Computer Vision Assignment - 5

Mohith Kurakula

SBU ID: 112504214

mohith.kurakula@stonybrook.edu

1. Introduction:

In this Assignment i did Action recognition using Recurrent Neural Network (RNN), (Long-Short Term Memory) LSTM in particular for the first 25 classes. The Features are extracted for the First 25 classes using vgg classifier (i.e. 3360 vids and 25 images in each vid) and the LSTM to predict the action which is performed in the data

.

2. Dataset:

The given dataset is called UCF101.It is a tar file when extracted contains 101 Classes and 13320 Vid files each containing 25 images.Apart from this there is annos Folder which contains actions.txt and videos_labels_subset.txt. Actions.txt contains the 101 actions and the videos_labels_subset contains vid name, class(1-101) and 1-2 which determines train or test in a row.

The size of Total Training Data for 101 Classes :	9537
The size of Total Testing Data for 101 Classes :	3783
The size of Total Training Data for 25 Classes :	2409
The size of Total Testing Data for 25 Classes :	951

3. Approach Taken:

The Data is extracted with the code given in the PIazza post.Then train data name ,labels and test data name, labels are extracted by using the rows in the videos_labels_subsets.txt a function is written for it.

Cropping:

The images given are in the size of 256x340 whereas your pre-trained model might take nxn images as inputs so we have to convert it into 224x224.For this 5-way cropping is done for each image (i.e. Top Left,Top Right,Bottom Left ,Bottom Right ,Center). Average is done for this and normalize it using the provided code

```
normalize = transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224,
0.225])
prep = transforms.Compose([ transforms.ToTensor(), normalize ])
prep(img)
```

Preprocessing:

And the loaddataset function is the code given in the previous assignment. As the colab is crashing if large data is loaded i took 150 files each time and extracted the features did it 23 times.The loading of the last 60(i.e 150x22=3300 remaining 60) are printed in the notebook. Saving of the extraction of features (savemat) for the first 150 and the last 60 are printed in the colab.For Feature extraction VGG is used and we take only the first layer of the classifier from the VGG architecture.

***************after Modifying***************

```
Sequential(
  (0): Linear(in_features=25088, out_features=4096, bias=True)
  (1): ReLU(inplace=True)
)
```

After the .mat files are saved in the drive we load these save files to get the training and testing datasets.

Modelling:
LSTM:

The RNN Architecture:

```
RNN(
  (rnn): LSTM(4096, 64, num_layers=2, batch_first=True)
  (fc1): Linear(in_features=64, out_features=25, bias=True)
)
```

out_features=25 because we are only taking 25 classes. The code for training and testing of the model is similar to the one which I used in the previous assignment. It is trained for 50 Epochs and after training the Results are as Follows:

Evaluation:

Accuracy:
The accuracy of Model for the train Data is : 99%
The accuracy of Model for the test Data is : 81%

SVM:

The accuracy of my Model for Test Data is : 100.00%
The accuracy of my Model is : 89.58%

Conclusion:

Even Though SVM is a simpler Model than LSTM-based RNN the accuracy got for SVM (89.58%) is greater than that of LSTM(81%) it out performs LSTM one of the reason for this may be due to the fact that there is less data for training.