Hand Gesture Assignment Write-up

Problem Statement: From video Sequence of 30 images the model has to accurately classify 5 different class of gestures.

Models considered for building the solution.

1. 3D Convolution Neural Network (Conv3D).
2. 2D Convolution Neural Network + Recurrent Neural Network (ConvGRU).

Experiments Performed and their outcome:

Model Architecture, 3D Convolution Neural Network:

Model 1 (Conv3D):

Activation Function – Relu

Conv3D - 64 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) – Stride (2,2,2)

Conv3D - 128 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) – Stride (2,2,2)

Conv3D - 256 - Filter size (3*3*3) – Stride (1,1,1)
Conv3D - 256 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) - Stride (2,2,2)

Fully-Connected -512
Output -SoftMax (5)

*Batch Normalization Performed after Activation function.

Note: *Changes made in the experiment has been highlighted by underlying them.

Experiment Number	Model	Hyperparameters	Observation	Result
1	Model 1 (Conv3D)	**Batch Size:**100 **Image Sequence:**15 (skip 1 image in sequence) **Image Size:** 100*100 **Epochs:** 40 **ImageNormalization:** Scaled all channels by 1/255 (0-1 range) **Optimizer:** SGD (Nestrov) **LearningRate:** 0.001 **Momentum:** 0.7	Out of memory error and thus the batch size has to be reduced to fit into GPU memory. In the next model Batch size has to be reduced.	Resource Exhausted Error: OOM, During model compilation

2	Model 1 (Conv3D)	**Batch Size: 50** **Image Sequence:**15 (skip 1 image in sequence) **Image Size:** 100*100 **Epochs:** 40 **ImageNormalization:** Scaled all channels by 1/255 (0-1 range) **Optimizer:** SGD (Nestrov) **LearningRate:** 0.001 **Momentum:** 0.7	Model has been overfitting for this configuration. Validation and test losses have been decrementing at slower rate and thus increasing the learning rate and Epoch numbers	Accuracy Score categorical_accuracy: 0.8847 val_categorical_accuracy: 0.5600
3	Model 1 (Conv3D)	**Batch Size: 50** **Image Sequence:**15 (skip 1 image in sequence) **Image Size:** 100*100 **Epochs:** 50 **ImageNormalization:** Scaled all channels by 1/255 (0-1 range) **Optimizer:** SGD (Nestrov) **LearningRate:** 0.01 **Momentum:** 0.7	Model overfitting problem still persists and thus reducing the image size and increasing the batch size to check the model accuracy.	Accuracy Score categorical_accuracy: 0.9598 val_categorical_accuracy: 0.6100
4	Model 1 (Conv3D)	**Batch Size: 70** **Image Sequence:**15 (skip 1 image in sequence) **Image Size:** 64*64 **Epochs:** 50 **ImageNormalization:** Scaled all channels by 1/255 (0-1 range) **Optimizer:** SGD (Nestrov) **LearningRate:** 0.01 **Momentum:** 0.7	Model overfitting problem still persists. And thus, introducing Droupout layer in the next model.	Accuracy Score categorical_accuracy: 0.9575 val_categorical_accuracy: 0.7300

5	Model 1 (Conv3D) + Droupout Layer	**Batch Size: 80** **Image Sequence:**15 (skip 1 image in sequence) **Image Size:** 64*64 **Epochs:** 50 **ImageNormalization:** Scaled all channels by 1/255 (0-1 range) **Optimizer:** SGD (Nestrov) **LearningRate:** 0.01 **Momentum:** 0.9	Model is underfitting and thus Droupout layer is not helping. Reducing image sequence to 10 and removing the dropout layer.	Accuracy Score categorical_accuracy: 0.6048 val_categorical_accuracy: 0.5800
6	Model 1 (Conv3D)	**Batch Size: 80** **Image Sequence:**10 (skip 2 image in sequence) **Image Size:** 64*64 **Epochs:** 50 **ImageNormalization:** Scaled all channels by 1/255 (0-1 range) **Optimizer:** SGD (Nestrov) **LearningRate:** 0.01 **Momentum:** 0.9	Model is underfitting. Better strategy is: 1. Changing image normalization, by subtracting each pixel value per channel by mean value of channel in test image folder. 2. Changing Activation function to “Elu”. 3. Batch normalizing before activation function.	Accuracy Score categorical_accuracy: 0.7391 val_categorical_accuracy: 0.5900

Model Architecture 3D Convolution Neural Network:

Model 2 (Conv3D):

Activation Function – Elu

Conv3D - 64 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) – Stride (2,2,2)

Conv3D - 128 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) – Stride (2,2,2)

Conv3D - 256 - Filter size (3*3*3) – Stride (1,1,1)
Conv3D - 256 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) - Stride (2,2,2)

Fully-Connected -512
Output -SoftMax (5)

*Batch Normalization Performed Before Activation function.

Experiment Number	Model	Hyperparameters	Observation	Result
7	Model 2 (Conv3D)	**Batch Size: 70** **Image Sequence:**15 (skip 1 image in sequence) **Image Size:** 80*80 **Epochs:** 30 **ImageNormalization:** Subtracting pixel value per channel by mean value per channel. **Optimizer:** SGD (Nestrov) **LearningRate:** 0.01 **Momentum:** 0.7	Model is performing better compared to previous models. But it can be improved This is the **best Model** that was obtained using conv3D architecture with an accuracy score of 72%.	Accuracy Score categorical_accuracy: 0.8649 val_categorical_accuracy: 0.7200 Total parameters: 15,994,373

Model Architecture 3D Convolution Neural Network:

Model 3 (Conv3D):

Activation Function – Elu

Conv3D - 64 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) – Stride (2,2,2)

Conv3D - 128 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) – Stride (2,2,2)

Conv3D - 256 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) - Stride (2,2,2)

Conv3D - 256 - Filter size (3*3*3) – Stride (1,1,1)
MaxPooling3D - Filter size (2*2*2) - Stride (2,2,2)

Fully-Connected -512
Output -SoftMax (5)

*Batch Normalization Performed Before Activation function.

Experiment Number	Model	Hyperparameters	Observation	Result
8	Model 3 (Conv3D)	**Batch Size:70** **Image Sequence:**16 (Img num: 0,1 + skip 1 image in sequence) **Image Size:** 80*80 **Epochs:** 35 **ImageNormalization:** Subtracting pixel value per channel by mean value per channel. **Optimizer:** SGD (Nestrov) **LearningRate:** 0.01 **Momentum:** 0.7	Overall Model performance has reduced but the performance of the model remains intact.	Accuracy Score categorical_accuracy: 0.6644 val_categorical_accuracy: 0.7100

Model Architecture 2D Convolution Neural Network + Recurrent Neural Network:

Model 4 (ConvGRU):

Activation Function – Relu

Base CNN Model: VGG19 with non-trainable image-net weights.

GRU - 64
GRU - 16

Fully-Connected -64
Output -SoftMax (5)

Experiment Number	Model	Hyperparameters	Observation	Result
9	Model 4 (ConvGRU)	**Batch Size:40** **Image Sequence:**15 (skip 1 image in sequence) **Image Size:** 120*120 **Epochs:** 40 **ImageNormalization:** Scaled all channels by 1/255 (0-1 range) **Optimizer:** SGD (Nestrov) **LearningRate:** 0.01 **Momentum:** 0.7	Model is underfitting. but still it can be improved by changing the image normalization strategy.	Accuracy Score categorical_accuracy: 0.5446 val_categorical_accuracy: 0.5300

Model Architecture 2D Convolution Neural Network + Recurrent Neural Network:

Model 5 (ConvGRU):

Activation Fun – Relu

Base CNN Model: VGG19 with non-trainable image-net weights.

GRU - 32
GRU - 16

Fully-Connected -64
Output -SoftMax (5)

Experiment Number	Model	Hyperparameters	Observation	Result
10	Model 5 (ConvGRU)	**Batch Size:40** **Image Sequence:**16 (Img num: 0,1 + skip 1 image in sequence) **Image Size:** 120*120 **Epochs:** 35 **ImageNormalization:** Subtracting pixel value per channel by mean value per channel. **Optimizer:** SGD (Nestrov) **LearningRate:** 0.01 **Momentum:** 0.7	Model performance has been improved and this is **best Model** that was obtained using ConvGRU architecture with an accuracy score of 82%.	Accuracy Score categorical_accuracy: 0.9447 val_categorical_accuracy: 0.8200 Total Params: 20,332,581

Conclusion:

Model 2 (Conv3D, experiment 8) and Model 5 (ConvGRU, experiment 11) have produced very good accuracy score of around 72% and 82% respectively. But ConvGRU based models have lot more parameters when compared to conv3D based models. And, since the overall objective is to obtain a better performing model with lesser number of parameters conv3D model can be used for final model evaluation.

Use Model 2 (Conv3D, experiment 8) architecture with corresponding .h5 file for test evaluation.

Further accuracy of the model can be improved using data augmentation technique.