_categorical_
accuracy: 0.1200 - lr: 0.0010
Epoch 4/5
4/4 [==============================] - ETA: 0s - loss: 1.6167 - cate
gorical_accuracy: 0.2800
Epoch 4: saving model to model_init_2022-04-3009_16_08.034975/model-
00004-1.61673-0.28000-1.60804-0.36000.h5
4/4 [==============================] - 121s 36s/step - loss: 1.6167
- categorical_accuracy: 0.2800 - val_loss: 1.6080 - val_categorical_
accuracy: 0.3600 - lr: 0.0010
Epoch 5/5
4/4 [==============================] - ETA: 0s - loss: 1.6145 - cate
gorical_accuracy: 0.1700
Epoch 5: saving model to model_init_2022-04-3009_16_08.034975/model-
00005-1.61446-0.17000-1.61237-0.20000.h5
```

```
4/4 [==============================] - 121s 32s/step - loss: 1.6145
- categorical_accuracy: 0.1700 - val_loss: 1.6124 - val_categorical_
accuracy: 0.2000 - lr: 0.0010
Execution time for this model is 865.5105922222137 seconds
```

Experiment-4 Generator-3

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed (TimeDistr  (None, 18, 64)           21933920
 ibuted)

 simple_rnn (SimpleRNN)      (None, 18, 32)            3104

 simple_rnn_1 (SimpleRNN)    (None, 16)                784

 dropout (Dropout)           (None, 16)                0

 dense_1 (Dense)             (None, 8)                 136

 dense_2 (Dense)             (None, 5)                 45

=================================================================
```

Model Training:

```
Epoch 1/5
9/9 [==============================] - ETA: 0s - loss: 1.7720 - cate
gorical_accuracy: 0.2296 Source path =  /content/drive/MyDrive/Proje
ct_data/Project_data/val ; batch size = 15

Epoch 1: saving model to model_init_2022-05-0106_09_32.484763/model-
00001-1.77196-0.22963-1.59131-0.25000.h5
9/9 [==============================] - 642s 79s/step - loss: 1.7720
- categorical_accuracy: 0.2296 - val_loss: 1.5913 - val_categorical_
accuracy: 0.2500 - lr: 0.0010
Epoch 2/5
9/9 [==============================] - ETA: 0s - loss: 1.6606 - cate
gorical_accuracy: 0.2296
Epoch 2: saving model to model_init_2022-05-0106_09_32.484763/model-
00002-1.66058-0.22963-1.62284-0.20000.h5
9/9 [==============================] - 750s 94s/step - loss: 1.6606
- categorical_accuracy: 0.2296 - val_loss: 1.6228 - val_categorical_
accuracy: 0.2000 - lr: 0.0010
Epoch 3/5
9/9 [==============================] - ETA: 0s - loss: 1.6182 - cate
gorical_accuracy: 0.2444
Epoch 3: saving model to model_init_2022-05-0106_09_32.484763/model-
00003-1.61819-0.24444-1.62288-0.21818.h5

Epoch 3: ReduceLROnPlateau reducing learning rate to 0.0005000000237
487257.
```

```
9/9 [==============================] - 693s 87s/step - loss: 1.6182
- categorical_accuracy: 0.2444 - val_loss: 1.6229 - val_categorical_
accuracy: 0.2182 - lr: 0.0010
Epoch 4/5
9/9 [==============================] - ETA: 0s - loss: 1.6598 - cate
gorical_accuracy: 0.2222
Epoch 4: saving model to model_init_2022-05-0106_09_32.484763/model-
00004-1.65978-0.22222-1.58682-0.25000.h5
9/9 [==============================] - 706s 83s/step - loss: 1.6598
- categorical_accuracy: 0.2222 - val_loss: 1.5868 - val_categorical_
accuracy: 0.2500 - lr: 5.0000e-04
Epoch 5/5
9/9 [==============================] - ETA: 0s - loss: 1.6545 - cate
gorical_accuracy: 0.2276
Epoch 5: saving model to model_init_2022-05-0106_09_32.484763/model-
00005-1.65452-0.22764-1.59653-0.30909.h5
9/9 [==============================] - 644s 72s/step - loss: 1.6545
- categorical_accuracy: 0.2276 - val_loss: 1.5965 - val_categorical_
accuracy: 0.3091 - lr: 5.0000e-04
Execution time for this model is 3437.321877479553 seconds
```

Experiment-5 Generator-3

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv3d (Conv3D)             (None, 16, 100, 100, 8)   656

 activation (Activation)     (None, 16, 100, 100, 8)   0

 conv3d_1 (Conv3D)           (None, 16, 100, 100, 8)   1736

 activation_1 (Activation)   (None, 16, 100, 100, 8)   0

 max_pooling3d (MaxPooling3D  (None, 8, 50, 50, 8)     0
 )

 conv3d_2 (Conv3D)           (None, 8, 50, 50, 16)     3472

 activation_2 (Activation)   (None, 8, 50, 50, 16)     0

 conv3d_3 (Conv3D)           (None, 6, 48, 48, 16)     6928

 activation_3 (Activation)   (None, 6, 48, 48, 16)     0

 max_pooling3d_1 (MaxPooling  (None, 3, 24, 24, 16)    0
 3D)

 conv3d_4 (Conv3D)           (None, 1, 22, 22, 32)     13856

 activation_4 (Activation)   (None, 1, 22, 22, 32)     0

 max_pooling3d_2 (MaxPooling  (None, 1, 11, 11, 32)    0
 3D)
```

```
 flatten (Flatten)           (None, 3872)              0

 dense (Dense)               (None, 64)                247872

 activation_5 (Activation)   (None, 64)                0

 dense_1 (Dense)             (None, 5)                 325

 activation_6 (Activation)   (None, 5)                 0

=================================================================
```

Model Training:

```
Epoch 1: saving model to model_init_2022-05-0110_35_45.069183/model-
00001-1.61730-0.16667-1.62476-0.35714.h5
6/6 [==============================] - 509s 100s/step - loss: 1.6173
- categorical_accuracy: 0.1667 - val_loss: 1.6248 - val_categorical_
accuracy: 0.3571 - lr: 0.0010
Epoch 2/5
6/6 [==============================] - ETA: 0s - loss: 1.6324 - cate
gorical_accuracy: 0.1429
Epoch 2: saving model to model_init_2022-05-0110_35_45.069183/model-
00002-1.63243-0.14286-1.61231-0.14286.h5
6/6 [==============================] - 499s 100s/step - loss: 1.6324
- categorical_accuracy: 0.1429 - val_loss: 1.6123 - val_categorical_
accuracy: 0.1429 - lr: 0.0010
Epoch 3/5
6/6 [==============================] - ETA: 0s - loss: 1.6084 - cate
gorical_accuracy: 0.1190
Epoch 3: saving model to model_init_2022-05-0110_35_45.069183/model-
00003-1.60844-0.11905-1.62307-0.14286.h5
6/6 [==============================] - 500s 100s/step - loss: 1.6084
- categorical_accuracy: 0.1190 - val_loss: 1.6231 - val_categorical_
accuracy: 0.1429 - lr: 0.0010
Epoch 4/5
6/6 [==============================] - ETA: 0s - loss: 1.6046 - cate
gorical_accuracy: 0.2143
Epoch 4: saving model to model_init_2022-05-0110_35_45.069183/model-
00004-1.60458-0.21429-1.57972-0.07143.h5
6/6 [==============================] - 494s 99s/step - loss: 1.6046
- categorical_accuracy: 0.2143 - val_loss: 1.5797 - val_categorical_
accuracy: 0.0714 - lr: 0.0010
Epoch 5/5
6/6 [==============================] - ETA: 0s - loss: 1.6372 - cate
gorical_accuracy: 0.3095
Epoch 5: saving model to model_init_2022-05-0110_35_45.069183/model-
00005-1.63717-0.30952-1.51710-0.35714.h5
6/6 [==============================] - 438s 88s/step - loss: 1.6372
- categorical_accuracy: 0.3095 - val_loss: 1.5171 - val_categorical_
accuracy: 0.3571 - lr: 0.0010
Execution time for this model is 2503.1452536582947 seconds
```

Experiment-6 Generator-4

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed_17 (TimeDi  (None, 18, 64)           21933920
 stributed)

 gru_34 (GRU)                (None, 18, 32)            9408

 gru_35 (GRU)                (None, 16)                2400

 dropout_20 (Dropout)        (None, 16)                0

 dense_55 (Dense)            (None, 8)                 136

 dense_56 (Dense)            (None, 5)                 45

=================================================================
```

Model Training:

```
Epoch 1/10
26/26 [==============================] - ETA: 0s - loss: 1.5504 - ca
tegorical_accuracy: 0.2836Source path =  /content/drive/Othercompute
rs/My Laptop/Project_data/val ; batch size = 13

Epoch 1: saving model to model_init_2022-04-3012_33_28.037408/model-
00001-1.55036-0.28356-0.68040-0.28022.h5
26/26 [==============================] - 77s 1s/step - loss: 1.5504
- categorical_accuracy: 0.2836 - val_loss: 0.6804 - val_categorical_
accuracy: 0.2802 - lr: 0.0010
Epoch 2/10
26/26 [==============================] - ETA: 0s - loss: 1.3915 - ca
tegorical_accuracy: 0.4118
Epoch 2: saving model to model_init_2022-04-3012_33_28.037408/model-
00002-1.39147-0.41176-0.16795-0.07280.h5
26/26 [==============================] - 67s 1s/step - loss: 1.3915
- categorical_accuracy: 0.4118 - val_loss: 0.1679 - val_categorical_
accuracy: 0.0728 - lr: 0.0010
Epoch 3/10
26/26 [==============================] - ETA: 0s - loss: 1.2008 - ca
tegorical_accuracy: 0.5128
Epoch 3: saving model to model_init_2022-04-3012_33_28.037408/model-
00003-1.20080-0.51282-0.01505-0.00275.h5
26/26 [==============================] - 66s 1s/step - loss: 1.2008
- categorical_accuracy: 0.5128 - val_loss: 0.0150 - val_categorical_
accuracy: 0.0027 - lr: 0.0010
Epoch 4/10
26/26 [==============================] - ETA: 0s - loss: 1.1545 - ca
tegorical_accuracy: 0.5158
Epoch 4: saving model to model_init_2022-04-3012_33_28.037408/model-
00004-1.15450-0.51584-0.01039-0.00687.h5
```

```
26/26 [==============================] - 59s 1s/step - loss: 1.1545
- categorical_accuracy: 0.5158 - val_loss: 0.0104 - val_categorical_
accuracy: 0.0069 - lr: 0.0010
Epoch 5/10
26/26 [==============================] - ETA: 0s - loss: 1.0576 - ca
tegorical_accuracy: 0.5566
Epoch 5: saving model to model_init_2022-04-3012_33_28.037408/model-
00005-1.05762-0.55656-0.15233-0.07418.h5
26/26 [==============================] - 67s 1s/step - loss: 1.0576
- categorical_accuracy: 0.5566 - val_loss: 0.1523 - val_categorical_
accuracy: 0.0742 - lr: 0.0010
Epoch 6/10
26/26 [==============================] - ETA: 0s - loss: 0.9900 - ca
tegorical_accuracy: 0.5913
Epoch 6: saving model to model_init_2022-04-3012_33_28.037408/model-
00006-0.98997-0.59125-0.14933-0.07555.h5

Epoch 6: ReduceLROnPlateau reducing learning rate to 0.0005000000237
487257.
26/26 [==============================] - 66s 1s/step - loss: 0.9900
- categorical_accuracy: 0.5913 - val_loss: 0.1493 - val_categorical_
accuracy: 0.0755 - lr: 0.0010
Epoch 7/10
26/26 [==============================] - ETA: 0s - loss: 0.8894 - ca
tegorical_accuracy: 0.6305
Epoch 7: saving model to model_init_2022-04-3012_33_28.037408/model-
00007-0.88944-0.63047-0.14591-0.07555.h5
26/26 [==============================] - 66s 1s/step - loss: 0.8894
- categorical_accuracy: 0.6305 - val_loss: 0.1459 - val_categorical_
accuracy: 0.0755 - lr: 5.0000e-04
Epoch 8/10
26/26 [==============================] - ETA: 0s - loss: 0.8169 - ca
tegorical_accuracy: 0.6712
Epoch 8: saving model to model_init_2022-04-3012_33_28.037408/model-
00008-0.81686-0.67119-0.14999-0.07280.h5

Epoch 8: ReduceLROnPlateau reducing learning rate to 0.0002500000118
743628.
26/26 [==============================] - 67s 1s/step - loss: 0.8169
- categorical_accuracy: 0.6712 - val_loss: 0.1500 - val_categorical_
accuracy: 0.0728 - lr: 5.0000e-04
Epoch 9/10
26/26 [==============================] - ETA: 0s - loss: 0.7962 - ca
tegorical_accuracy: 0.6727
Epoch 9: saving model to model_init_2022-04-3012_33_28.037408/model-
00009-0.79624-0.67270-0.12846-0.08379.h5
26/26 [==============================] - 66s 1s/step - loss: 0.7962
- categorical_accuracy: 0.6727 - val_loss: 0.1285 - val_categorical_
accuracy: 0.0838 - lr: 2.5000e-04
Epoch 10/10
26/26 [==============================] - ETA: 0s - loss: 0.7368 - ca
tegorical_accuracy: 0.7059
Epoch 10: saving model to model_init_2022-04-3012_33_28.037408/model
-00010-0.73681-0.70588-0.14561-0.07143.h5

Epoch 10: ReduceLROnPlateau reducing learning rate to 0.000125000005
9371814.
```

```
26/26 [==============================] - 66s 1s/step - loss: 0.7368
- categorical_accuracy: 0.7059 - val_loss: 0.1456 - val_categorical_
accuracy: 0.0714 - lr: 2.5000e-04
Execution time for this model is 691.1251420974731 seconds
```

Experiment-7 Generator-5

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed_23 (TimeDi  (None, 18, 64)           21933920
 stributed)

 gru_22 (GRU)                (None, 18, 32)            9408

 gru_23 (GRU)                (None, 16)                2400

 dropout_15 (Dropout)        (None, 16)                0

 dense_38 (Dense)            (None, 8)                 136

 dense_39 (Dense)            (None, 5)                 45

=================================================================
```

Model Training:

```
Epoch 1/10
42/42 [==============================] - ETA: 0s - loss: 1.6960 - ca
tegorical_accuracy: 0.2139
Epoch 1: saving model to model_init_2022-05-0110_13_15.923942/model-
00001-1.69602-0.21386-1.72439-0.23077.h5
42/42 [==============================] - 112s 1s/step - loss: 1.6960
- categorical_accuracy: 0.2139 - val_loss: 1.7244 - val_categorical_
accuracy: 0.2308 - lr: 0.0010
Epoch 2/10
42/42 [==============================] - ETA: 0s - loss: 1.6481 - ca
tegorical_accuracy: 0.2108
Epoch 2: saving model to model_init_2022-05-0110_13_15.923942/model-
00002-1.64807-0.21084-1.71295-0.20192.h5
42/42 [==============================] - 95s 1s/step - loss: 1.6481
- categorical_accuracy: 0.2108 - val_loss: 1.7130 - val_categorical_
accuracy: 0.2019 - lr: 0.0010
Epoch 3/10
42/42 [==============================] - ETA: 0s - loss: 1.6085 - ca
tegorical_accuracy: 0.2741
Epoch 3: saving model to model_init_2022-05-0110_13_15.923942/model-
00003-1.60855-0.27410-1.65812-0.26923.h5
42/42 [==============================] - 96s 1s/step - loss: 1.6085
- categorical_accuracy: 0.2741 - val_loss: 1.6581 - val_categorical_
accuracy: 0.2692 - lr: 0.0010
Epoch 4/10
```

```
42/42 [==============================] - ETA: 0s - loss: 1.5786 - ca
tegorical_accuracy: 0.2651
Epoch 4: saving model to model_init_2022-05-0110_13_15.923942/model-
00004-1.57859-0.26506-1.68373-0.29808.h5
42/42 [==============================] - 97s 1s/step - loss: 1.5786
- categorical_accuracy: 0.2651 - val_loss: 1.6837 - val_categorical_
accuracy: 0.2981 - lr: 0.0010
Epoch 5/10
42/42 [==============================] - ETA: 0s - loss: 1.5438 - ca
tegorical_accuracy: 0.3178
Epoch 5: saving model to model_init_2022-05-0110_13_15.923942/model-
00005-1.54381-0.31777-1.58436-0.29808.h5
42/42 [==============================] - 97s 1s/step - loss: 1.5438
- categorical_accuracy: 0.3178 - val_loss: 1.5844 - val_categorical_
accuracy: 0.2981 - lr: 0.0010
Epoch 6/10
42/42 [==============================] - ETA: 0s - loss: 1.5325 - ca
tegorical_accuracy: 0.3253
Epoch 6: saving model to model_init_2022-05-0110_13_15.923942/model-
00006-1.53254-0.32530-1.57960-0.34615.h5
42/42 [==============================] - 97s 1s/step - loss: 1.5325
- categorical_accuracy: 0.3253 - val_loss: 1.5796 - val_categorical_
accuracy: 0.3462 - lr: 0.0010
Epoch 7/10
42/42 [==============================] - ETA: 0s - loss: 1.5133 - ca
tegorical_accuracy: 0.3148
Epoch 7: saving model to model_init_2022-05-0110_13_15.923942/model-
00007-1.51332-0.31476-1.60042-0.33654.h5
42/42 [==============================] - 97s 1s/step - loss: 1.5133
- categorical_accuracy: 0.3148 - val_loss: 1.6004 - val_categorical_
accuracy: 0.3365 - lr: 0.0010
Epoch 8/10
42/42 [==============================] - ETA: 0s - loss: 1.4926 - ca
tegorical_accuracy: 0.3404
Epoch 8: saving model to model_init_2022-05-0110_13_15.923942/model-
00008-1.49257-0.34036-1.56150-0.32692.h5
42/42 [==============================] - 96s 1s/step - loss: 1.4926
- categorical_accuracy: 0.3404 - val_loss: 1.5615 - val_categorical_
accuracy: 0.3269 - lr: 0.0010
Epoch 9/10
42/42 [==============================] - ETA: 0s - loss: 1.4802 - ca
tegorical_accuracy: 0.3464
Epoch 9: saving model to model_init_2022-05-0110_13_15.923942/model-
00009-1.48020-0.34639-1.59241-0.32692.h5
42/42 [==============================] - 96s 1s/step - loss: 1.4802
- categorical_accuracy: 0.3464 - val_loss: 1.5924 - val_categorical_
accuracy: 0.3269 - lr: 0.0010
Epoch 10/10
42/42 [==============================] - ETA: 0s - loss: 1.4467 - ca
tegorical_accuracy: 0.3810
Epoch 10: saving model to model_init_2022-05-0110_13_15.923942/model
-00010-1.44674-0.38102-1.57979-0.36538.h5

Epoch 10: ReduceLROnPlateau reducing learning rate to 0.000500000023
7487257.
42/42 [==============================] - 97s 1s/step - loss: 1.4467
- categorical_accuracy: 0.3810 - val_loss: 1.5798 - val_categorical_
accuracy: 0.3654 - lr: 0.0010
Execution time for this model is 994.5941445827484 seconds
```

Experiment-8 Generator-5

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed_24 (TimeDi  (None, 18, 2, 2, 2048)   21802784
 stributed)

 time_distributed_25 (TimeDi  (None, 18, 8192)         0
 stributed)

 gru_24 (GRU)                (None, 32)                789696

 dense_40 (Dense)            (None, 32)                1056

 batch_normalization_1418 (B  (None, 32)               128
 atchNormalization)

 activation_1418 (Activation  (None, 32)               0
 )

 dropout_16 (Dropout)        (None, 32)                0

 dense_41 (Dense)            (None, 5)                 165

=================================================================
```

Model Training:

```
42/42 [==============================] - ETA: 0s - loss: 2.0454 - ca
tegorical_accuracy: 0.2274
Epoch 1: saving model to model_init_2022-05-0110_13_15.923942/model-
00001-2.04540-0.22741-1.85165-0.24038.h5
42/42 [==============================] - 122s 1s/step - loss: 2.0454
- categorical_accuracy: 0.2274 - val_loss: 1.8517 - val_categorical_
accuracy: 0.2404 - lr: 5.0000e-04
Epoch 2/20
42/42 [==============================] - ETA: 0s - loss: 1.8403 - ca
tegorical_accuracy: 0.2952
Epoch 2: saving model to model_init_2022-05-0110_13_15.923942/model-
00002-1.84029-0.29518-1.72840-0.27885.h5
42/42 [==============================] - 105s 1s/step - loss: 1.8403
- categorical_accuracy: 0.2952 - val_loss: 1.7284 - val_categorical_
accuracy: 0.2788 - lr: 5.0000e-04
Epoch 3/20
42/42 [==============================] - ETA: 0s - loss: 1.8141 - ca
tegorical_accuracy: 0.2711
Epoch 3: saving model to model_init_2022-05-0110_13_15.923942/model-
00003-1.81407-0.27108-1.62075-0.37500.h5
42/42 [==============================] - 105s 1s/step - loss: 1.8141
- categorical_accuracy: 0.2711 - val_loss: 1.6208 - val_categorical_
accuracy: 0.3750 - lr: 5.0000e-04
Epoch 4/20
```

```
42/42 [==============================] - ETA: 0s - loss: 1.7259 - ca
tegorical_accuracy: 0.3268
Epoch 4: saving model to model_init_2022-05-0110_13_15.923942/model-
00004-1.72587-0.32681-1.52996-0.39423.h5
42/42 [==============================] - 103s 1s/step - loss: 1.7259
- categorical_accuracy: 0.3268 - val_loss: 1.5300 - val_categorical_
accuracy: 0.3942 - lr: 5.0000e-04
Epoch 5/20
42/42 [==============================] - ETA: 0s - loss: 1.6093 - ca
tegorical_accuracy: 0.3449
Epoch 5: saving model to model_init_2022-05-0110_13_15.923942/model-
00005-1.60932-0.34488-1.53403-0.42308.h5
42/42 [==============================] - 105s 1s/step - loss: 1.6093
- categorical_accuracy: 0.3449 - val_loss: 1.5340 - val_categorical_
accuracy: 0.4231 - lr: 5.0000e-04
Epoch 6/20
42/42 [==============================] - ETA: 0s - loss: 1.4902 - ca
tegorical_accuracy: 0.4051
Epoch 6: saving model to model_init_2022-05-0110_13_15.923942/model-
00006-1.49020-0.40512-1.57137-0.40385.h5
42/42 [==============================] - 105s 1s/step - loss: 1.4902
- categorical_accuracy: 0.4051 - val_loss: 1.5714 - val_categorical_
accuracy: 0.4038 - lr: 5.0000e-04
Epoch 7/20
42/42 [==============================] - ETA: 0s - loss: 1.3714 - ca
tegorical_accuracy: 0.4142
Epoch 7: saving model to model_init_2022-05-0110_13_15.923942/model-
00007-1.37141-0.41416-1.35832-0.48077.h5
42/42 [==============================] - 104s 1s/step - loss: 1.3714
- categorical_accuracy: 0.4142 - val_loss: 1.3583 - val_categorical_
accuracy: 0.4808 - lr: 5.0000e-04
Epoch 8/20
42/42 [==============================] - ETA: 0s - loss: 1.3322 - ca
tegorical_accuracy: 0.4714
Epoch 8: saving model to model_init_2022-05-0110_13_15.923942/model-
00008-1.33220-0.47139-1.45915-0.46154.h5
42/42 [==============================] - 105s 1s/step - loss: 1.3322
- categorical_accuracy: 0.4714 - val_loss: 1.4592 - val_categorical_
accuracy: 0.4615 - lr: 5.0000e-04
Epoch 9/20
42/42 [==============================] - ETA: 0s - loss: 1.2849 - ca
tegorical_accuracy: 0.5000
Epoch 9: saving model to model_init_2022-05-0110_13_15.923942/model-
00009-1.28488-0.50000-1.36675-0.51923.h5
42/42 [==============================] - 104s 1s/step - loss: 1.2849
- categorical_accuracy: 0.5000 - val_loss: 1.3667 - val_categorical_
accuracy: 0.5192 - lr: 5.0000e-04
Epoch 10/20
42/42 [==============================] - ETA: 0s - loss: 1.2143 - ca
tegorical_accuracy: 0.5377
Epoch 10: saving model to model_init_2022-05-0110_13_15.923942/model
-00010-1.21426-0.53765-1.52890-0.50000.h5
42/42 [==============================] - 103s 1s/step - loss: 1.2143
- categorical_accuracy: 0.5377 - val_loss: 1.5289 - val_categorical_
accuracy: 0.5000 - lr: 5.0000e-04
Epoch 11/20
42/42 [==============================] - ETA: 0s - loss: 1.2121 - ca
tegorical_accuracy: 0.5120
```

```
Epoch 11: saving model to model_init_2022-05-0110_13_15.923942/model
-00011-1.21213-0.51205-1.27306-0.57692.h5
42/42 [==============================] - 103s 1s/step - loss: 1.2121
- categorical_accuracy: 0.5120 - val_loss: 1.2731 - val_categorical_
accuracy: 0.5769 - lr: 5.0000e-04
Epoch 12/20
42/42 [==============================] - ETA: 0s - loss: 1.1385 - ca
tegorical_accuracy: 0.5467
Epoch 12: saving model to model_init_2022-05-0110_13_15.923942/model
-00012-1.13851-0.54669-1.36495-0.52885.h5
42/42 [==============================] - 104s 1s/step - loss: 1.1385
- categorical_accuracy: 0.5467 - val_loss: 1.3650 - val_categorical_
accuracy: 0.5288 - lr: 5.0000e-04
Epoch 13/20
42/42 [==============================] - ETA: 0s - loss: 1.1481 - ca
tegorical_accuracy: 0.5497
Epoch 13: saving model to model_init_2022-05-0110_13_15.923942/model
-00013-1.14813-0.54970-1.43675-0.56731.h5
42/42 [==============================] - 105s 1s/step - loss: 1.1481
- categorical_accuracy: 0.5497 - val_loss: 1.4367 - val_categorical_
accuracy: 0.5673 - lr: 5.0000e-04
Epoch 14/20
42/42 [==============================] - ETA: 0s - loss: 1.0499 - ca
tegorical_accuracy: 0.6039
Epoch 14: saving model to model_init_2022-05-0110_13_15.923942/model
-00014-1.04986-0.60392-1.28469-0.55769.h5
42/42 [==============================] - 105s 1s/step - loss: 1.0499
- categorical_accuracy: 0.6039 - val_loss: 1.2847 - val_categorical_
accuracy: 0.5577 - lr: 5.0000e-04
Epoch 15/20
42/42 [==============================] - ETA: 0s - loss: 1.0773 - ca
tegorical_accuracy: 0.5934
Epoch 15: saving model to model_init_2022-05-0110_13_15.923942/model
-00015-1.07731-0.59337-1.27099-0.56731.h5
42/42 [==============================] - 103s 1s/step - loss: 1.0773
- categorical_accuracy: 0.5934 - val_loss: 1.2710 - val_categorical_
accuracy: 0.5673 - lr: 5.0000e-04
Epoch 16/20
42/42 [==============================] - ETA: 0s - loss: 1.0543 - ca
tegorical_accuracy: 0.5949
Epoch 16: saving model to model_init_2022-05-0110_13_15.923942/model
-00016-1.05426-0.59488-1.26326-0.56731.h5
42/42 [==============================] - 104s 1s/step - loss: 1.0543
- categorical_accuracy: 0.5949 - val_loss: 1.2633 - val_categorical_
accuracy: 0.5673 - lr: 5.0000e-04
Epoch 17/20
42/42 [==============================] - ETA: 0s - loss: 1.0142 - ca
tegorical_accuracy: 0.6205
Epoch 17: saving model to model_init_2022-05-0110_13_15.923942/model
-00017-1.01416-0.62048-1.26699-0.52885.h5
42/42 [==============================] - 105s 1s/step - loss: 1.0142
- categorical_accuracy: 0.6205 - val_loss: 1.2670 - val_categorical_
accuracy: 0.5288 - lr: 5.0000e-04
Epoch 18/20
42/42 [==============================] - ETA: 0s - loss: 0.9642 - ca
tegorical_accuracy: 0.6506
Epoch 18: saving model to model_init_2022-05-0110_13_15.923942/model
-00018-0.96416-0.65060-1.21050-0.57692.h5
```

```
42/42 [==============================] - 105s 1s/step - loss: 0.9642
- categorical_accuracy: 0.6506 - val_loss: 1.2105 - val_categorical_
accuracy: 0.5769 - lr: 5.0000e-04
Epoch 19/20
42/42 [==============================] - ETA: 0s - loss: 0.9283 - ca
tegorical_accuracy: 0.6506
Epoch 19: saving model to model_init_2022-05-0110_13_15.923942/model
-00019-0.92835-0.65060-1.22024-0.57692.h5
42/42 [==============================] - 104s 1s/step - loss: 0.9283
- categorical_accuracy: 0.6506 - val_loss: 1.2202 - val_categorical_
accuracy: 0.5769 - lr: 5.0000e-04
Epoch 20/20
42/42 [==============================] - ETA: 0s - loss: 0.9221 - ca
tegorical_accuracy: 0.6822
Epoch 20: saving model to model_init_2022-05-0110_13_15.923942/model
-00020-0.92211-0.68223-1.20541-0.47115.h5
42/42 [==============================] - 103s 1s/step - loss: 0.9221
- categorical_accuracy: 0.6822 - val_loss: 1.2054 - val_categorical_
accuracy: 0.4712 - lr: 5.0000e-04
Execution time for this model is 2106.233428001404 seconds
```

Experiment-9 Generator-5

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv3d (Conv3D)             (None, 18, 120, 120, 8)   656

 activation_1419 (Activation  (None, 18, 120, 120, 8)  0
 )

 batch_normalization_1419 (B  (None, 18, 120, 120, 8)  32
 atchNormalization)

 max_pooling3d (MaxPooling3D  (None, 9, 60, 60, 8)     0
 )

 conv3d_1 (Conv3D)           (None, 9, 60, 60, 16)     3472

 activation_1420 (Activation  (None, 9, 60, 60, 16)    0
 )

 batch_normalization_1420 (B  (None, 9, 60, 60, 16)    64
 atchNormalization)

 max_pooling3d_1 (MaxPooling  (None, 4, 30, 30, 16)    0
 3D)

 conv3d_2 (Conv3D)           (None, 4, 30, 30, 32)     13856

 activation_1421 (Activation  (None, 4, 30, 30, 32)    0
 )

 batch_normalization_1421 (B  (None, 4, 30, 30, 32)    128
 atchNormalization)

 max_pooling3d_2 (MaxPooling  (None, 2, 15, 15, 32)    0
 3D)

 conv3d_3 (Conv3D)           (None, 2, 15, 15, 64)     55360

 activation_1422 (Activation  (None, 2, 15, 15, 64)    0
 )

 flatten_17 (Flatten)        (None, 28800)             0

 dense_42 (Dense)            (None, 32)                921632

 activation_1423 (Activation  (None, 32)               0
 )

 dropout_17 (Dropout)        (None, 32)                0

 dense_43 (Dense)            (None, 16)                528

 activation_1424 (Activation  (None, 16)               0
 )

 dropout_18 (Dropout)        (None, 16)                0

 dense_44 (Dense)            (None, 5)                 85

 activation_1425 (Activation  (None, 5)                0
 )

=================================================================
```

Model Training:

```
Epoch 1/5
42/42 [==============================] - ETA: 0s - loss: 1.8648 - ca
tegorical_accuracy: 0.2048
Epoch 1: saving model to model_init_2022-05-0110_13_15.923942/model-
00001-1.86482-0.20482-2.05614-0.17308.h5
42/42 [==============================] - 95s 1s/step - loss: 1.8648
- categorical_accuracy: 0.2048 - val_loss: 2.0561 - val_categorical_
accuracy: 0.1731 - lr: 0.0010
Epoch 2/5
42/42 [==============================] - ETA: 0s - loss: 1.6317 - ca
tegorical_accuracy: 0.2319
Epoch 2: saving model to model_init_2022-05-0110_13_15.923942/model-
00002-1.63172-0.23193-1.77599-0.22115.h5
42/42 [==============================] - 94s 1s/step - loss: 1.6317
- categorical_accuracy: 0.2319 - val_loss: 1.7760 - val_categorical_
accuracy: 0.2212 - lr: 0.0010
Epoch 3/5
42/42 [==============================] - ETA: 0s - loss: 1.6584 - ca
tegorical_accuracy: 0.2063
Epoch 3: saving model to model_init_2022-05-0110_13_15.923942/model-
00003-1.65836-0.20633-1.69638-0.16346.h5
```

```
42/42 [==============================] - 94s 1s/step - loss: 1.6584
- categorical_accuracy: 0.2063 - val_loss: 1.6964 - val_categorical_
accuracy: 0.1635 - lr: 0.0010
Epoch 4/5
42/42 [==============================] - ETA: 0s - loss: 1.6461 - ca
tegorical_accuracy: 0.2154
Epoch 4: saving model to model_init_2022-05-0110_13_15.923942/model-
00004-1.64613-0.21536-1.67121-0.30769.h5
42/42 [==============================] - 94s 1s/step - loss: 1.6461
- categorical_accuracy: 0.2154 - val_loss: 1.6712 - val_categorical_
accuracy: 0.3077 - lr: 0.0010
Epoch 5/5
42/42 [==============================] - ETA: 0s - loss: 1.6456 - ca
tegorical_accuracy: 0.2364
Epoch 5: saving model to model_init_2022-05-0110_13_15.923942/model-
00005-1.64561-0.23645-1.73691-0.23077.h5
42/42 [==============================] - 94s 1s/step - loss: 1.6456
- categorical_accuracy: 0.2364 - val_loss: 1.7369 - val_categorical_
accuracy: 0.2308 - lr: 0.0010
Execution time for this model is 504.09635496139526 seconds
```

Experiment-10 Generator-5

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed (TimeDistr  (None, 18, 64)           15009664
 ibuted)

 gru (GRU)                   (None, 18, 32)            9408

 gru_1 (GRU)                 (None, 16)                2400

 dropout_3 (Dropout)         (None, 16)                0

 dense_4 (Dense)             (None, 8)                 136

 dense_5 (Dense)             (None, 5)                 45

=================================================================
```

Model Training:

```
Epoch 1/5
42/42 [==============================] - ETA: 0s - loss: 1.6805 - ca
tegorical_accuracy: 0.1700  Source path =  /content/drive/MyDrive/Pr
oject_data/Project_data/val ; batch size = 25

Epoch 1: saving model to model_init_2022-04-3009_16_08.034975/model-
00001-1.68045-0.17000-1.56649-0.24000.h5
42/42 [==============================] - 656s 214s/step - loss: 1.68
05 - categorical_accuracy: 0.1700 - val_loss: 1.5665 - val_categoric
al_accuracy: 0.2400 - lr: 0.0010
```

```
Epoch 2/5
42/42 [==============================] - ETA: 0s - loss: 1.6054 - ca
tegorical_accuracy: 0.2900
Epoch 2: saving model to model_init_2022-04-3009_16_08.034975/model-
00002-1.60536-0.29000-1.59764-0.20000.h5
42/42 [==============================] - 629s 210s/step - loss: 1.60
54 - categorical_accuracy: 0.2900 - val_loss: 1.5976 - val_categoric
al_accuracy: 0.2000 - lr: 0.0010
Epoch 3/5
42/42 [==============================] - ETA: 0s - loss: 1.6675 - ca
tegorical_accuracy: 0.2000
Epoch 3: saving model to model_init_2022-04-3009_16_08.034975/model-
00003-1.66747-0.20000-1.53375-0.36000.h5
42/42 [==============================] - 291s 97s/step - loss: 1.667
5 - categorical_accuracy: 0.2000 - val_loss: 1.5338 - val_categorica
l_accuracy: 0.3600 - lr: 0.0010
Epoch 4/5
42/42 [==============================] - ETA: 0s - loss: 1.6202 - ca
tegorical_accuracy: 0.1700
Epoch 4: saving model to model_init_2022-04-3009_16_08.034975/model-
00004-1.62019-0.17000-1.51384-0.32000.h5
42/42 [==============================] - 335s 86s/step - loss: 1.620
2 - categorical_accuracy: 0.1700 - val_loss: 1.5138 - val_categorica
l_accuracy: 0.3200 - lr: 0.0010
Epoch 5/5
42/42 [==============================] - ETA: 0s - loss: 1.5838 - ca
tegorical_accuracy: 0.2500
Epoch 5: saving model to model_init_2022-04-3009_16_08.034975/model-
00005-1.58381-0.25000-1.60315-0.24000.h5
42/42 [==============================] - 289s 68s/step - loss: 1.583
8 - categorical_accuracy: 0.2500 - val_loss: 1.6031 - val_categorica
l_accuracy: 0.2400 - lr: 0.0010
Execution time for this model is 2349.9846670627594
```

Experiment-11 Generator-6

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed_44 (TimeDi  (None, 18, 64)           21933920
 stributed)

 gru_42 (GRU)                (None, 18, 32)            9408

 gru_43 (GRU)                (None, 16)                2400

 dropout_32 (Dropout)        (None, 16)                0

 dense_71 (Dense)            (None, 8)                 136

 dense_72 (Dense)            (None, 5)                 45

=================================================================
```

Model Training:

```
Epoch 1/15
34/34 [==============================] - ETA: 0s - loss: 1.6905 - ca
tegorical_accuracy: 0.2075
Epoch 1: saving model to model_init_2022-05-0113_25_06.686063/model-
00001-1.69046-0.20746-1.72242-0.32000.h5
34/34 [==============================] - 125s 2s/step - loss: 1.6905
- categorical_accuracy: 0.2075 - val_loss: 1.7224 - val_categorical_
accuracy: 0.3200 - lr: 0.0010
Epoch 2/15
34/34 [==============================] - ETA: 0s - loss: 1.6175 - ca
tegorical_accuracy: 0.2612
Epoch 2: saving model to model_init_2022-05-0113_25_06.686063/model-
00002-1.61753-0.26119-1.71183-0.26000.h5
34/34 [==============================] - 97s 1s/step - loss: 1.6175
- categorical_accuracy: 0.2612 - val_loss: 1.7118 - val_categorical_
accuracy: 0.2600 - lr: 0.0010
Epoch 3/15
34/34 [==============================] - ETA: 0s - loss: 1.5802 - ca
tegorical_accuracy: 0.2836
Epoch 3: saving model to model_init_2022-05-0113_25_06.686063/model-
00003-1.58021-0.28358-1.67626-0.28000.h5
34/34 [==============================] - 94s 1s/step - loss: 1.5802
- categorical_accuracy: 0.2836 - val_loss: 1.6763 - val_categorical_
accuracy: 0.2800 - lr: 0.0010
Epoch 4/15
34/34 [==============================] - ETA: 0s - loss: 1.5669 - ca
tegorical_accuracy: 0.2925
Epoch 4: saving model to model_init_2022-05-0113_25_06.686063/model-
00004-1.56692-0.29254-1.69010-0.34000.h5
34/34 [==============================] - 100s 2s/step - loss: 1.5669
- categorical_accuracy: 0.2925 - val_loss: 1.6901 - val_categorical_
accuracy: 0.3400 - lr: 0.0010
Epoch 5/15
34/34 [==============================] - ETA: 0s - loss: 1.5198 - ca
tegorical_accuracy: 0.3313
Epoch 5: saving model to model_init_2022-05-0113_25_06.686063/model-
00005-1.51980-0.33134-1.62065-0.34000.h5
34/34 [==============================] - 99s 1s/step - loss: 1.5198
- categorical_accuracy: 0.3313 - val_loss: 1.6206 - val_categorical_
accuracy: 0.3400 - lr: 0.0010
Epoch 6/15
34/34 [==============================] - ETA: 0s - loss: 1.4829 - ca
tegorical_accuracy: 0.3343
Epoch 6: saving model to model_init_2022-05-0113_25_06.686063/model-
00006-1.48289-0.33433-1.60697-0.34000.h5
34/34 [==============================] - 103s 2s/step - loss: 1.4829
- categorical_accuracy: 0.3343 - val_loss: 1.6070 - val_categorical_
accuracy: 0.3400 - lr: 0.0010
Epoch 7/15
34/34 [==============================] - ETA: 0s - loss: 1.4579 - ca
tegorical_accuracy: 0.3940
Epoch 7: saving model to model_init_2022-05-0113_25_06.686063/model-
00007-1.45786-0.39403-1.64501-0.33000.h5
34/34 [==============================] - 94s 1s/step - loss: 1.4579
- categorical_accuracy: 0.3940 - val_loss: 1.6450 - val_categorical_
accuracy: 0.3300 - lr: 0.0010
Epoch 8/15
34/34 [==============================] - ETA: 0s - loss: 1.4449 - ca
tegorical_accuracy: 0.4015
Epoch 8: saving model to model_init_2022-05-0113_25_06.686063/model-
00008-1.44486-0.40149-1.59915-0.39000.h5
34/34 [==============================] - 94s 1s/step - loss: 1.4449
- categorical_accuracy: 0.4015 - val_loss: 1.5991 - val_categorical_
accuracy: 0.3900 - lr: 0.0010
Epoch 9/15
34/34 [==============================] - ETA: 0s - loss: 1.4148 - ca
tegorical_accuracy: 0.4209
Epoch 9: saving model to model_init_2022-05-0113_25_06.686063/model-
00009-1.41478-0.42090-1.56330-0.43000.h5
34/34 [==============================] - 95s 1s/step - loss: 1.4148
- categorical_accuracy: 0.4209 - val_loss: 1.5633 - val_categorical_
accuracy: 0.4300 - lr: 0.0010
Epoch 10/15
34/34 [==============================] - ETA: 0s - loss: 1.3791 - ca
tegorical_accuracy: 0.4478
Epoch 10: saving model to model_init_2022-05-0113_25_06.686063/model
-00010-1.37915-0.44776-1.62833-0.44000.h5
34/34 [==============================] - 94s 1s/step - loss: 1.3791
- categorical_accuracy: 0.4478 - val_loss: 1.6283 - val_categorical_
accuracy: 0.4400 - lr: 0.0010
Epoch 11/15
34/34 [==============================] - ETA: 0s - loss: 1.3694 - ca
tegorical_accuracy: 0.4761
Epoch 11: saving model to model_init_2022-05-0113_25_06.686063/model
-00011-1.36939-0.47612-1.55054-0.41000.h5
34/34 [==============================] - 93s 1s/step - loss: 1.3694
- categorical_accuracy: 0.4761 - val_loss: 1.5505 - val_categorical_
accuracy: 0.4100 - lr: 0.0010
Epoch 12/15
34/34 [==============================] - ETA: 0s - loss: 1.3283 - ca
tegorical_accuracy: 0.4881
Epoch 12: saving model to model_init_2022-05-0113_25_06.686063/model
-00012-1.32832-0.48806-1.54423-0.41000.h5
34/34 [==============================] - 94s 1s/step - loss: 1.3283
- categorical_accuracy: 0.4881 - val_loss: 1.5442 - val_categorical_
accuracy: 0.4100 - lr: 0.0010
Epoch 13/15
34/34 [==============================] - ETA: 0s - loss: 1.3127 - ca
tegorical_accuracy: 0.5000
Epoch 13: saving model to model_init_2022-05-0113_25_06.686063/model
-00013-1.31268-0.50000-1.53334-0.45000.h5
34/34 [==============================] - 100s 2s/step - loss: 1.3127
- categorical_accuracy: 0.5000 - val_loss: 1.5333 - val_categorical_
accuracy: 0.4500 - lr: 0.0010
Epoch 14/15
34/34 [==============================] - ETA: 0s - loss: 1.3055 - ca
tegorical_accuracy: 0.5060
Epoch 14: saving model to model_init_2022-05-0113_25_06.686063/model
-00014-1.30545-0.50597-1.47917-0.49000.h5
34/34 [==============================] - 95s 1s/step - loss: 1.3055
- categorical_accuracy: 0.5060 - val_loss: 1.4792 - val_categorical_
accuracy: 0.4900 - lr: 0.0010
Epoch 15/15
34/34 [==============================] - ETA: 0s - loss: 1.2678 - ca
tegorical_accuracy: 0.5284
```

```
Epoch 15: saving model to model_init_2022-05-0113_25_06.686063/model
-00015-1.26776-0.52836-1.48131-0.48000.h5
34/34 [==============================] - 94s 1s/step - loss: 1.2678
- categorical_accuracy: 0.5284 - val_loss: 1.4813 - val_categorical_
accuracy: 0.4800 - lr: 0.0010
Execution time for this model is 1478.5138931274414 seconds
```

Experiment-12 Generator-6

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed_65 (TimeDi  (None, 18, 2, 2, 2048)   21802784
 stributed)

 time_distributed_66 (TimeDi  (None, 18, 8192)         0
 stributed)

 gru_57 (GRU)                (None, 32)                789696

 batch_normalization_3414 (B  (None, 32)               128
 atchNormalization)

 dropout_50 (Dropout)        (None, 32)                0

 dense_87 (Dense)            (None, 32)                1056

 dropout_51 (Dropout)        (None, 32)                0

 dense_88 (Dense)            (None, 5)                 165

=================================================================
```

Model Training:

```
Epoch 1/10
34/34 [==============================] - ETA: 0s - loss: 2.0550 - ca
tegorical_accuracy: 0.2552
Epoch 1: saving model to model_init_2022-05-0113_25_06.686063/model-
00001-2.05496-0.25522-1.70560-0.34000.h5
34/34 [==============================] - 165s 2s/step - loss: 2.0550
- categorical_accuracy: 0.2552 - val_loss: 1.7056 - val_categorical_
accuracy: 0.3400 - lr: 0.0010
Epoch 2/10
34/34 [==============================] - ETA: 0s - loss: 1.6654 - ca
tegorical_accuracy: 0.4000
Epoch 2: saving model to model_init_2022-05-0113_25_06.686063/model-
00002-1.66541-0.40000-1.61284-0.32000.h5
34/34 [==============================] - 114s 2s/step - loss: 1.6654
- categorical_accuracy: 0.4000 - val_loss: 1.6128 - val_categorical_
accuracy: 0.3200 - lr: 0.0010
Epoch 3/10
34/34 [==============================] - ETA: 0s - loss: 1.4609 - ca
tegorical_accuracy: 0.4582
Epoch 3: saving model to model_init_2022-05-0113_25_06.686063/model-
00003-1.46088-0.45821-1.46185-0.53000.h5
34/34 [==============================] - 113s 2s/step - loss: 1.4609
- categorical_accuracy: 0.4582 - val_loss: 1.4618 - val_categorical_
accuracy: 0.5300 - lr: 0.0010
Epoch 4/10
34/34 [==============================] - ETA: 0s - loss: 1.1786 - ca
tegorical_accuracy: 0.5657
Epoch 4: saving model to model_init_2022-05-0113_25_06.686063/model-
00004-1.17857-0.56567-1.25129-0.51000.h5
34/34 [==============================] - 113s 2s/step - loss: 1.1786
- categorical_accuracy: 0.5657 - val_loss: 1.2513 - val_categorical_
accuracy: 0.5100 - lr: 0.0010
Epoch 5/10
34/34 [==============================] - ETA: 0s - loss: 1.0845 - ca
tegorical_accuracy: 0.5866
Epoch 5: saving model to model_init_2022-05-0113_25_06.686063/model-
00005-1.08447-0.58657-1.25782-0.60000.h5
34/34 [==============================] - 115s 2s/step - loss: 1.0845
- categorical_accuracy: 0.5866 - val_loss: 1.2578 - val_categorical_
accuracy: 0.6000 - lr: 0.0010
Epoch 6/10
34/34 [==============================] - ETA: 0s - loss: 0.9756 - ca
tegorical_accuracy: 0.6448
Epoch 6: saving model to model_init_2022-05-0113_25_06.686063/model-
00006-0.97557-0.64478-1.31670-0.62000.h5
34/34 [==============================] - 109s 2s/step - loss: 0.9756
- categorical_accuracy: 0.6448 - val_loss: 1.3167 - val_categorical_
accuracy: 0.6200 - lr: 0.0010
Epoch 7/10
34/34 [==============================] - ETA: 0s - loss: 0.9000 - ca
tegorical_accuracy: 0.6687
Epoch 7: saving model to model_init_2022-05-0113_25_06.686063/model-
00007-0.89998-0.66866-1.29458-0.62000.h5

Epoch 7: ReduceLROnPlateau reducing learning rate to 0.0005000000237
487257.
34/34 [==============================] - 110s 2s/step - loss: 0.9000
- categorical_accuracy: 0.6687 - val_loss: 1.2946 - val_categorical_
accuracy: 0.6200 - lr: 0.0010
Epoch 8/10
34/34 [==============================] - ETA: 0s - loss: 0.7375 - ca
tegorical_accuracy: 0.7597
Epoch 8: saving model to model_init_2022-05-0113_25_06.686063/model-
00008-0.73747-0.75970-1.15098-0.65000.h5
34/34 [==============================] - 108s 2s/step - loss: 0.7375
- categorical_accuracy: 0.7597 - val_loss: 1.1510 - val_categorical_
accuracy: 0.6500 - lr: 5.0000e-04
Epoch 9/10
34/34 [==============================] - ETA: 0s - loss: 0.7370 - ca
tegorical_accuracy: 0.7448
Epoch 9: saving model to model_init_2022-05-0113_25_06.686063/model-
00009-0.73699-0.74478-1.16342-0.65000.h5
34/34 [==============================] - 112s 2s/step - loss: 0.7370
- categorical_accuracy: 0.7448 - val_loss: 1.1634 - val_categorical_
accuracy: 0.6500 - lr: 5.0000e-04
Epoch 10/10
```

```
49/67 [==============================] - ETA: 0s - loss: 0.6839 - ca
tegorical_accuracy: 0.7592 - val_loss: 1.1634 - val_categorical_accu
racy: 0.6500 - lr: 5.0000e-04
Execution time for this model is 1160.9841442893721 seconds
```

Experiment-13 Generator-6

Model Architecture:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed_65 (TimeDi  (None, 18, 2, 2, 2048)   21802784
 stributed)

 time_distributed_66 (TimeDi  (None, 18, 8192)         0
 stributed)

 gru_57 (GRU)                (None, 32)                789696

 batch_normalization_3414 (B  (None, 32)               128
 atchNormalization)

 dropout_50 (Dropout)        (None, 32)                0

 dense_87 (Dense)            (None, 32)                1056

 dropout_51 (Dropout)        (None, 32)                0

 dense_88 (Dense)            (None, 5)                 165

=================================================================
```

Model Training:

```
Epoch 1/10
34/34 [==============================] - ETA: 0s - loss: 2.0290 - categor
ical_accuracy: 0.2269
Epoch 1: saving model to model_init_2022-05-0113_25_06.686063/model-00001
-2.02904-0.22687-1.70682-0.29000.h5
34/34 [==============================] - 157s 2s/step - loss: 2.0290 - ca
tegorical_accuracy: 0.2269 - val_loss: 1.7068 - val_categorical_accuracy:
0.2900 - lr: 5.0000e-04
Epoch 2/10
34/34 [==============================] - ETA: 0s - loss: 1.7182 - categor
ical_accuracy: 0.3582
Epoch 2: saving model to model_init_2022-05-0113_25_06.686063/model-00002
-1.71816-0.35821-1.45182-0.36000.h5
34/34 [==============================] - 116s 2s/step - loss: 1.7182 - ca
tegorical_accuracy: 0.3582 - val_loss: 1.4518 - val_categorical_accuracy:
0.3600 - lr: 5.0000e-04
Epoch 3/10
34/34 [==============================] - ETA: 0s - loss: 1.4583 - categor
ical_accuracy: 0.4030
Epoch 3: saving model to model_init_2022-05-0113_25_06.686063/model-00003
-1.45829-0.40299-1.56733-0.32000.h5
34/34 [==============================] - 118s 2s/step - loss: 1.4583 - ca
tegorical_accuracy: 0.4030 - val_loss: 1.5673 - val_categorical_accuracy:
0.3200 - lr: 5.0000e-04
Epoch 4/10
34/34 [==============================] - ETA: 0s - loss: 1.2825 - categor
ical_accuracy: 0.4866
Epoch 4: saving model to model_init_2022-05-0113_25_06.686063/model-00004
-1.28253-0.48657-1.36009-0.48000.h5
34/34 [==============================] - 120s 2s/step - loss: 1.2825 - ca
tegorical_accuracy: 0.4866 - val_loss: 1.3601 - val_categorical_accuracy:
0.4800 - lr: 5.0000e-04
Epoch 5/10
34/34 [==============================] - ETA: 0s - loss: 1.1760 - categor
ical_accuracy: 0.5343
Epoch 5: saving model to model_init_2022-05-0113_25_06.686063/model-00005
-1.17599-0.53433-1.38989-0.49000.h5
34/34 [==============================] - 118s 2s/step - loss: 1.1760 - ca
tegorical_accuracy: 0.5343 - val_loss: 1.3899 - val_categorical_accuracy:
0.4900 - lr: 5.0000e-04
Epoch 6/10
34/34 [==============================] - ETA: 0s - loss: 1.1649 - categor
ical_accuracy: 0.5478
Epoch 6: saving model to model_init_2022-05-0113_25_06.686063/model-00006
-1.16488-0.54776-1.14451-0.66000.h5
34/34 [==============================] - 122s 2s/step - loss: 1.1649 - ca
tegorical_accuracy: 0.5478 - val_loss: 1.1445 - val_categorical_accuracy:
0.6600 - lr: 5.0000e-04
Epoch 7/10
34/34 [==============================] - ETA: 0s - loss: 1.0821 - categor
ical_accuracy: 0.5761
Epoch 7: saving model to model_init_2022-05-0113_25_06.686063/model-00007
-1.08205-0.57612-1.25446-0.56000.h5
34/34 [==============================] - 120s 2s/step - loss: 1.0821 - ca
tegorical_accuracy: 0.5761 - val_loss: 1.2545 - val_categorical_accuracy:
0.5600 - lr: 5.0000e-04
Epoch 8/10
34/34 [==============================] - ETA: 0s - loss: 0.9900 - categor
ical_accuracy: 0.6343
Epoch 8: saving model to model_init_2022-05-0113_25_06.686063/model-00008
-0.98998-0.63433-1.30486-0.55000.h5
34/34 [==============================] - 119s 2s/step - loss: 0.9900 - ca
tegorical_accuracy: 0.6343 - val_loss: 1.3049 - val_categorical_accuracy:
0.5500 - lr: 5.0000e-04
Epoch 9/10
34/34 [==============================] - ETA: 0s - loss: 0.8825 - categor
ical_accuracy: 0.6910
Epoch 9: saving model to model_init_2022-05-0113_25_06.686063/model-00009
-0.88251-0.69104-1.11593-0.61000.h5
34/34 [==============================] - 119s 2s/step - loss: 0.8825 - ca
tegorical_accuracy: 0.6910 - val_loss: 1.1159 - val_categorical_accuracy:
0.6100 - lr: 5.0000e-04
Epoch 10/10
34/34 [==============================] - ETA: 0s - loss: 0.8858 - categor
ical_accuracy: 0.6761
Epoch 10: saving model to model_init_2022-05-0113_25_06.686063/model-0001
0-0.88578-0.67612-1.27196-0.62000.h5
34/34 [==============================] - 119s 2s/step - loss: 0.8858 - ca
tegorical_accuracy: 0.6761 - val_loss: 1.2720 - val_categorical_accuracy:
0.6200 - lr: 5.0000e-04
Execution time for this model is 1231.1138339042664 seconds
```