Report on Assignment 4

Rahul Bangar, Gururaj Mulay, Tariqul Sifat

April 17, 2017

Abstract

Goal of this assignment is to integrate the object detection and tracking systems from assignment 2 and 3 by applying a pre-trained deep convolutional network (VGG [1], [2]) on the attention windows and tracked windows to classify the objects within those windows.

1 Introduction

Our system consist of three major parts. First is the attention window detection wherein the system captures the most interesting windows from a video frame. Although we applied it on a dynamic video, it is particularly useful in finding static attention windows such as house, lamp-posts, etc. Second is the tracking where the systems tracks a detected window in frame sequences though time. Both produce a list of rectangular attention windows written to a csv file. Finally, the trained VGG classier processes these attention windows to output the most probable label of that window captured either from the static background or the dynamic foreground.

A short overview of all the implemented components in the system is given below. The next section will discuss the integration of the components mentioned here.

- **Attention Window Detection:** PA2
  - SIFT: to detect the keypoints of interest
    - Threshold-based detection (naive, naive2)
    - Clustering-based detection (HCA, DBSCAN)
  - Saliency-based detection
- **Foreground Detection and Tracking:** PA3
  - Foreground Detection: MoG (actually used), ViBE
  - Window Tracking: MOSSE
- **Classification** PA4
  - Pre-trained VGG

For attention window detection, we are currently using DBSCAN to cluster the keypoints that form the window. In order to detect the foreground, we currently use Mixture of Gaussians approach [6].

VGG: It is a 16 layer Deep Convolutional Neural Network with 1000 output classes having an architecture of layers as [C-C-P-C-C-P-C-C-C-P-C-C-C-P-C-C-C-P-F-F-F], where C-Convolutional, P-Pooling, F-Fully-Connected layers. The details of the VGG, which are out of the scope here, can be found at [1]. We use a pre-trained VGG net to classify the attention windows outputted by the system that were detected from background and foreground. For a given image, VGG gives the class name and its probability after doing a forward pass.

2 Description

A detailed description of the building components – attention window detection and tracking – could be found in earlier reports. In this section, we describe the integration of the three systems, viz. detection, tracking, and classification systems. Also, we discuss about the post-processing of the captured attention windows from each frame of the test video.

		Ground Truth		
		Positive	Negative	Total
SUT Output	Positive	$TP = 145$	$FP = 0$	$TP + FP = 145$
	Negative	$FN = 11$	$TN = 63$	$FN + TN = 74$
	Total	$TP + FN = 156$	$FP + TN = 63$	

Table 1: Confusion Matrix (on video example1.mov with overlapThreshold=0.25)

2.1 Integration of Components

Avoiding foreground detection in PA2 (foreground overlap checking): Since we are using SIFT features to find the attention window, our system finds attention windows from foreground as well as background. However, we only need the background objects, because the foreground objects will anyhow be detected and tracked using background subtraction algorithms like MoG, ViBE, etc. [6] [7] and the tracking algorithms like MOSSE [4]. Therefore, we ensure that a window detected from PA2 is from the static background of the video frame. For each frame we build a union of the foreground rectangles and make sure that no attention window overlap with the union of the foreground rectangles more than a certain threshold(30%).

2.2 Pre-processing Images and Testing with VGG

Detecting a large enough attention window: A lot of the attention windows detected by the our PA2 are very small even after clustering. We made sure that the smallest size of the attention windows is 224x224. We did not upscale the attention windows but grabbed a larger chunk of image from the frames if the original attention window is smaller than 224 in any dimension.

Avoiding Repetation of detected classes: Over the period of the whole video our PA2(attention window detector) can end up detecting same object multiple time. We ensure that a window detected at a particular time frame is not detected again after a specific 'no repetition' period (30 frames) has passed. To that extend, we make sure that our top five strongest(in terms of activation) objects are unique(in terms of the class label).

3 Evaluation

We improved our older system with some logical changes and bug removal. This resulted in improvement (in TP, window-overlap area) with respect to the older evaluation measures.

Re-evaluation of PA2: This metric measures the percentage of frames where an interesting attention window was captured. For our systems the percentage is 34.24% since remaining windows were predominantly detected from the trees.

Re-evaluation of PA3: Using the 'Frame-based Metric' described earlier in PA3 report, the confusion matrix w.r.t. ground truth is shown in table 1. With respect to 'Window-overlap Metric,' the IOR–OOR mean increased from 0.2 & 0.1 to 0.70 & 0.63 for truck0 & truck1 respectively.

3.1 Evaluation of object classification using VGG-net (PA4):

We tested the system on two videos, viz., example1.mov, example2.mp4 both of which are submitted along with the code. However, we have the ground truth only for the first video. Therefore, the following evaluations are only on the first video.

VGG paper reports the top-1 error rate on their test dataset to be 28.07%. We tested the VGG net on the test images produced by our detection and tracking systems. As a ground truth, we hand-labeled the images produced as the attention windows by the two mentioned systems. The testing errors for the second system (tracked attention windows from PA3) was 4.86% and 98.7% for track0 and track1 respectively.

Outputs of classification on PA2: Table 2 lists the top 5 (in terms of activation level) objects classified by the VGG net.

Outputs of classification on PA3: Table 3 lists all the tracks detected by the MOSSE algorithm. The match % indicates the percentage of the frames where classifier output matched with the ground truth (top-1 sense). Moreover, for the second track, VGG classified it as minivan

Output Label	Ground Truth	Activation
barn	barn/house	0.82
flagpole,flagstaff	pole	0.7
pole	pole	0.56
boathouse	house	0.5
thrasher	house	0.47

Table 2: activation levels for top 5 attention windows from PA2

Track Label (%)	Ground Truth	Match %
pickup truck (95.14)	pickup truck	95.14
minivan (76.67)	pickup truck	1.3

Table 3: percent match of tracked windows with ground truth labels

76.67 % of the times as indicated by the value in the parenthesis. However, it is classified as a pickup truck (which is the ground truth) only 1.3% of the times.

4 Conclusion

The results from evaluation suggest that the VGG classification network works significantly better on the tracked windows from the foreground as compared with the stationary windows from the background. The background attention windows, being trees and cropped objects, cause VGG to mis-classify them as something other that the actual ground truth label. While tracking the motion, we are able to easily capture whole objects which enables VGG to easily classify the objects. Therefore, we conclude that the background attention window detection and classification needs improvements to get more accurate results.

References

[1] Simonyan K. and Zisserman A., Very Deep Convolutional Networks for Large-Scale Image Recognition https://arxiv.org/pdf/1409.1556/

[2] Pre-trained VGG http://www.cs.colostate.edu/~cs510/yr2017sp/more_assignments/vgg.zip

[3] Itseez, Open Source Computer Vision Library, 2015, https://github.com/itseez/opencv

[4] David S., Bolme J., Ross Beveridge, Bruce A. Draper, Yui Man Lui, Visual Object Tracking using Adaptive Correlation Filters, Computer Science Department, Colorado State University, Fort Collins, CO 80521, USA

[5] Open Source Computer Vision Library Samples (Python), Steven Puttemans, 2016, https://github.com/opencv/opencv/blob/master/samples/python/mosse.py

[6] Stauffer C., Grimson W., Adaptive background mixture models for real-time tracking, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139

[7] O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. In IEEE Transactions on Image Processing, 20(6):1709-1724, June 2011.

[8] A. Godil, et al., Performance Metrics for Evaluating Object and Human Detection and Tracking Systems, NISTIR 7972, July 2014

[9] F. Bashir, Performance Evaluation of Object Detection & Tracking Systems, CVPR, June '06

[10] http://web.mit.edu/vondrick/vatic/ + https://dbolkensteyn.github.io/vatic.js/