July 29, 2018

Today we discuss possible future work for ONR project. First we recap several available datasets for video scene and shot detection. For now, we find TRECVID, BBC Planet earth, RAI and OVSD. Previously, people use color histograms, edge detectors to extract features from stream of images. Then some naïve change detection like difference and thresholding can be used. Recently, people try to adapt deep models especially convolutional neural network to video scene and shot detection. They use 3D convolution (2D spatial + time) and treat this problem as classification problem. Each frame can be labeled as in-trainsition, non-transition or sharp, gradual transition and so on.

Previously, Jie and I have discussed possible future works of this project. We agreed on some kind of dictionary building framework in order to create an embedding of abstract notion from images and video. During the discussion, Vahid pointed out that what we were describing is closely related to a topic could video indexing. I did some google search on this topic and I found out that there exists a newly published industry level video indexer called [Microsoft Video Indexer](https://vi.microsoft.com/). It illustrates part of our idea and the differences are mainly technical. For example, we want to the dictionary built automatically from an unsupervised way instead of prescribing a range of labels. In fact, video shot and scene change detection are recognized as low-level video indexing. They are mainly the primary step to high-level video indexing where we extract abstract notion like “man, women, sitting, laughing, etc”.

During the meeting, we have agreed on exploring change detection on coded image and video. Normally, people do computation based on the raw data decoded coded video file like mp4, avi etc. However, it would be computationally favorable if we can apply algorithm directly on the coded version. Vahid briefly introduce how MPEG works. We can use this information as our starting point for understand how these scheme work and the coded images look like. Then we summarize two main possible tracks for algorithm on coded image in general:

“Compression from AI”

Deep models like VGG network and other off-the-shelf network have a series of convolutional neural network as feature extractor and a number of fully-connected layers as classifier. Typically, features extracted from convolution layers are low dimension representation of the original image. This feature extraction process can be regarded as compressing image to useful features. In video setting, we can treat those features as a coded version of video like MPEG. Then, if we were able to perform change detection on extracted features then we do not need to decode images first and then run our algorithm. Instead, we can simply run our algorithm directly on coded video.

One possible solution to this track is to use autoencoder as our model. Typically, we encode our raw image to a low dimensional vector and then decode it back to itself. It is simply a regression task, and we can treat the low dimensional vector as our features. Then we can perform shot boundary detection algorithm on those features. This method is different from existing techniques. Most of their methods only encode the raw data and use the features to classify. However, there does not exist a mapping from those feature back to its original image. Thus, we can hardly recognize features from naive CNN as a coding scheme like MPEG. The parameters of the autoencoder may be optimized not only from the loss with respect to the decoder but also with respect to the purpose our analysis, i.e. shot boundary detection. Simply put a classifier on the feature should produce a result. We would have a deep model based coding scheme for a specific purpose like shot boundary detection.

“AI from Compressed data”

Since there are existing coding standards for audio, image and video. Normally, we will decode the compressed data first, extract features from the raw data, and finally apply some algorithm like shot boundary detection on those features. The question is that whether we can directly extract features from the existing coding standards. We may not be able to do as good as using raw data. However, extracting features from coded image consumes less memory and we can perform analysis on raw data only if necessary. So far, most deep models cannot handle high resolution picture and we have to preprocess the image into low resolution first (typically 32x32, 228x228 for ImageNet). Thus there is a trade-off between compression and analysis. In situation where memory or transmission bandwidth is scarce, balancing between the coding scheme and performance of algorithm is essential.

The coding scheme introduced in the first track is not unique. Apart from all those existing coding scheme MPEG, simply optimize the autoencoder with respect to other or even no purpose can create totally different coding scheme and thus different coded representation. We may be able to map any kind of coded image to a low dimensional vector. After some study of MPEG and other standard video coding scheme, we will try to use only coded image as our input instead of raw image. We hypothesize that with those coded image we can still perform shot boundary detection and other algorithms. Besides, it would be interesting if we were able to convert between images coded from different encoders without explicitly decode them to the raw data first.