Parking Analysis via Image Processing

CSU Bakersfield
School of Natural Sciences, Mathematics, and Engineering

Joshua Annis, Drake Floyd, Sean Fontes, Miguel Navarrete

Introduction

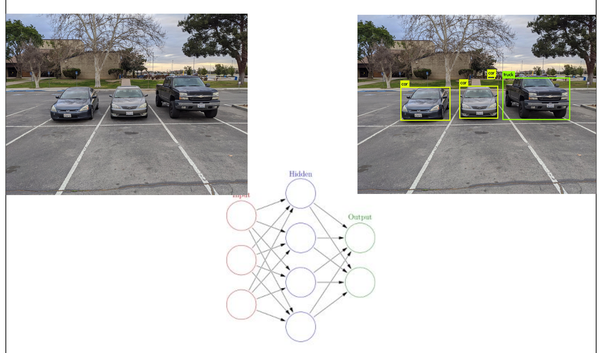
The goal of this project is to determine the viability of the use of image processing to determine parking lot capacity for a given lot or lots at any time through the use of object tracking with input from a still shot image. Furthermore, this project implements an algorithm within the object tracking program to predict imminent parking space availability based on historical data trends of occupancy.

Significance

We aim to significantly reduce the time spent searching for parking by providing an alternative means of locating available spaces. A perquisite of this project would be the reduction of harmful car emissions emanating from the estimated seventeen hours per year per driver spent searching for an available space. Furthermore, the added benefit per capita includes a savings on the maintenance and upkeep of both the roads and vehicles, notwithstanding the added benefit of savings in gas.

Object Detection Background

Object detection is a field of computer vision that involves taking either videos or still images as input and passing it through various different types of algorithms/programs in an attempt to identify and classify various types of objects such as people and cars.



Methodology

In order to determine the viability of detecting vehicles in the context of single board computing, it became readily apparent fairly quickly that the initial approach of using a haar cascade paired with OpenCV was inadequate and produced poor, unusable data. After attempting to refine the initially produced results, the efforts to optimize the haar cascade approach was eventually abandoned as fruitless. Instead, the approach shifted to incorporating a neural network for detection. Specifically, the project is now built around YOLO (You Only Look Once) for use in the scheme for detecting vehicles. Below is a more in depth analysis of the variety of approaches that were explored in the viability testing.

OpenCV—Haar Cascade:

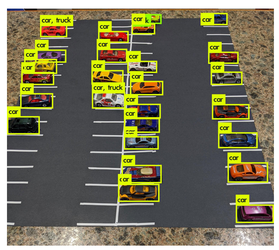
OpenCV offers a relatively simple setup/implementation. This Framework can also be compiled in C, C++, and Python making it very versatile in terms of writing programs that are capable of taking advantage of all of its features. Furthermore, OpenCV supports the use of cascade classifiers, meaning that it is possible to utilize a pre-trained cascade classifier in order to detect objects in any given image. However, this method proved highly ineffective due to its low degree of accuracy, lack of pre-trained models, and the overall inefficiency necessitated a search for alternatives. The results of the initial attempts can be seen below.

Input

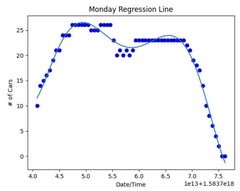
Detections

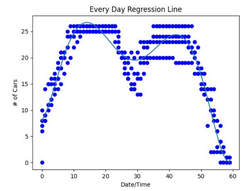
Darknet—YOLO:

YOLO (You Only Look Once) is a convolutional neural net framework developed by Joseph Redmon based in the C programming language. The framework itself boasts a pretrained model capable of searching for and detecting eighty different objects simultaneously in either a still image or a video file. As of now, this is a permanent solution to the problem of detecting model cars as this framework boasts a high degree of accuracy, with little to no manual filtering required. However, as this program is incredibly accurate, so too is it computationally demanding. Due to this, the primary method for vehicle detection for this project, has shifted from a live video input, to a still image input being captured and processed at a regular interval.



Predictions

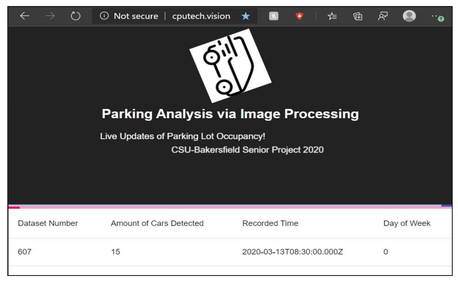




The graphs above illustrate modelling predictions on data representative of a day (left) and of a week (right). Our predictions are conducted through the use of a polynomial regression.

Front-End/ Back-End

Using React.js to create an interactive UI, the front end has a simple design for viewing state from our MySQL database. React was used as a base in the development of the single-page application for viewing the live updates of parking lot space capacities. Every ten minutes the database is called with a REST API using Axios, a promise-based HTTP client for Node.js and React.js, to provide support for request and response interceptors, transformers and auto-conversion to JSON.



Conclusion

In this work we proposed a methodology that can monitor the occupancy rate of a parking lot through the use of a real time object detection neural network. Our results show that it is viable to track occupancy rates in a given parking lot and analyze that data in order to produce a working prediction model.