Car Distance Measuring Using a Dashboard-Mounted Cell Phone

Jakub Kupsik, Computer Science

Mentor: Dr. Yu Sun

Kellogg Honors College Capstone Project

Abstract

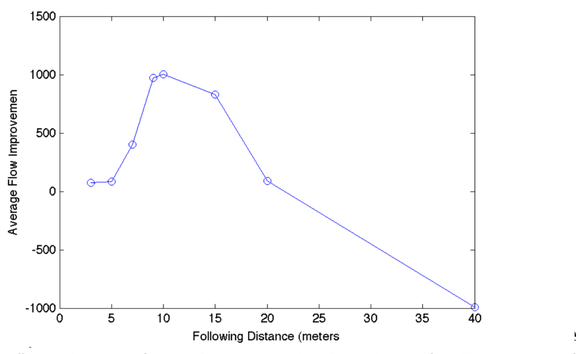
Traffic congestion on the freeways is one of the worst issues in California. What worsens this phenomenon is driving habits: human beings have a tendency to drive closer to the car in front of them as they slow down, which makes merging difficult and exacerbates the slowness. With the proper information and guidance, however, human drivers may be able to correct their behavior and drive better. Such guidance may come from their phones.

This project is a demonstration that a dashboard-mounted cell phone is capable of analyzing road conditions live using its video feed and relaying that information back to the driver. This is done using a real-time object detector model, which is capable of running at a decent frame rate on the mobile device to identify the car in front of the driver. With the video analyzed, the cell phone calculates the relative distance and, over time, relative speed of that car. The cell phone can then display this measurement as well as recommend whether the driver should speed up or slow down to match the speed of the vehicle in front of them.

Literature Review

The concept of leaving a space gap during slowed speed in traffic conditions has been reviewed by many researchers in the field. The following graph from Chacon indicates how vehicle throughput relates to the following distance based upon his simulation, showing the distance of 10 meters as having significant flow improvement over under 5 meters. (1)

Figure 10: Average Flow Improvement versus Minimum Following Distance



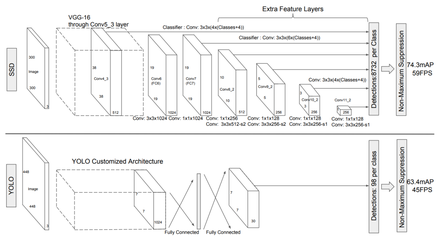
control" system, where two constants K1 and K2 are used to sc

$$a_{ACC} = K_1(g(t) - g_{ACC}) + K_2\Delta v(t)$$

n and relati

Object detection is the computer science problem of having artificial intelligence identify and locate target objects within an image. Object detectors have recently had a revolution, with the emergence of real-time "Single Shot" object detectors. These detectors trade accuracy for speed, and are capable of producing decently accurate identification at incredible speeds. As a result, it is possible to run these object detectors live on cell phones.

The two most popular object detector methods are SSD and YOLO. (3) These methods have the same general function: the image is run through a convolution neural net network which generates which of 10,000 object candidates are present in the image. SSD is more commonly used. (4)



Goal

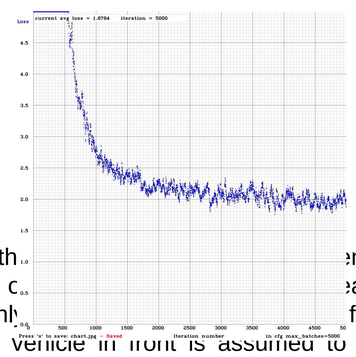
Create a real time mobile application capable of deriving the relative distance and velocity of the vehicle in front of it when mounted on the dashboard of a car.

Methodology

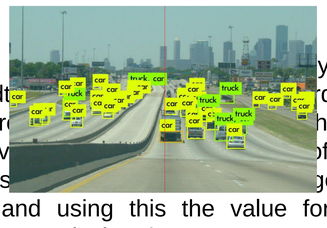
The only useful feature of a cell phone for this purpose is the camera. Therefore, the phone needs to interpret the visual data and identify the target vehicle in front. Object detection is a technology which can be used to solve this problem.

The object detection algorithm SSD-MobileNet was used to produce these results. The algorithm utilized the MobileNet feature extractor(5). It is the most popular choice for mobile applications. It is compact enough to run on a mobile device while being accurate enough to generate useful data. The Tensorflow Android Demo (6) was used as a testbed and template for the core functionality of the project.

The weights for the model were pre-trained on the COCO dataset. (7) The COCO dataset is a popular object detection challenge; other people have already trained the SSD-MobileNet algorithm on this dataset. It contains 80 classes, among which are some of the targets which matter for this project: car and truck. Since the other 78 classes are not needed, however, transfer learning was used to retrain the model to only detect the three objects of importance. Very well labeled dash-cam video was retrieved from Song's "Driving Dataset in the Wild" (8) and converted into the proper format for training. The model was trained for 5000 epochs on the labels for car, van, and truck.



Using th ent size and location of all c easured. The program is onl front. Which vehicle is the vehicle in front is assumed to be the one closest to the center bottom of the picture.



The y regressing the target's widt d roads, vehicles have a narr herefore assumed that every v f 8 feet. A dozen photographs ge car at known distances, and using this the value for the conversion constant K was calculated.

$$d = \frac{w_{apparant}}{w_{actual}} * K$$

Methodology cont.

The distances of previous detections are saved and used to calculate the velocity. The formula used for this purpose is as follows, using the last 10 velocities:

$$\bar{v} = \frac{\sum_{i=0}^{n}\sum_{j=i}^{n}(d_i - d_j)}{n}$$

If a distance varies from another by more that 10%, it is assumed to be an error. If 10 of these errors end up matching, the model assumes a different vehicle is now in front and restarts the averaging process.

Program

The final program contains the following processes:

1. Take video using the phone camera
2. Convert, downscale, crop the video stream
3. Interpret the picture using object detection
4. Locate the car directly in front
5. Regress the apparent size to relative distance
6. Compare to previous to obtain relative velocity
7. Display and record the relative velocity to the user

Conclusion

This project shows that it is possible to have a cell phone measure the distance and velocity of a car using only visual input.

The output data is prone to frequent errors. This is in part to the inaccuracy of the mobile object detector. Additional training is required to increase the accuracy.

There are several limitations to this project which can be fixed. The biggest one is the assumption that the car in front is centered in the image; this only holds true for straight roads. There are two additional features which would allow the application to correct for curved roads: lane detection, and car angle detection.

There is also the limitation that other road vehicles such as motorcycles are not included. As long as the model is trained to detect them and there is a separate distance regression, there is no reason why this approach shouldn't work on motorcycles.

A more difficult problem involves measuring the absolute speed of the vehicle. This might be possible using the parallax of the environment, assuming it is static, but it would need to be done using a video-specific computer vision technique.

References

1. Chacon, Scott Henry - The Effect of Driver Behavior on Freeway Traffic Flow, Princeton University, https://orfe.princeton.edu/~alaink/Papers/driver_behavior_Chacon_TRB10.pdf
2. Kerner, Borris S. - *Effect of Driver Behavior on Spatiotemporal Congested Traffic Patterns at Highway Bottlenecks in the Framework of Three-Phase Traffic Theory https://arxiv.org/pdf/1012.5159.pdf*
3. Redmon, Joseph; Farhadi, Ali - YOLOv3: An Incremental Improvement https://pjreddie.com/media/files/papers/YOLOv3.pdf
4. Lui, Wei et al. SSD: Single Shot MultiBox Detector https://arxiv.org/abs/1512.02325
5. Howard, Andrew G. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications https://arxiv.org/abs/1704.04861
6. Tensorflow Android Demo - https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android
7. COCO Dataset http://cocodataset.org/#home
8. Song, Byung Gon Driving dataset in the wild: Various driving scenes https://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-92.pdf