Assignment #2
CS510 2014
"Object Detection"
Due Friday, March 7th

Introduction

In this assignment, you will build object detectors for up to three objects that appear in videos in the ASLAN data set. (You may build more if you wish.) As described below, the choice of objects and techniques is up to you, as long as your detector finds the frames and locations where the object appears, when a new ASLAN video.

This course is meant to prepare you for starting to do research, however, so building object detectors is not enough. You must also write a report that describes how your detectors work and *much more importantly* what you learned from your detectors. How well do they work? When and why do they fail? Are the failures false positives (claiming the object is present when it isn't) or false negatives (failures to detect the object when it is present)? This report must be concise – no more than two pages per object detector. But the quality of the report determines half your grade.

Performance Task

You will write programs that detect at least three objects in images. You may write one program that detects three or more types of objects, or you may write three or more programs, each of which detects one type of object. Your program should read in a video from the ASLAN data set and write out a text file that describes the positions of detected objects. Your program may also take other arguments, so long as you provide a README file that describes how to run your program and so long as you provide the extra data (e.g. templates).

Every line of the output file should have a string followed by five numbers:

```
Object frame UpperLeftX UpperLeftY LowerRightX LowerRightY
```

The object is the type of object (e.g. "bicycle"). It should read as a string, i.e. not contain any spaces. The frame is the frame number (starting with 0) that the object appears in. The next four numbers describe the coordinates of box within the frame that contains the object. The box should be as small as possible and still contain the object (this is called a bounding box). There should be one line of text for every frame in which the object appears, so that if an object appears at frame 50 and disappears at frame 100, your output is 50 lines long. (If two instances of your object appear at the same time, there could be two lines with the same frame number.)

Implementation Hints

There are many ways to design a detector. We have already discussed combining structure tensors with Pearson's correlation to find a template. Before this assignment is due we will also have discussed PCA and MOSSE correlation filters. You might also add an object tracker to "hold on to" detected objects across frames or an image pyramid to help compensate for changes in scale. Pick whatever method(s) you like.

You will quickly discover that most of the work is not in building the detector. Most of the work is in selecting the objects to detect, collecting training and test samples, and evaluating the performance of your detector. The choice of object is critical. Objects or object parts that are flexible or articulated are obviously difficult. Surface markings and specularities also degrade performance. Objects that tend to appear in canonical views are helpful. Objects that are too unusual are of limited value (they won't appear often enough to help you classify videos at the end). Objects that appear in almost all videos are not discriminative. I recommend selecting some objects and then testing and evaluating a detection technique. When you see the results, you may want to pick new objects.

Writing the Report

Half your grade is dependent on the ability of your program to detect objects. The other half depends on the report you write. Your report should do two things: (1) it should briefly describe your approach, so that another person taking this class knows what you did and (2) it should describe what you learned from the successes and failures of this approach on the ASLAN data set. When does your approach work, when does it fail, and why? In general, I would expect your report to include both numbers (how often does it work?) and examples (it works on this case because… it doesn't work on this case because…). The more interesting and well justified your observations are, the higher your grade.

Submission

Your program may be written in either C++ or Python. You will make a linux tar file that contains (1) all your source code, (2) one example of an input video and the corresponding output text file, and (3) a README file that contains exact instructions as to how I should compile and execute your program. If your Python environment depends on environment variables (and most do), those must be included in the README file. You will email the tar file to draper@cs.colostate.edu by no later than midnight on Friday, March 7th.

Restrictions

This is an individual programming assignment. All work must be your own. You may not borrow code from other students, the internet, or any other source.