A USDOT NATIONAL UNIVERSITY TRANSPORTATION CENTER

The Intelligent Mobility Meter - Portable Fine-Grained Data Collection and Analysis of Pedestrian, Cyclist, and Motor Vehicle Traffic

Final Research Report

Principal Investigator: Bernardo R. Pires

Student Collaborators: Shreyank Ranganath, Reid Yesson, Dongyoon Kim, Tarek Sahyoun

Contract # 69A3551747111

Carnegie Mellon University
The Robotics Institute

The Intelligent Mobility Meter – Portable Fine-Grained Data Collection and Analysis of Pedestrian, Cyclist, and Motor Vehicle Traffic

Final Report – April 23, 2019

Principal Investigator: Bernardo R. Pires

ORCID ID: https://orcid.org/0000-0003-0591-4250

Student Collaborators: Shreyank Ranganath, Reid Yesson, Dongyoon Kim, Tarek Sahyoun

DISCLAIMER

1. Project Description

The Intelligent Mobility Meter (IMM) is a portable data collection and analysis platform which will be able to collect fine-grained statistics on pedestrian, cyclist and vehicular traffic. The objective of the IMM project is to provide accurate and actionable data to government officials and transit advocates. The meter is an expansion of the UTC T-SET funded project “Automatic Counting of Pedestrians and Cyclists”, which researched, developed and deployed a robust pedestrian and bike counting system. The deployed system was employed - in collaboration with the City of Pittsburgh - to collect usage data on Pittsburgh’s bike lanes. However current system capabilities are limited to counting pedestrians and bikes. To create a true mobility meter, this project expanded the analysis

2. Introduction

Usage statistics have the capability to inform policy makers and transportation advocates on the best design for the infrastructure of the future. They also have the potential to help the current traffic engineers identify and resolve infrastructure problems. However, it is not feasible to place data collection devices everywhere. The Intelligent Mobility Meter (IMM) is a portable device that has the capability to collect fine-grained statistics on the behavior of all road participants in any key area.

The IMM grew from a need to obtain statistics on usage of bike lanes, however it has now grown beyond that niche need to collect data for all road participants. Statistics about pedestrians, motor and non-motor vehicles provide important information for government officials to build safe infrastructures for walking, biking and driving. In addition to pedestrian and bicyclist data, having information about the number of motor vehicles can give key

3. Objectives

For purposes of obtaining the counts of objects moving in a given direction at a location, the problem statement is broken down into three different technical objectives:

- Generate consistent detections of cars, pedestrians and bicycles in each frame of the video as they move through the field of view of the camera.
- Once detected, accurately track each of these objects - cars, bikes and pedestrians by assigning a unique ID for each object as soon as the objects appear in the field of view of the camera.
- Use all the tracks obtained by the detected objects to determine the total counts of the objects passing through a location in different directions.

4. Previously Developed System

On the previous UTC T-SET funded project "Automatic Counting of Pedestrians and Cyclists", we developed robust vision-based pedestrian and cyclist counting system (Figure 1). This system consists of data collection hardware prototype system and an accurate computer vision based counting system. The presented method can work under different lighting and weather conditions. Approximately 50 hours of data was collected in different locations around CMU campus using the prototype. In order to label the pedestrians and the bikes in the recorded data, a web-based object labeling software was implemented. Unlike the existing object labeling tools in the images, our developed labeling tool includes novel properties to reduce the labeling time by incorporating spatial and geometric constraints between the frames of the videos. We labeled 10 hours of our recorded dataset (541 pedestrians and 111 cyclists) using this tool to train and test our counting method.

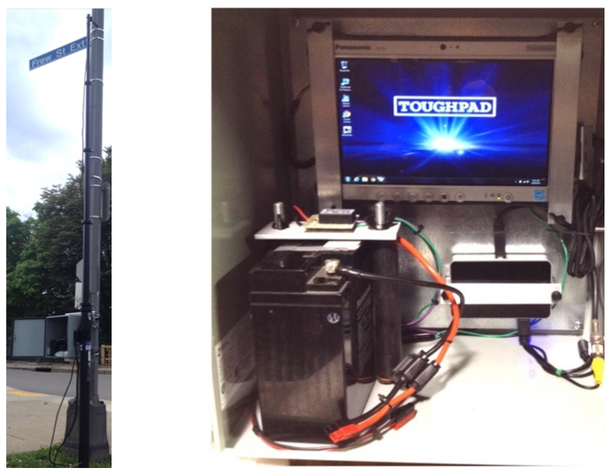


FIGURE 1. Previously developed hardware prototype.

4.1. Detection

The previously developed system used the Faster R-CNN algorithm [1], which is one of the popular object detection algorithms at present. Its popularity stems from being one of the first detectors to get accurate detections while processing ~ 7 - 10 frames per second (fps), a great achievement towards making real-time object detection. Despite its popularity, the Faster R-CNN has quite a complex architecture involving many independent stages in its detection pipeline (see paper for details). The complexity in its architecture makes it a challenge to train the network as it can become difficult to isolate exactly at which stage and why the training process fails when setting up the training of the network.

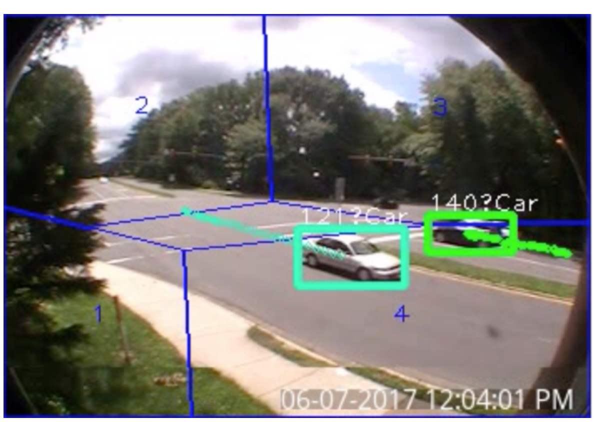
The Faster R-CNN implementation we used was pre-trained on the VOC dataset (2007 and 2012) [2], which spans images of 20 different classes including bikes, cars, pedestrians, which we aim to detect. Though performing reasonably on the video data we obtained, inability to train the network on custom data was putting the network at a disadvantage. Although, we were focused on detecting bikes, pedestrians and cars (all of which are included in the classes of the VOC dataset), our camera angle was different (slightly aerial), giving a different view of the pedestrian, bikes and cars, than the "direct" viewing angle of the images present in the VOC dataset, which the Faster R-CNN was pre-trained on. As we further refine our algorithms, it became apparent that to improve detection accuracy and get consistent detections on our videos, we would need to train the network on our own images created from the videos we were provided. Due to the training limitations mentioned above, a decision was made to move to a novel and more recent detection methodology.

4.2 Tracking

The tracker used on the project was adapted from [3]. This tracker uses many different metrics like appearance similarity between objects, distance between bounding boxes of consecutive detections etc. to track a detected object as it moves across the field of view of the camera (consecutive frames). It is robust to occlusions and works as expected. Getting early and consistent detections for each frame would allow the tracker to assign a track ID to an object early and track it better as it moved “through” consecutive frames.

4.3 Counting

The counting strategy previously used was to estimate the start and end point of an object’s track and then fit the track using a polynomial curve fitting strategy. This fitted (extended) track allowed the algorithm to determine the start and end regions traversed by the object. The counting regions were drawn on the video as “masks” based on the different directions available to be traversed in a location (see Figure 2). The extended version of the track allowed us to get an estimate of the regions traversed by an object even in cases where the tracks were too short or when the tracks were not picked up early enough to “belong” to a start region. With this information we would update the total counts of the number of



	1	2	3	4
1:	0	0	7	7
2:	0	0	13	25
3:	0	5	0	0
4:	5	22	0	0

FIGURE 2. Detections and Masks (Note Car 121 – light green track).

Figure 2 highlights the masks drawn representing different regions (1, 2, 3, 4) in a location. Car 121 traversed regions 2 to 4 so the count of 2 to 4 was incremented from 25 to 26 (Visible in the count matrix adjacent to the video going up from 25 to 26 for 2 to 4). In Figure 2, car 121 (121 is the unique ID assigned by the tracker to the car) is detected early enough to "start" from region 2. But often, the detector detects objects a few frames later than required. This can make the "start" region of the track uncertain because the start point of the object becomes ambiguous (i.e. if car 121's start point was outside 2, we would not be able to determine its start point). To solve this problem, the polynomial curve fitting strategy was used to predict an extended track based on the shape of the entire track. Figure

	1	2	3	4
1:	0	0	7	7
2:	0	0	13	26
3:	0	5	0	0
4:	5	22	0	0

FIGURE 3. Polynomial curve fit on the tracks of 121 (Red extensions at start and end)

For tracks having a simple curve, the polynomial curve fitting strategy works well. However, for tracks representing objects traversing a complex curved path, the polynomial curve fitting strategy often miscalculated the fit and resulted in count increments for wrong regions. One example of this is shown in Figure 4, in which car 156 travels from region 3 to region 4 (as seen by its track) whereas the polynomial curve fit however wrongly counts it as travelling from region 1 to region 4 (see the red extensions wrongly applied to the track in Figure 4).

	1	2	3	4
1:	0	0	10	8
2:	0	0	15	32
3:	0	5	0	0
4:	8	32	0	0

FIGURE 4. Limitations of the polynomial curve fitting strategy.

5. Novel IMM System

The novel IMM system was developed specifically to address the limitations of the detector and the counting technique used in the previous implementation. Specifically, the Faster R-CNN algorithm was replaced with a newer detection algorithm, SSD, and a new

5.1 Detection

The Faster R-CNN algorithm was replaced with SSD: Single Shot MultiBox Detector [4], which proposes a much simpler network architecture, and avoids the complexity of having multiple different parts in the network architecture. The entire image is input to the network, and the localization and detection of the objects happens simultaneously in a single pass (hence the name “Single Shot”). In other words, both localization and detections of objects happen in a single forward pass through the network. The algorithm outputs the detection class and bounding box coordinates of all the detected objects in a frame.

In Faster R-CNN the localization and detection happen separately at different stages making it more complex that SSD. SSD gives comparable performance (better in some cases) to the Faster R-CNN while taking away the complexity of the Faster R-CNN (See [4]). The simple architecture of SSD addressed the training limitation that we faced earlier, and we were able to train the detector on our custom data easily. An example of the result comparing the counts output by the two algorithms – SSD and Faster R-CNN are included below for different regions in a location. The true counts (counted manually) are also included to compare the performance of the two algorithms to the ground truth.

In Figure 5 below:

- TOTALS represent the true counts (counted manually).
- FRCNN represents the Faster R-CNN network trained on just the VOC dataset (weights were provided in the implementation used).
- SSD represents the SSD network trained on just the VOC dataset (weights were provided in the implementation used).
- SSD_DATA represents the performance of SSD network trained on VOC dataset and on our custom data of around 400 images.

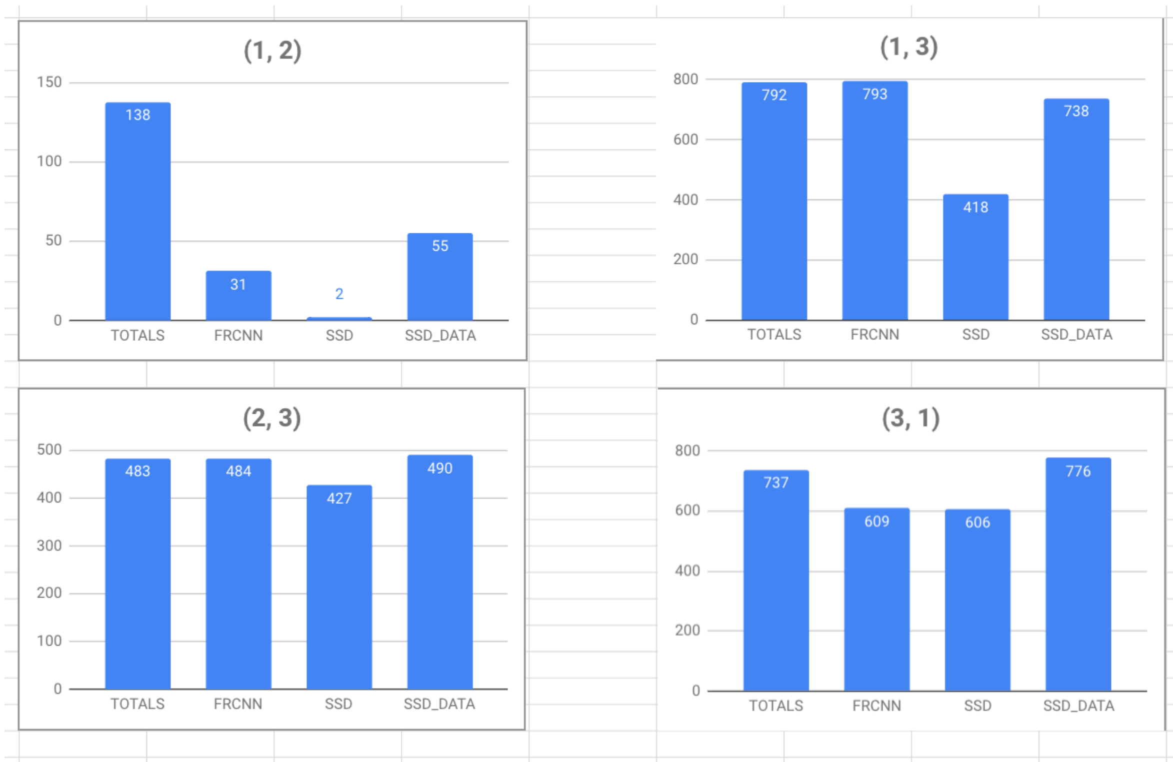


FIGURE 5. Performance comparison of Faster R-CNN and SSD on our data.

As is evident from the graphs in Figure 5, training the SSD network on our data helped the SSD perform better (compared to when untrained on our data) and provided comparable performance to the Faster R-CNN. With more training, the performance of SSD could

5.2 Counting

The new counting technique utilizes the shape of the tracks in a different way compared to the previous polynomial curve fitting technique to determine the object counts. The details of this technique are discussed below:

1. A few *representative* tracks are chosen and used as the standard to represent traversal of an object from point A to point B. For example, in the image below, the track of car 156 is one of the longest detected tracks for cars moving from region 3 to region 4, so it is one of the tracks that are used to represent tracks moving from 3 to 4.

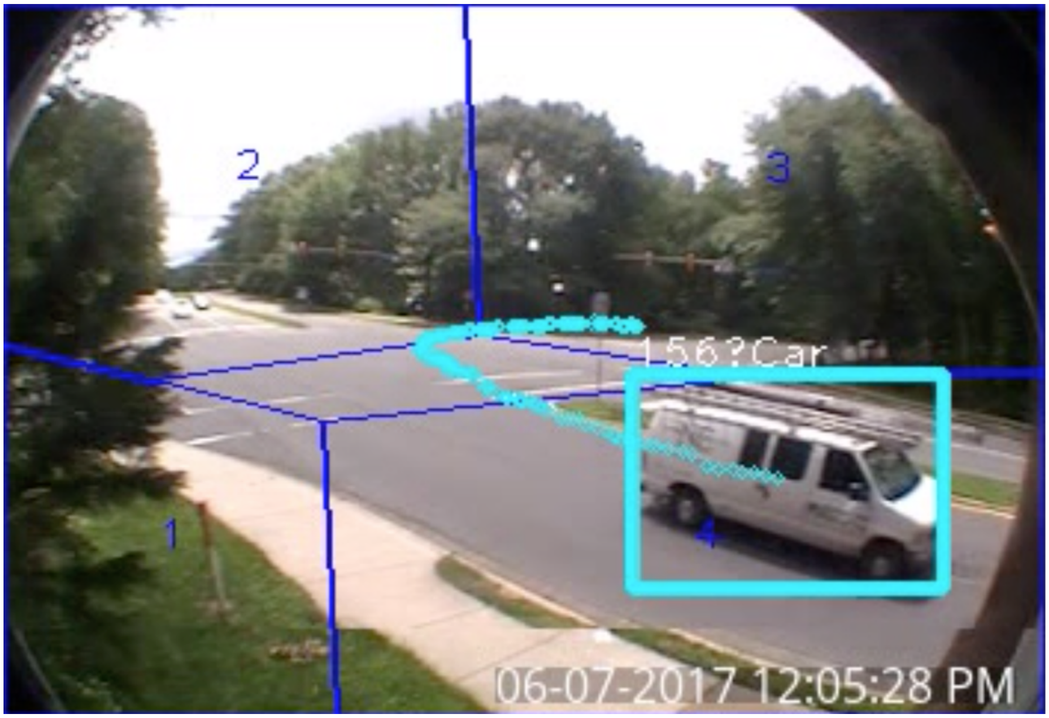


FIGURE 6. "Ideal" track for region 3 to 4.

2. Three to five such *representative* tracks are selected to represent the traversal of an object for each combination of different start-end regions. Each point in the track has a (x, y) value.
3. The tracks for all other objects, named *candidate* tracks, should now be classified into one of the different start-end combinations. This is done by comparing each track to one of the *representative* tracks.
4. This comparison is done by using a binary occupancy matrix where each entry corresponds to a portion of the image. On this matrix, a 1 value represents an image area that was traversed during the track, whereas a 0 value represents an area that was not traversed.
5. The binary occupancy matrix of each *candidate* track is matched to the *representative* track that has the closest occupancy matrix (as measured by the number of presence entries, i.e. 1s, at the same location in both matrices.in the binary occupancy matrix which lowest number of different entries in the binary matrix).

Using this new counting technique, we solve the problem faced with the previously used polynomial curve fitting technique, which wasn't very accurate for tracks having complex curves. With the new technique, the complexity of the curve of the track does not affect our ability to predict which start-end region a given track belongs to.

6. Counted Data

As part of this project, a large body of data was counted for the Virginia Department of Transportation (VDOT). Using the methods described above, the team counted motor vehicle, cyclist and pedestrian traffic for approximately **140 hours of video** collected by the Department. To ensure data accuracy the videos where counted automatically and then verified manually using spot checks. This large-scale effort contributes to a project by the VDOT to assess the effects of road diets on the traffic patterns. (VDOT point of contact: Peter

7. Conclusions and Future Work

This project allowed us to significantly expand the technical capabilities of our previous counting projects and create a true mobility meter than can identify, track and count all road users. The large-scale data counting conducted for the Virginia Department of Transportation demonstrated that the IMM is mature enough to tackle difficult real-world counting projects.

References

[1] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Shaoqing Ren and Kaiming He and Ross B. Girshick and Jian Sun, IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 39, pp. 1137-1149, 2015.

[2] The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, Everingham, M. and Van~Gool, L. and Williams, C. K. I. and Winn, J. and Zisserman, A., http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[3] Robust Online Multi-object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning, Seung-Hwan Bae and Kuk-Jin Yoon, 2014 IEEE Conference on Computer Vision and Pattern Recognition, 23-28 June 2014.

[4] SSD: Single Shot MultiBox Detector, Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, Alexander C., Lecture Notes in Computer Science, Springer International Publishing, pp. 21-37, 2016.