Human Activity Detection for Surveillance Video Compression

Problem Statement

Activity detection is a major problem in smart videos surveillance. It is a fundamental problem in computer vision, i.e. to detect the activity of human in surveillance videos. These applicats need real-time detection performance, but it is generally vey time consuming to detect the actual activity. The time consumption is due to the heavy size of the video clip of surveillance and low computation power of these systems. This heavy size is because of the resolution of the cctv camera. It becomes important to reduce the resolution of video clip and to detect what activity is been performed by the subjects. There are many solutions provided in deep learning until now, but none of them are efficient when there are lots of details in the video and it becomes difficult to detect the actual activity. In such case if rest of the details are compressed, it will be easy to apply attention to the actual activity.

Background

The above problem we are discussing about two parts, one is compression and other one is activity detection. We will see more about how these can participate in performing real-time recognition of human activity on surveillance cameras.

Compression and human activity are two different operations on a video. There are many approaches of compressing the video, some focus on overall compression and some on adaptive approaches. Much more practical application is adaptive compression of video. Adaptive video compression is about compressing only those parts of videos in which there is least focus, rest all the things are not compressed.

In human activity recognition there are many computer vision techniques to detect the human activity, starting from image processing to machine learning and then deep learning too. There are many hybrid techniques which are using two different techniques to identify the activity like applying image processing and machine learning techniques together. There are many approaches which use two or more machine learning techniques to detect human activity. For deep learning approaches, since we are talking about activity recognition from video many CNN based approaches can be found. These CNN's based approaches drastically improved the efficiency of detecting the activities. Our focus is to integrate adaptive compression with the deep learning approaches to improve the efficiency in real-time activity recognition.

Methodology

There are two parts of recognizing the human activity from surveillance video clips i.e compression and then activity detection. Following are the steps to perform the activity recognition in conjunction with the compression:

Step 1: Extract visual features from the surveillance video.

Step 2: Adaptive Video Compression.

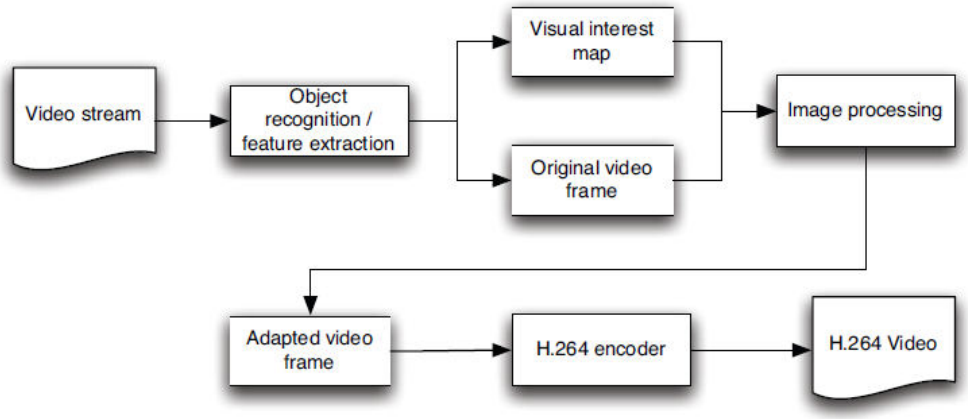

Figure 1: Adaptive Video Compression ["Adaptive Video Compression For Video Surveillance Applications" by Andrew D. Bagdanov, Marco Bertini, Alberto Del Bimbo, Lorenzo Seidenari. In 2011 IEEE International Symposium on Multimedia]

Step 3: Prepare input for model, i.e. extracted feature from step 1 is passed to the network.

Step 4: One of the approach is using Contextual Multi-Scale Region Convolution 3D Network (CMS_RC3D). This model has 4 network layers, first is Feature extraction layer, which extracts the features, but here we can remove this layer as we are already passing the extracted feature to the network. Second layer is Temporal Feature Pyramid Network, third layer as Activity Proposal Network and last layer as Activity Classification Network.

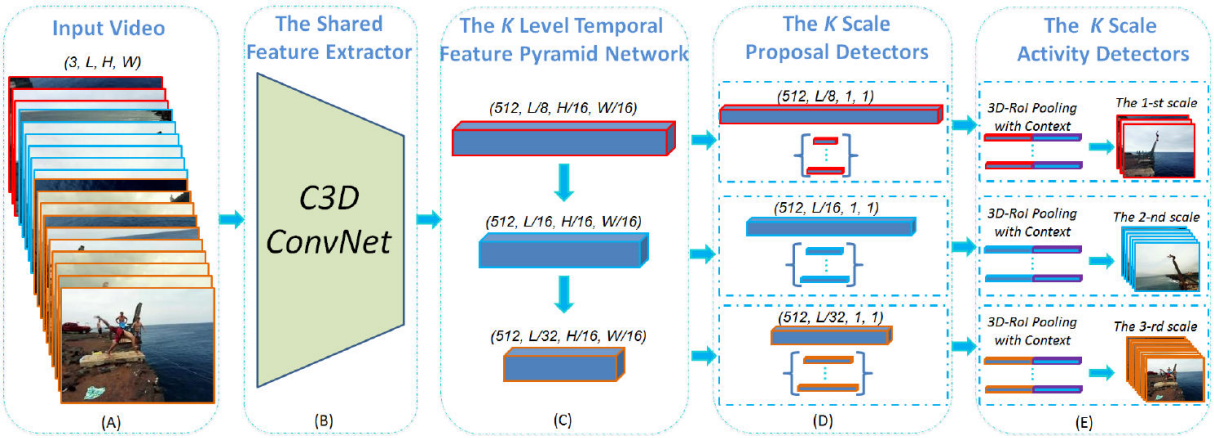

Figure 2: The Pipeline of CMS-RC3D ["Contextual Multi-Scale Region Convolutional 3D Network for Activity Detection" by Yancheng Bai, Huijuan Xu, Kate Saenko, Bernard Ghanem. arXiv:1801.09184v1 [cs.CV] 28 Jan 2018]

Experimental Design

Dataset: There are many video databases for such applications, the list can be found on this link. The dataset that are provided on the link are of different type, the only common about all the dataset is that they all have classes of activities.

Evaluation Measures: The model can be evaluated based on accuracy of prediction of class. The model can be evaluated using Mean Average Precision (mAP) over all classes. There are

many other methods to evaluate the accuracy. The evaluation can also be done on F1 scores, this measurement can help to understand whether the model will work properly in real world scenario.

Software and Hardware Requirements: Python based Computer Vision and Deep Learning libraries will be exploited for the development and experimentation of the project. Tools such as Anaconda Python, and libraries such as OpenCV, Tensorflow, and Keras will be utilized for this process.