Kevin Pan

Deep Learning

Final Project Report

Due: May. 12. 2021

Problem Statement:

The goal of this project is to recognize the human actions in video data. There are a total of 11 classes, such as “play tennis”, “playing basketball”, etc. Each video is preprocessed to have 30 frames, 64 by 64 pixels, and 3 channels (RGB). The solution involves using a deep neural network model to train on the train videos along with their ground truth labels. Then the model is used to classifiy the test videos. Finally, the model preformance will be calculated.

Model Description:

The model utilizes a Two Stream Convolutional Neural Network. The first stream will capture the video details and the second stream will capture the temporal dependency. The output of both streams will be fused into 1.

In order to achieve this, stream 1 consists of CNN + MLP. The input is the second video frame of the batch videos, and the output has the shape of (batch * 11 classes). For the convolution layer, each filter size is 3*3 and strides is. Valid padding will be used across all convolution layers. The detailed structure is as such:

Input (batchSize * 1 * 64 * 64 * 3)
Conv2D (32 filters)
ReLu Activation
Batch Norm
MaxPool2D
Conv2D (64 filters)
ReLu Activation
Batch Norm
MaxPool2D
Conv2D (64 filters)
ReLu Activation
Batch Norm
MaxPool2D
MLP
Output (batchSize * classes)

Stream 2 consists of CNN + LSTM. The input is batch number of 30 frame videos (batchSize * 30 frames * 64 * 64 * 3), and the output has the shape of (batch * 11 classes). For the convolution layer, each filter size is 3*3 and strides is. Valid padding will be used across all convolution layers. The detailed structure is as such:

Input (batchSize * 30 * 64 * 64 * 3)

Conv2D (32 filters)

ReLu Activation

Batch Norm

MaxPool2D

Conv2D (64 filters)

ReLu Activation

Batch Norm

MaxPool2D

Conv2D (64 filters)

ReLu Activation

Batch Norm

MaxPool2D

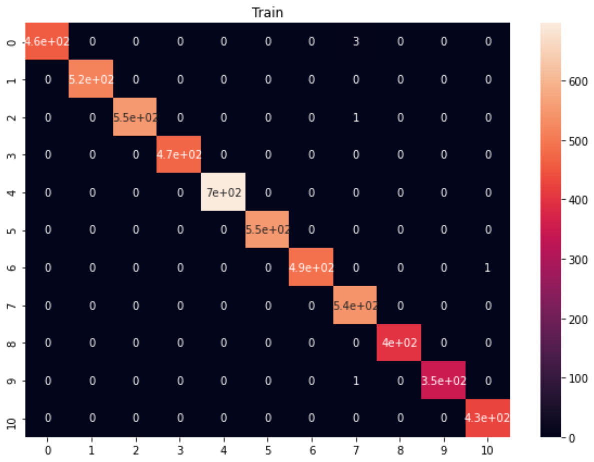
LSTM

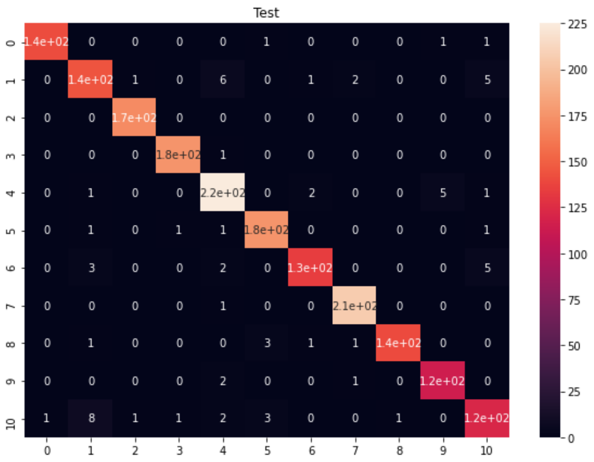
Reshape

Output (batchSize * classes)

Finally, the output from both streams are fused using: output = output0 + output1.

Confusion Matrix:

<table>
<tr><th>Train Dataset</th><th>Test Dataset</th></tr>
<tr><td>


</td><td>


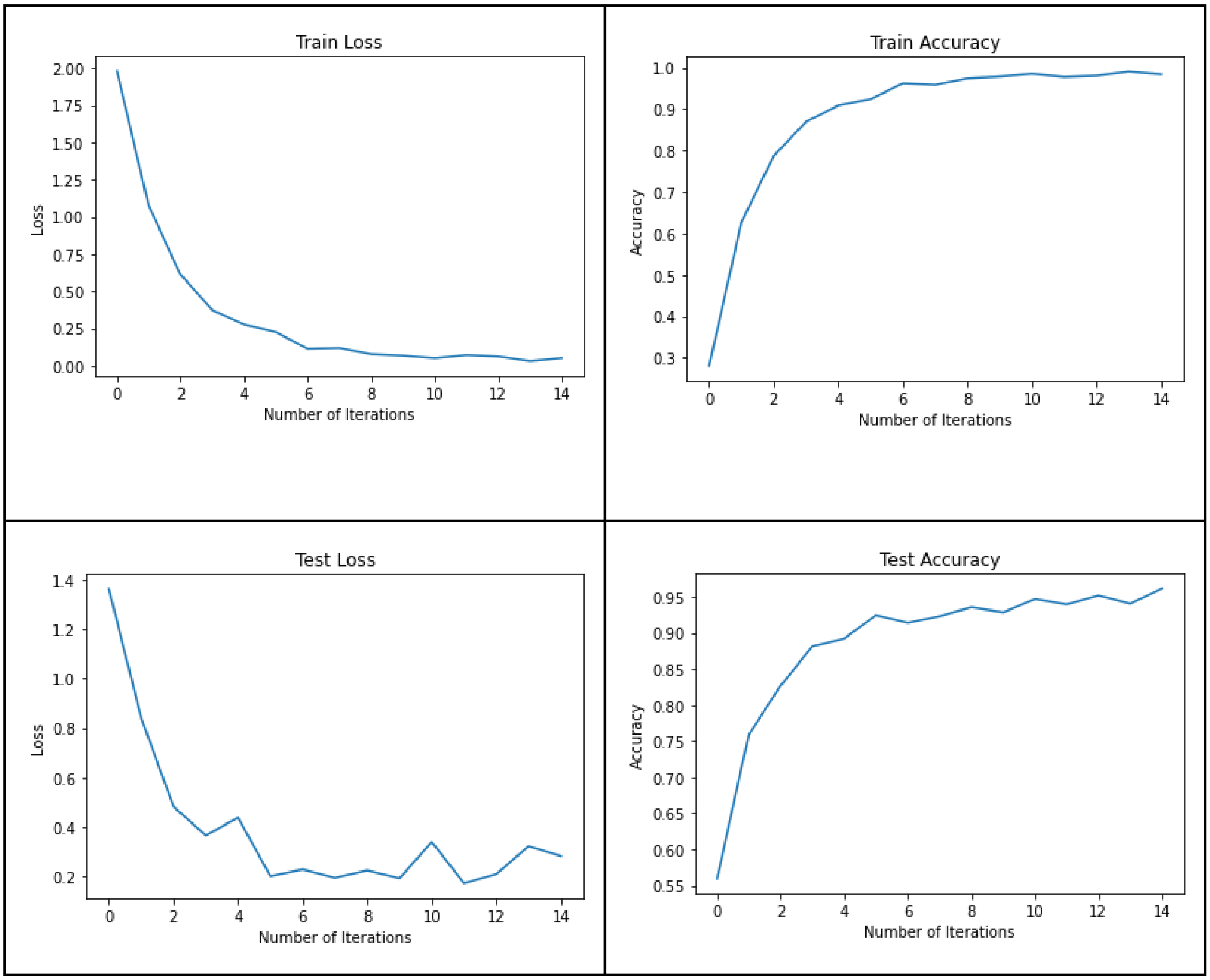
</td></tr>
</table>

Model Evaluation Results:

final train accuracy: 0.9979853479853481

final test accuracy: 0.9615664845173042

Plots of Errors(Loss) and Accuracies:



Results and Discussions:

Programming Environment:

Python Version: 3.8

Tensorflow Version: 2.4.1

GPU: GeForce RTX 2060

Iteration_number = 15

Batch size = 30

Learning rate = 0.001

The hyper parameters includes: learning_rate, batch_size, iteration_number

The learning_rate affects the rate of convergence. In the project, it is set to 0.001.

The batch_size is how many samples are selected from 1 batch. In the project, it is set to 30.

The iteration_number is how many epochs the program will perform. In the project, it is set to 15 which gives the best preformance.