CS 231N Project Milestone
Complex Video Querying with Weak Supervision

Swetha Revanur
Stanford University
srevanur@cs.stanford.edu

Vishnu Sarukkai
Stanford University
sarukkai@stanford.edu

1. Introduction

As cameras become an increasingly ubiquitous part of modern life, videos offer a valuable source of data for a variety of applications. A significant number of these cameras are stationary, from traffic cameras, to home security and geostationary satellites. However, it is intractable for humans to manually observe most of this incoming content. As a result, in order to fully utilize the data that is available to us, it is crucial to be able to query across untrimmed videos to quickly locate events in a large search space.

1.1. Action Recognition

While images lend themselves to object detection, videos provide us with added temporal contributions as well. Therefore, the obvious benefit gained from video data is that we can now detect not only objects, but also actions. When analyzing images, the classical task of object detection identifies the area of the image where a given object is located. Given that actions are changes in an object over time, first detecting regions of interest (RoIs) in each frame that may contain objects is a prerequisite for classifying actions as well. This leads us to the notion of action tubes, which are temporal sequences of RoIs corresponding to every objects action. We apply action tubes in this paper to detect actions over time.

1.2. Weak Supervision

Given that visual sources of data have become more prevalent, acquiring labeled training data has become the primary bottleneck in supervised machine learning with them. This motivates the need for an alternative, more flexible supervision strategy. Here, we explore weak supervision, which employs methods to take a dataset that is largely unlabeled and tag the rest of the dataset with lower-quality or noisier labels. Specifically, we aim to accomplish this task in the domain of action recognition in videos, where we will use weak labeling to architect a model for complex action classification. By complex actions, we refer to queries such as car turning left or truck skidding into a lamp post. This will allow us to create general models for various complex video queries without the need for a benchmark dataset.

2. Problem Statement

The dataset we are using for this paper is a 19-hour stationary video from a four-way street intersection in Jackson Hole, Wyoming. This video contains 312321 frames. In addition, we have also obtained bounding boxes for the vehicles present in each frame of the video by courtesy of Daniel Kang and Paroma Varma of Stanford's InfoLab. These bounding boxes are associated with 4205 different objects. Though the task of identifying bounding boxes for objects in video over time is an important one, we believe that there is already significant literature that has tackled this problem. Rather, in this paper we focus on the problem of labeling the data effectively with weak supervision for complex action detection. The effectiveness of the labeling algorithm will be reflected in the accuracy of the action detection.

Using weak supervision, we will be labeling sequences of bounding boxes with an action as well, creating an action tube. These actions will be composed of three labels: the type of vehicle, the incoming direction of the vehicle, and the outgoing direction of the vehicle. Next, we will train an RNN to predict the actions associated with the action tubes. As discussed in the section below, it is possible that we will need to update the processing of the action tubes. The accuracy of a prediction for a single tube will be measured by the fraction of the labels (out of 3) that are predicted correctly. For instance, a prediction for an action that correctly identifies the type of vehicle and the incoming direction but fails to identify the outgoing direction will have an accuracy of $\frac{2}{3}$. We will then calculate the average accuracy across all tubes in the test dataset. We hope to obtain accuracy over 0.9, in line with the 0.1 error threshold used in action detection in Kang et. al.'s "BlazeIt: Fast Exploratory Video Queries using Neural Networks." In addition, the work in BlazeIt on aggregate queries can be used

to verify our model on a case-by-case basis.

3. Technical Approach

3.1. Initial Labeling

To begin, we manually labeled 7500 frames (or 2.4% of the total dataset) encompassing the motion of 22 vehicles. Each frame was given three labels: $vehicle$, $inDirection$, and $outDirection$. $inDirection$ tells us which direction the vehicle enters the frame from, and analogously, $outDirection$ tells us which direction it exits from. A $vehicle$ is either a car (given label 0), or a truck (given label 1). Possible values for $inDirection$ and $outDirection$ are front (0), right (1), back (2), and left (3).

3.2. Tube Normalization

Given the bounding boxes discussed above, for each object, we can isolate frame-specific regions of interest (RoIs).

Figure 1: Bounding box surrounding car in frame

We can do this for all frames that the object is present in. By linking these RoIs over time, we produce action tubes. However, we are now confronted by two issues. First, within a tube, the RoIs all have varying spatial dimensions. Furthermore, action tubes may have different time spans, ranging from 10 frames to several hundreds. To address this problem, we apply what we call tube normalization. Tube normalization is a two-step process that reduces varying volume action tubes to a consistent single 20x20 pixel image. First, each frame is padded such that its new dimensions are divisible by 20. Next, spatial max pooling is applied to transform each frame in the tube to a 20x20 pixel image. Finally, we apply temporal max pooling to generate a single 20x20 pixel image that serves as a summary frame of the entire tube (ergo for every unique object). All frames discussed thus far have a third dimension, which contains the RGB channel metadata.

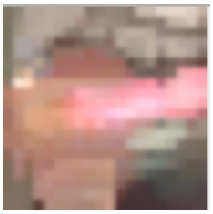

Figure 2: Summary frame from the car above's action tube

3.3. Baseline Model

For our baseline model, each tubes summary frame is flattened and passed into a fully-connected neural network with one hidden layer. The output layer is of size 32, so one score is produced for each of the 32 classes representing all permutations of 2 vehicles, 4 incoming directions, and 4 outgoing directions. We optimize over a cross-entropy loss using stochastic gradient descent with Nesterov momentum. This model was implemented using PyTorch. Given the limited number of labeled tubes, the model grossly overfits. Accuracy here is the ratio of number of correctly labeled tubes to the total number of tubes.

3.4. Details of Weak Supervision

Out of the 4205 action tubes in the dataset, we will eventually manually label 15%. First, we will set aside 10% of the action tubes as the test dataset and manually label it in order to eventually use it for our accuracy computation. Next, we will label 5% of the data as a starting point for weakly supervised learning, which will in turn provide weak labels for the remaining 85% of the data. Of the 90% of data not in the testing dataset, 70% will be allocated for training, and the remaining 20% will be used as validation. The true labels and weakly supervised labels will be shuffled randomly prior to the training-validation split.

Moving forward, after finishing the manual labeling of 15% of the dataset, we will noisily label the rest of the data through weak supervision. We will experiment with the method of weak supervision best suited to the task, but we hope to try implementing a variety of heuristics, and we will also attempt to use multiple biased classification models such as boosting that have had some success on small datasets.

In building a full-fledged model, we will use recurrent neural networks (RNNs), perfectly suited to capturing temporal relationships between frames of video. After applying the preprocessing steps described earlier to normalize the size of every image to 20x20 pixels, we will not apply temporal max pooling to generate a single 20x20 pixel image. Rather, we will feed the series of images into an RNN which relies on a combination of convolutional layers, normalization layers, and rectified linear units (ReLUs).

4. Preliminary Results and Future Directions

At this point in the research process, we were able to successfully label 7500 frames that comprise 22 action tubes. For this small set, we were able to apply tube normalization and architect a baseline model that accepts the output

of tube normalization. Going forward, we will be fleshing out our weakly supervised approach and better understanding how that fits into an RNN-based model.

5. References

U. Ahsan et al., DiscrimNet: Semi-Supervised Action Recognition from Videos using Generative Adversarial Networks, CVPR 2018.

D. Kang et al., BlazeIt: Fast Exploratory Video Queries using Neural Networks, PVLDB, 2018.

L. Wang et al., UntrimmedNets for Weakly Supervised Action Recognition and Detection, CVPR 2017.

G. Ye et al., Large-Scale Video Event Detection, Columbia University, 2015.