

Car Finder - Report

Overview

As part of an assignment in CSCI-B657: Computer Vision, I wrote C++ code to detect the number and position of cars in a parking lot. This could be used with satellite imagery to determine the popularity of competing businesses or military strength in an area. It works decently under certain conditions: the size of the cars must be close to an expected ratio of the image, the view must be from relatively over head, and sunny weather is assumed. One strength of my program is rotating images where the cars are angled so that the cars point straight up. The algorithm is based around template matching, so this factor is very important. Negatively, parking lots where the cars point in many different directions are problematic. Perhaps, an expansion of the project would be to segment the parking lots into groups, run the rotation/template matching on each segment, and combine the results back into one image.

How To Use

1) I've tested this on Mac OS X and Linux, and believe it should work on Windows using Cygwin to help you compile. You may have to install LibPNG, or play with the g++ build command depending on the configuration of this library on your machine.

2) Open a terminal and navigate to the CarFinder_CPP/Code folder.

3) Run the g++ command in buildText `g++ carFinder.cpp -o bin/run_cars -lpng -l.`

4) The executable run_cars will be placed in the bin folder

5) Drag run_cars into your terminal window, and then drag any .png image into the terminal

6) Three example images can be found in the images folder

7) Output will be in the bin folder as foundCars.png

Results



