Traffic Anomalies Detection with A Semi-supervised Approach

Daniel Shumaker
Michigan State University
shumak37@msu.edu

Luan Nguyen
Michigan State University
nguye590@msu.edu

Pengyu Chu
Michigan State University
chupengy@msu.edu

1. Introduction

With sensors tracking traffic, signaling systems, and infrastructure, our transportation systems are becoming smarter. With computer vision and deep learning, there are many opportunities to solve real-world problems with data gained from multi-cameras. Solving these problems can lead to safer cities and can save many lives. We propose a solution to one of these problems: detecting traffic anomalies. Examples of these anomalies are lane violations, illegal U-turns, wrong direction driving, crashes, and stalled vehicles. These can be extremely dangerous pDTarticularly in traffic highways and intersections. The potential solutions will get the humans in the loop to pay attention to meaningful visual information in situations where timely intervention can save lives.[1]

Unfortunately, progress has been limited for several reasons like missing data labels, and the lack of high-quality models that convert data to decent forms. In other words, it is hard to get data with labels from the real world. Due to the lack of labels, anomalies cannot be classified with current algorithms. Thus, we plan to seek a semi-supervised approach to address this problem and focus on the research and development of techniques that rely more on transfer learning and semi-supervised learning.

With some pre-trained model like VGG 16 [2], we can leverage transfer learning to combine these models with our approach and detect the abnormal traffic behaviors from traffic camera video data which is provided by NVIDIA corporation.

2. General Plan

- Look through relevant videos from traffic cameras filmed on highways and intersections in the training and test data sets
- Get object detection algorithms related to traffic environment
- Detect static objects in the background and moving vehicles, then compare different status of objects to sense abnormal actions
- Implement the algorithm on the training data, with some popular frameworks like Pytorch or Tensorflow
- Compare our findings to the benchmark algorithm on the test data set. We could also try to improve the accuracy and detection speed
- Write a paper using our findings

First, with the help of [**?**], we will detect static objects by comparing pixel values in an image. The pixels that are similar in color and intensity will become black, the other pixels, the ones on the border of objects, will become white. That will give us contours for the objects in the frame. We will compare these contours with a few more ones generated from other pictures in the given video. The objects that remain will be our stationary objects.

Building off of that, we will detect the lanes of the road. Scrolling through pictures, we will find paint on the road, and we will project a line along that paint to account for one lane. Then, the tasks will be detecting the moving objects. This will have a similar procedure to the previous steps. We will get these objects from the boundaries of what is left on the screen after the static objects have been discovered. Now there has been several state-of-the-art solutions like YOLOv3[**?**] which address the objects detection problems regardless of their state. Benefiting by one-stage detection, these solutions can be real-time with a fair accuracy.

By means of [**?**], we will get the status of each of the classified moving objects. This includes the car's origin: the central part of the object, the car's speed: the distance the origin travels in a frame, the car's direction: direction the origin moves in a frame, and the surrounding space: distance between two origins in the same frame.

In the last step, we will implement our learning algorithms to detect anomalies, compare our findings with a benchmark, and we will discuss our findings in a report.

3. Timeline

- Week of Feb 18: literature review
- Week of Feb 25: data preprocessing
- Week of Mar 4: develop baseline
- Week of Mar 18: finish developing version 1
- Week of Mar 25: improve the model, make version 2.0
- Week of Apr 8: begin to write report

References

[1] M. Naphade, D. C. Anastasiu, A. Sharma, V. Jagrlamudi, H. Jeon, K. Liu, M.-C. Chang, S. Lyu, and Z. Gao. The nvidia ai city challenge. *IEEE Smart-World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smart-World/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2017.

[2] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.