Tracked-base semi-supervised learning on camera-based system

Phumchanit Watanaprakornkul
Stanford Univerisity
Stanford University. Stanford, California 94305
yiam@stanford.edu

Abstract

One problem in supervised learning is that we need a lot of data in order to get a reliable classifier. In object recognition, there is a way to reduce the size of manually labeled object needed by using track information of an object. This semi-supervised learning method using track information has already been done on data from laser range finder. The goal of this project is to extend such result to camera-based system. We could do background subtraction and segmentation on frames and then applied the aforementioned semi-supervised learning algorithm.

1. Introduction

Supervised learning object recognition algorithm has one main weakness: the need for large amount of labeled training data. One way to remedy the weakness is to generate more labeled data from a few manually labeled ones. We can label object of one type in one frame then use a model-free segmentation and tracking to find the place for the same object in the next frame. We then label the object in the next frame of the same type.

The idea of using tracking information to generate more labeled data for supervised learning is already done in [1]. However, the data from LIDAR sensor used in [1] is very different from frames that we get from normal camera. We cannot apply exact same process because the segmentation and track classification will be very different. In addition, camera is cheaper than LIDAR sensor.

2. Dataset

Our dataset is a video of overhead view from Hoover tower to the fountain in front of it. The camera is not moving. We are interested in recognizing moving objects, in this case, pedestrians and bikers. We label a few objects in a few frames of the video for training data.

3. Algorithm

The main part of this project is to generate training data for the object classifier. One approach could be that we can specify positions of our objects of interest in the video and track them. However, our model free segmentation and tracking have limitations which can cause us to lose the object sometimes. In addition, some we cannot add more training data in some cases such as the labeled pedestrian just walk under a tree within a few frames.

The better approach is from the training video, we generate a list of objects and their positions in each frames. Then, we can choose objects that appear in many frames and label a few of those manually. This way, we get more data for the classifier.

There are mainly four steps for doing semi-supervised learning method using track information. We first subtrack the background from our video. Then, we do segmentation to extract objects of interest from each frames. We then do tracking to map the same objects of interest across frames. These three steps are model free and they will give a list of tracked objects for us to label. The last step is classification.

3.1. Background subtraction

Given that our camera is not moving and we are interested in moving objects, we can remove the non-moving background. This can be done by standard background subtraction algorithm.

We use openCV Gaussian mixture based background subtraction algorithm described in [3]. This algorithm works in most cases. For a frame, all frames before it are included in our Gaussians and the later frames are more important than the former. This handles brightness change, and small movement of the camera; however, it creates some problem as well. When our object of interest stops moving for a few frames in the video, it disappears into the background. We will detect it when it starts moving again and we will not be able to associate the objects before and after the stop to be the same object.

We might be able to fix this issue by use tracking information; however, we currently choose to just ignore it because we already have enough data and we can pick other tracked object anyway.

3.2. Image Segmentation

After removing the background, we then have to divide the frame into segments/cluster of possible objects. We do this on the foreground that we get from background subtraction.

The first approach for this model-free segmentation is to use mean-shift. However, it is difficult to assign a correct window size for mean shift and the algorithm has sizable running time. Given that we have many frames to process and each frames has millions pixels, mean shift is too slow.

We have a simpler and faster method that works well for our problem. We can operate on the foreground mask, which is a matrix of bits, instead of the foreground image. And given that in general, our object of interest will not overlap each other. We can just group the connected pixels in the mask together as a cluster. We do this by using flood fill. This is a lot faster than mean shift. Sometimes, pixels of an object are not connected because our background subtraction is imperfect. We can fix this by applying Gaussian blur filter to fill the missing pixels in the foreground mask. Alternately, we can get the similar result by modifying flood fill to go to near pixels in some radius instead of only adjacent pixels. From our tests, radius r = 3 works well with openCV Gaussian mixture based background subtraction.

The limitation of this method is that it cannot deal with overlapped objects such as a group of pedestrians walking together. It will cluster the whole group as one object. Therefore, our method is ineffective when there are a lot of object of interest in a frame at once and they are potentially overlap. In our test video, there is only one group of pedestrians walking across the scene, so we can just pick other pedestrians as training data.

Note that we could try to use mean shift on a group of pedestrians in the foreground; however, the resulting clusters will be several heads and several shirts instead of separated pedestrians. In general, mean shift does not work for overlapped object of interest either.

3.3. Track Classification

In this step, we add the tracking information into each segment in the frame, determining that this segment come from which segment in the previous frame.

Seeing that simple methods work well for segmentation, we approach this part with another simple and fast method. For each cluster of pixels representing an object of interest, we use the center of the cluster (the mean) to represent its position. We then match the cluster with the nearest cluster in the previous frame, labeling them as the same object. There are cases where this would mistakenly match two different objects together. For example, when one object moves out of the scene at the same time the other one moves in. We fix this by setting a threshold for the distance we consider.

Another complication is that the background subtraction and segmentation is not perfect, so sometimes we lost the object in a few frames. For example, two pedestrians walk in opposite direction overlap each other in a few frames before walking away. Our segmentation cannot handle the overlap, so we lost the objects in that frame. To fix this, we look back into more frames in the past and match objects based on distance of the clusters and the distances in time frame.

This method works better than we expected. Initially, we are going to implement Kalman filter to predict the position of a cluster in the next frame and then match the objects there. However, the matching from simple method above is good enough. In the test video, we have some noises in a cluster of swirling tree leafs, but those are not in the region of our objects of interest. So, it posts no problem. We might attempt Kalman filter depending on the result of the classification.

3.4. Object Classification

We then use the extra training data labeled by tracking information to train our usual object classifier. This part is not the main point of the project. We will probably use part based classifier as described in [4]. We are currently working on this part.

4. Result & Evaluation

We have an acceptable implementation of the first three steps: background subtraction, segmentation, and tracking. The test result of the test video is at http://stanford.edu/~yiam/tracked.avi. The same object is labeled the same number. From that video, we can pick a few objects and use them as training data when we have a working classifier.

The goal of our experiment is to label only a few objects (about ten labels from our 10 minutes video with pedestrians and bikes as our objects of interest). Then, show that using only those few manually labeled data, our algorithm produce similar accuracy to common object recognition algorithm. We probably test the accuracy by eye at first (let the classifier draw a box on recognized object in each frame). Hopefully, we could manually label some part of the video as test data. (For test data, it’s more work since we need to label everything in every frame.)

References

[1] Alex Teichman, and Sebastian Thrun. “Tracking-based semi-supervised learning”.

[2] Thanarat Horprasert, David Harwood, Larry S. Davis. “A Robust Background Subtraction and Shadow Detection”

[3] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection", Proc. 2nd European Workshop on Advanced Video-Based Surveillance Systems, 2001. < http://personal.ee.surrey.ac.uk/Personal/R.Bowden/publications/avbs01/avbs01.pdf >

[4] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, No. 9, September 2010

[5] N. Martorana, L. Masi, M. Meoni, “Kalman and Condensation in video-tracking”. Universita Degli Studidi Firenze

[6] Gary Bradski and Adrian Kaehler. “Learning OpenCV” O'Reilly Media, 2008.

5. Appendix

This project extends the result of tracking-based supervised learning done on LIDAR data[1] to data from normal camera. This project is supervised by Alex Teichman.

Future Distribution Permission

The author of this report gives permission for this document to be distributed to Stanford-affiliated students taking future courses.