Deep Learning Course Project - Gesture Recognition

Problem Statement

Imagine you are working as a data scientist at a home electronics company which manufactures state of the art smart televisions. You want to develop a feature in the smart-TV that can recognize five different gestures performed by the user which will help users control the TV without using a remote.

The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

Each video is a sequence of 30 frames (or images).

## Objectives:

1. Generator: The generator should be able to take a batch of videos as input without any error. Steps like cropping, resizing and normalization should be performed successfully.
2. Model: Develop a model that is able to train without any errors which will be judged on the total number of parameters (as the inference(prediction) time should be less) and the accuracy achieved.Start training on a small amount of data and then proceed further.
3. Write up: This should contain the detailed procedure followed in choosing the final model. The write up should start with the reason for choosing the base model, then highlight the reasons and metrics taken into consideration to modify and experiment to arrive at the final model.

## Generator

This is one of the most important part of the code. The overall structure of the generator has been given. In the generator, you are going to preprocess the images as you have images of 2 different dimensions as well as create a batch of video frames. You have to experiment with img\_idx, y,z and normalization such that you get high accuracy.

## Model

Here you make the model using different functionalities that Keras provides. Remember to use Conv3D and MaxPooling3D and not Conv2D and Maxpooling2D for a 3D convolution model. You would want to use TimeDistributed while building a Conv2D + RNN model. Also remember that the last layer is the softmax. Design the network in such a way that the model is able to give good accuracy on the least number of parameters so that it can fit in the memory of the webcam.

Model1 - Conv3D Architecture

1. batch\_size = 32
2. num\_epochs = 20
3. Image resolution =(100 , 100)
4. Normalization method /255.

**Note:**

1. Each epoch is taking 40-45 sec for training.
2. Till 15th epoch the validation accuracy is poor and validation loss is very high. After 15th epoch the validation accuracy started improving and validation loss started decreasing.

Model2 - Conv3D Architecture

1. batch\_size = 32
2. num\_epochs = 20
3. Image resolution =(80 , 80)
4. Normalization method /255.

**Note:**

1. We tried to create a Conv3D model by considering training image resolution (120,120)
2. We end up with the Error: ResourceExhaustedError

**Decision:**

1. Change the batch size to 16 and keep Image Resolution (80,80) check the model performance.

Model3 - Conv3D Architecture

1. batch\_size = 16
2. num\_epochs = 20
3. Image resolution =(80,80)
4. Normalization method /255.

**Note:**

1. There is no improvement in the model performance
2. The loss of both training and validation dataset becomes nan means loss diverge to infinity.

Model4 - Conv3D Architecture

1. batch\_size = 32
2. num\_epochs = 30 # Increase the number of epochs
3. Image resolution =(100 , 100)
4. Normalization method /255.

**Note:**

1. The above model shows that afetr 15th epoch the accuracy for validation dataset starts improving and validation loss starts decreases.
2. This is a good sign we can get a good model with high validation accuracy and low validation loss if we train the model further.

Model 5 - CNN-LSTM Model

1. Transfer learning using VGG16 imagenet
2. batch\_size = 32
3. Number of Epochs = 30
4. Image Size = 100\*100
5. Optimizer = SGD

**Observation**

1. Training Accuracy: 78
2. Validation Accuracy: 63
3. Train Loss: 0.95
4. Val Loss: 1.09
5. We need to explore more models to increase accuracy.

Model 6 - CNN GRU and Transfer Learning

1. Transfer learning using VGG16 imagenet
2. batch\_size = 32
3. Number of Epochs = 20
4. Image Size = 100\*100
5. Optimizer = Adam

**Observation**

1. Training Accuracy: 99.8
2. Validation Accuracy: 85
3. Train Loss: 0.02
4. Val Loss: 0.64
5. This model is overfitting. So next we will apply some techniques to control overfitting.

Model 7

CNN, LSTM and GRU with transfer learning

**Observation**

1. Training Accuracy: 97
2. Validation Accuracy: 79
3. Train Loss: 0.11
4. Val Loss: 0.64
5. The model is overfitting and we need to control overfitting by providing learning rate and drop out layers

Model 7\_2

Try adding learning rate and dropout layers

**Observation**

1. Training Accuracy: 93.9
2. Validation Accuracy: 81
3. Train Loss: 0.41
4. Val Loss: 0.66
5. Good model but we can still further try to decrease gap between training and validation accuracy

Model 8.1: Conv3D Archtecture with data augmentation

Build the Conv3D model with data augmentation

1. batch\_size =20
2. num\_epochs = 20
3. Image Resolution=(100,100)
4. Training Accuracy: 62
5. Validation Accuracy: 63

Note: Good Model

Model 9

Lets apply augmentation on our best model so far which is 7.2

**Observation**

1. Training Accuracy: 97.6
2. Validation Accuracy: 82
3. Train Loss: 0.21
4. Val Loss: 0.66
5. This model is overfitting and we need to control the same.

**Model 10**

Transfer Learrning using Mobilenet and GRU

**Observation**

1. Training Accuracy: 99.6
2. Validation Accuracy: 94
3. Train Loss: 0.027
4. Val Loss: 0.14
5. Best Model with highest training and validation accuracy.