Spotlight: A Smart Video Highlight Generator

Jun-Ting (Tim) Hsieh, Chengshu (Eric) Li, Wendi (Wendy) Liu
Project Mentor: De-An Huang, Stanford AI Lab
Stanford University, Department of Computer Science

Introduction

Despite the amazing progress we have made in classifying and describing images, we cannot yet achieve the same with videos. Motivated to help people understand videos in the most efficient manner, we use various deep learning models in this project to generate important highlights from video clips.

YouTube vimeo GoPro Be a HERO.

Problem Statement

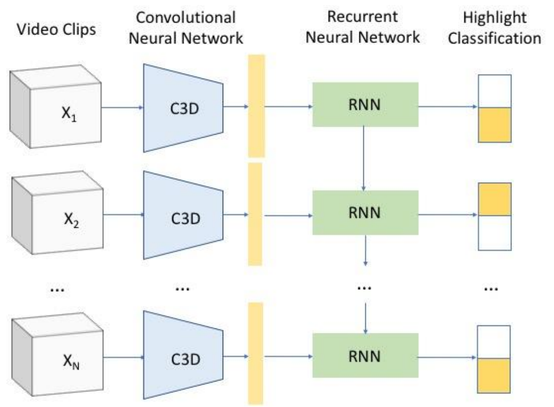
The goal of this project is to build a model that recognizes the "interesting" parts of a video clip. Given a video, the model is expected to return a list of time/frame intervals that represent the highlight parts of the video. The focus of this project is on user-generated videos (UGV).

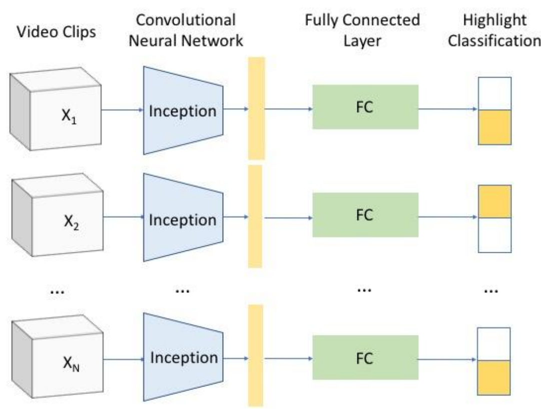
Dataset

Number of videos		3687
Number of frames		3900000
Summary	Video	Highlight
Median	25.9s	3.3s
Mean	35.2s	6.8s
SD	29.7s	8.5s

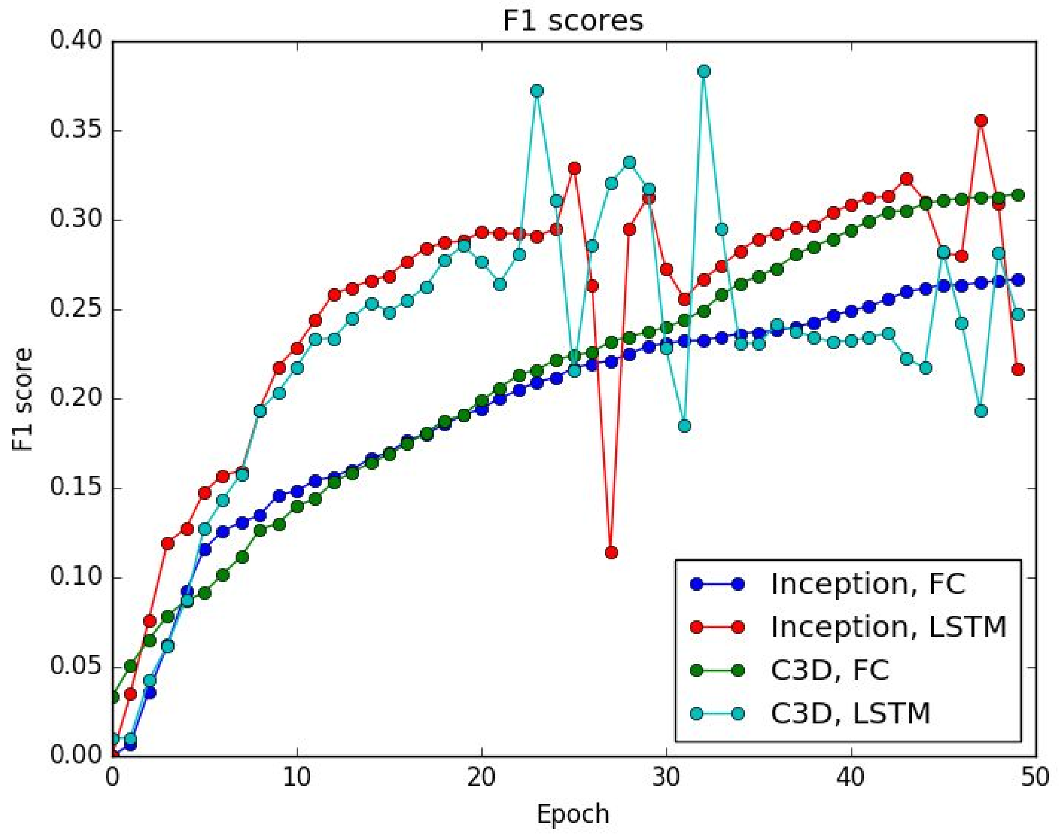
Models

Our model is divided into two phases: video-to-embedding and embedding-to-label. First, video clips are transformed into embeddings. Then, each embedding is classified as either a highlight or not. For the first layer, we use C3D and Inception. For the second layer, we use fully connected layers and bi-directional LSTM. In total, we run four experiments: 1) C3D + FC, 2) C3D + BLSTM, 3) Inception + FC and 4) Inception + BLSTM.



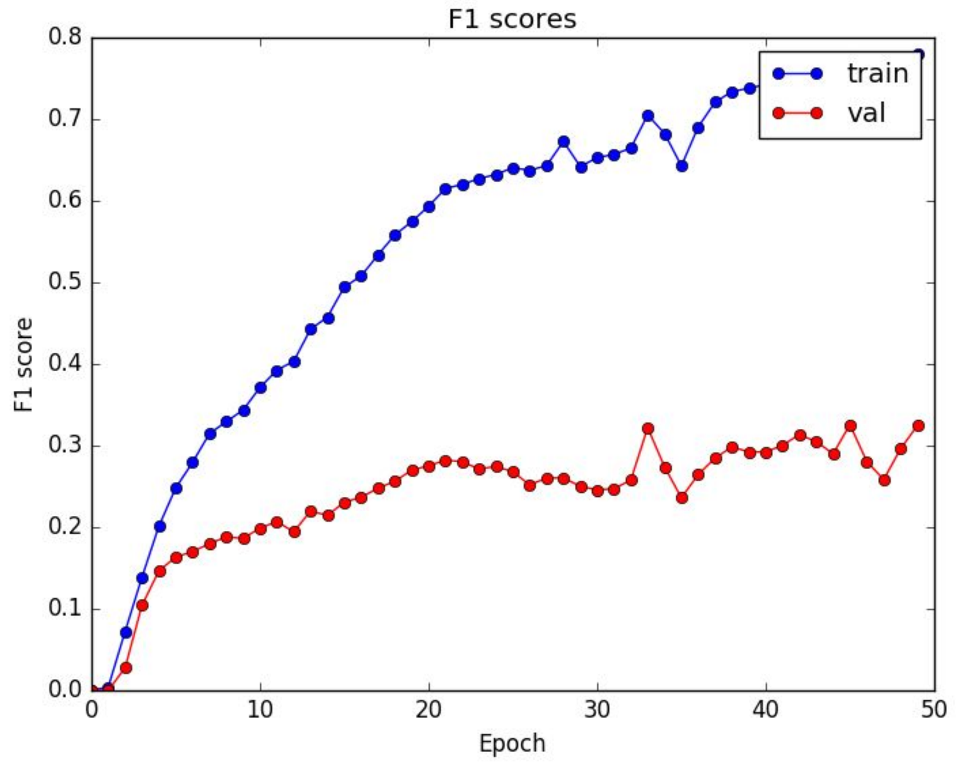


Results



Analysis

As expected, LSTM performs slightly better than fully connected layers. Yet we also notice that LSTM has greater variance during training and requires more fine-grained hyperparameter search. In order to alleviate over-fitting, we also try dropout. This actually decreases the validation F1 score, which we plan to investigate next.



Conclusion

Given the great variety of topics for UGVs, we have achieved decent results for classifying video highlights: our F1 score is two times higher than that of random classification. For future work, we will try to use ResNet, VGG and SqueezeNet for the first layer of our model. We will also try SVM hinge loss, as opposed to cross entropy loss. We might also look for other ways to control over-fitting, like image distortion and regularization.