Mobility21

A USDOT NATIONAL
UNIVERSITY TRANSPORTATION CENTER

Carnegie Mellon University

Penn
UNIVERSITY of PENNSYLVANIA

THE OHIO STATE UNIVERSITY

CCAC

Improving rush hour traffic flow by computer-vision-based parking detection and regulations

Christoph Mertz (PI, ORCID: 0000-0001-7540-5211), Sean Qian (Co-PI),
Justin Chiang

FINAL RESEARCH REPORT

Contract # 69A3551747111

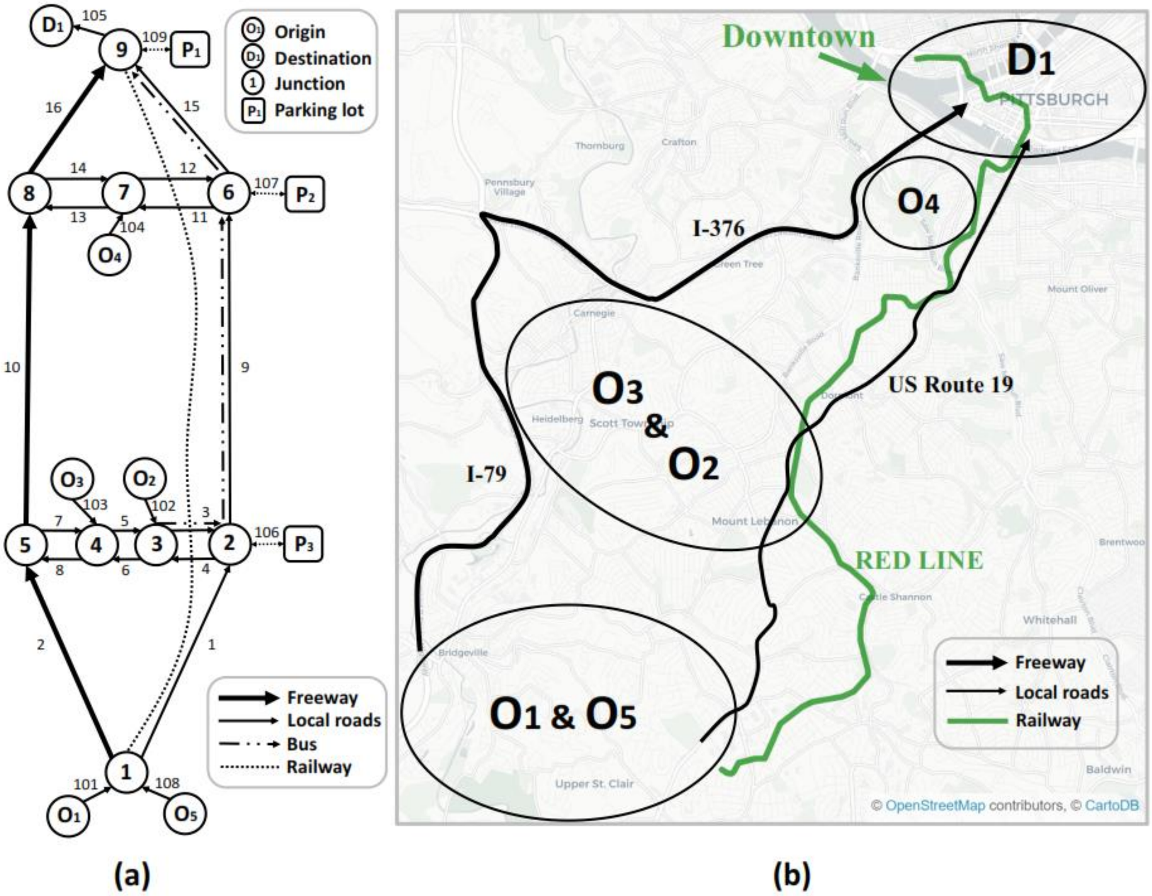
DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Introduction

To optimize our transportation network through a detailed analysis of all the traffic flows one needs to have two main ingredients. First it is necessary to have a holistic modeling framework that can manage the complexity of this multi-modal system. Second is comprehensive data for all the traffic statistics. In this project we want to demonstrate with an example how parking information can be incorporated into a multi-modal dynamic user equilibrium (MMDUE) model and develop a computer vision tool that can provide street parking data. The details of the MMDUE can be found in the paper [1], in this report we only describe the example.

Multimodal Network



© OpenStreetMap contributors, © CartoDB

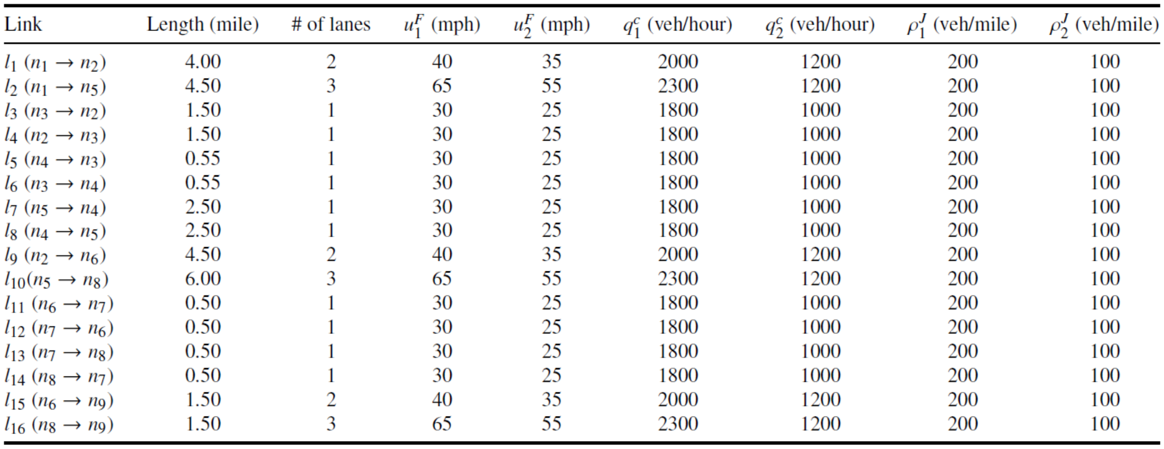
Figure 1 (a) A simplified multi-modal network for downtown and southern Pittsburgh region used in the experiments. Note the bus line represents the aggregation of all high-frequency bus routes in this area. Nodes and links are labeled on the figure. (b) The real network map. Origins 1 to 5 are all large residential zones. Downtown (D1) is the only destination.

The multi-modal dynamic user equilibrium (MMDUE) model is tested in the network shown in Figure 1. The network is an abstraction of the real multi-modal network in the Pittsburgh region, covering the district of Upper St Clair, Bridgeville, Mt Lebanon, Scott Township, Carnegie, Green Tree, Banksville, Mt Washington and Downtown Pittsburgh, with I-79, I-376, US Route 19 and a light rail Red Line connecting the suburbs and the downtown, see Figure 1(b). The simplified network is shown in Figure 1(a), with five origins $O_1 – O_5$, one destination D_1, nine nodes (junctions), 16 links (road segments), three parking areas, one bus route, and one railway line. The parking area can be viewed as the aggregation of all parking space in that area, and we only considered one bus route that represents the aggregation of all high-frequency routes and one railway line. We also assume all travelers own or have access to a private car, which means. This allows all travelers to choose any travel modes in this experiment.

We have made efforts on calibrating the traffic flows and OD demand in the Pittsburgh metropolitan area in a separate project [2][3][4], so the OD demands were employed in this study. The total passenger demand is set as 30,000 for the morning peak hours (5 AM to 9 AM) in this area, from all origins $O_1 – O_5$ to the only destination D_1. Note origins $O_1 – O_5$ are all large residential districts, and the only destination D_1 is downtown Pittsburgh. All link parameters are listed in Table 1 and all modes and paths are listed in Table 2. O_2 is the bus origin and only for releasing buses to the network. O_5 is the truck origin and only for releasing trucks to the network. All other parameters used in the experiment are listed in

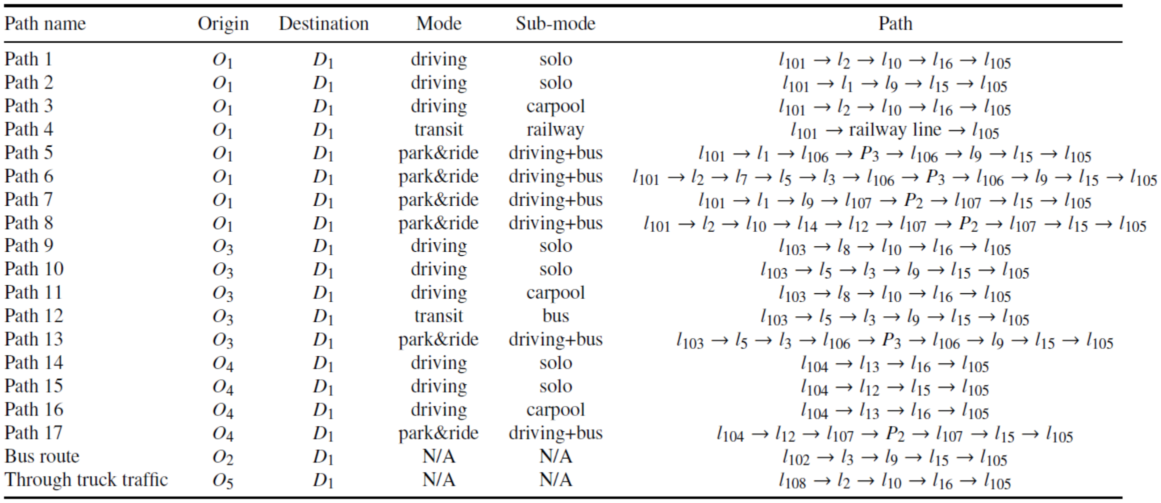
Table 3.

Table 1 Link parameters. u^F: free-flow speed, q^c: lane capacity, ρ^J : jam density. Subscript 1: passenger cars, 2: buses and trucks.



Link	Length (mile)	# of lanes	u_1^F (mph)	u_2^F (mph)	q_1^c (veh/hour)	q_2^c (veh/hour)	ρ_1^J (veh/mile)	ρ_2^J (veh/mile)
l_1 ($n_1 \rightarrow n_2$)	4.00	2	40	35	2000	1200	200	100
l_2 ($n_1 \rightarrow n_5$)	4.50	3	65	55	2300	1200	200	100
l_3 ($n_3 \rightarrow n_2$)	1.50	1	30	25	1800	1000	200	100
l_4 ($n_2 \rightarrow n_3$)	1.50	1	30	25	1800	1000	200	100
l_5 ($n_4 \rightarrow n_3$)	0.55	1	30	25	1800	1000	200	100
l_6 ($n_3 \rightarrow n_4$)	0.55	1	30	25	1800	1000	200	100
l_7 ($n_5 \rightarrow n_4$)	2.50	1	30	25	1800	1000	200	100
l_8 ($n_4 \rightarrow n_5$)	2.50	1	30	25	1800	1000	200	100
l_9 ($n_2 \rightarrow n_6$)	4.50	2	40	35	2000	1200	200	100
l_{10}($n_5 \rightarrow n_8$)	6.00	3	65	55	2300	1200	200	100
l_{11} ($n_6 \rightarrow n_7$)	0.50	1	30	25	1800	1000	200	100
l_{12} ($n_7 \rightarrow n_6$)	0.50	1	30	25	1800	1000	200	100
l_{13} ($n_7 \rightarrow n_8$)	0.50	1	30	25	1800	1000	200	100
l_{14} ($n_8 \rightarrow n_7$)	0.50	1	30	25	1800	1000	200	100
l_{15} ($n_6 \rightarrow n_9$)	1.50	2	40	35	2000	1200	200	100
l_{16} ($n_8 \rightarrow n_9$)	1.50	3	65	55	2300	1200	200	100

Table 2 All modes and paths in the numerical experiment.



Path name	Origin	Destination	Mode	Sub-mode	Path
Path 1	O_1	D_1	driving	solo	$l_{101} \rightarrow l_2 \rightarrow l_{10} \rightarrow l_{16} \rightarrow l_{105}$
Path 2	O_1	D_1	driving	solo	$l_{101} \rightarrow l_1 \rightarrow l_9 \rightarrow l_{15} \rightarrow l_{105}$
Path 3	O_1	D_1	driving	carpool	$l_{101} \rightarrow l_2 \rightarrow l_{10} \rightarrow l_{16} \rightarrow l_{105}$
Path 4	O_1	D_1	transit	railway	$l_{101} \rightarrow$ railway line $\rightarrow l_{105}$
Path 5	O_1	D_1	park&ride	driving+bus	$l_{101} \rightarrow l_1 \rightarrow l_{106} \rightarrow P_3 \rightarrow l_{106} \rightarrow l_9 \rightarrow l_{15} \rightarrow l_{105}$
Path 6	O_1	D_1	park&ride	driving+bus	$l_{101} \rightarrow l_2 \rightarrow l_7 \rightarrow l_5 \rightarrow l_3 \rightarrow l_{106} \rightarrow P_3 \rightarrow l_{106} \rightarrow l_9 \rightarrow l_{15} \rightarrow l_{105}$
Path 7	O_1	D_1	park&ride	driving+bus	$l_{101} \rightarrow l_1 \rightarrow l_9 \rightarrow l_{107} \rightarrow P_2 \rightarrow l_{107} \rightarrow l_{15} \rightarrow l_{105}$
Path 8	O_1	D_1	park&ride	driving+bus	$l_{101} \rightarrow l_2 \rightarrow l_{10} \rightarrow l_{14} \rightarrow l_{12} \rightarrow l_{107} \rightarrow P_2 \rightarrow l_{107} \rightarrow l_{15} \rightarrow l_{105}$
Path 9	O_3	D_1	driving	solo	$l_{103} \rightarrow l_8 \rightarrow l_{10} \rightarrow l_{16} \rightarrow l_{105}$
Path 10	O_3	D_1	driving	solo	$l_{103} \rightarrow l_5 \rightarrow l_3 \rightarrow l_9 \rightarrow l_{15} \rightarrow l_{105}$
Path 11	O_3	D_1	driving	carpool	$l_{103} \rightarrow l_8 \rightarrow l_{10} \rightarrow l_{16} \rightarrow l_{105}$
Path 12	O_3	D_1	transit	bus	$l_{103} \rightarrow l_5 \rightarrow l_3 \rightarrow l_9 \rightarrow l_{15} \rightarrow l_{105}$
Path 13	O_3	D_1	park&ride	driving+bus	$l_{103} \rightarrow l_5 \rightarrow l_3 \rightarrow l_{106} \rightarrow P_3 \rightarrow l_{106} \rightarrow l_9 \rightarrow l_{15} \rightarrow l_{105}$
Path 14	O_4	D_1	driving	solo	$l_{104} \rightarrow l_{13} \rightarrow l_{16} \rightarrow l_{105}$
Path 15	O_4	D_1	driving	solo	$l_{104} \rightarrow l_{12} \rightarrow l_{15} \rightarrow l_{105}$
Path 16	O_4	D_1	driving	carpool	$l_{104} \rightarrow l_{13} \rightarrow l_{16} \rightarrow l_{105}$
Path 17	O_4	D_1	park&ride	driving+bus	$l_{104} \rightarrow l_{12} \rightarrow l_{107} \rightarrow P_2 \rightarrow l_{107} \rightarrow l_{15} \rightarrow l_{105}$
Bus route	O_2	D_1	N/A	N/A	$l_{102} \rightarrow l_3 \rightarrow l_9 \rightarrow l_{15} \rightarrow l_{105}$
Through truck traffic	O_5	D_1	N/A	N/A	$l_{108} \rightarrow l_2 \rightarrow l_{10} \rightarrow l_{16} \rightarrow l_{105}$

Table 3 All other model parameters

Name	Values				
Morning commute model	$\alpha = \$6.4/\text{hour}$	$\beta = \$3.9/\text{hour}$	$\gamma = \$15.2/\text{hour}$	$t^* = 9\text{AM}$	
Logit model	$\alpha^1, \alpha^1_{g(1)\in G(1)} = 1.5$	$\alpha^2, \alpha^2_{g(2)\in G(2)} = 1.0$	$\alpha^4, \alpha^4_{g(4)\in G(4)} = 2.0$	$\beta_1 = 1.0$	$\beta_2^{m\in\{1,2,4\}} = 1.0$
Parking-1 (CBD)	$p_1 = \$10$	$\epsilon_1 = 2$ min	$E_1 = 10000$		
Parking-2	$p_2 = \$3$	$\epsilon_2 = 1$ min	$E_2 = 20000$		
Parking-3	$p_3 = \$3$	$\epsilon_3 = 1$ min	$E_3 = 20000$		
Bus	$\delta^{rs}_k = \$2.75$	$\sigma^{rs}_{k,t} = 0$	frequency = 15 min	waiting = 7.5 min	
Railway	$\delta^{rs}_k = \$3.75$	$\sigma^{rs}_{k,t} = 0$	frequency = 12 min	waiting = 6.0 min	full trip = 40.0 min
Carpooling	$\Delta^{rs}_{k,t}(1) = 0$	$\Delta^{rs}_{k,t}(2) = \$1$			
Dynamic network loading	unit time = $5s$	# of intervals = 2880	start time = 5AM	end time = 9AM	

Convergence

We solve the MMDUE with the solution framework presented in [SEAN-REFERENCE SECTION]. Changes in the equilibrium gap against the number of iterations using our proposed solution and quadratic solver are presented in Figure 2. All the experiments below are conducted on a desktop with Intel Core i7-6700K CPU @ 4.00GHz x 8, 2133 MHz 2 x 16GB RAM, 500GB SSD. We run our algorithm for 100 iterations, and the proposed method outperforms the quadratic optimization method in terms of the equilibrium gap. It takes the proposed method 211s to complete 100 iterations, comparing to the 227s taken by the quadratic solver.

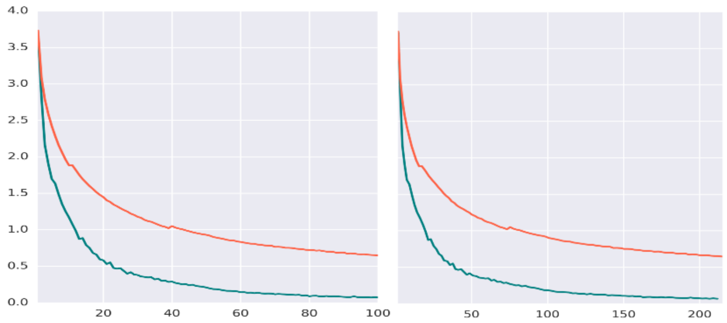


Figure 2 Convergence curves for both gradient projection methods.

Solution of network flow and mode choices

The path choice probability over the morning peak for OD pair (1, 1) and OD pair (3, 1) is presented in Figure 3. For OD pair (1,1), most travelers departing in 5 AM-6 AM choose carpool and railway to commute, and more travelers select park-and-ride (PnR) after 7 AM. The generalized cost of metro is relatively low, as a result of low railway fare. After 8 AM, taking railway will incur a high late arrival cost penalty, so the path flow of railway decreases dramatically. Using carpool directly to CBD is another preferred choice before 8 AM since the roads to CBD are not congested yet. After 8 AM, cruising time for parking in the CBD area increases, leading to a decline in the carpool probability. In contrast, the generalized park-and-ride cost is low when the CBD is congested because of the savings on travel time and parking fee. Hence park-and-ride mode becomes the most preferable choice after 8 AM. At 7 AM, the flow of paths 7 and 8 has a spike, possibly due to the unstable UE solution when the cost of path 7 and 8 coincide. The stability of MMDUE can be improved in future research by the method proposed by [5][6]. Similar choice patterns can be observed for OD pair (3,1). One observation is that there are few travelers choosing solo-driving all the way to downtown, which is probably attributed to the high parking fee in the downtown, relatively low inconvenience cost for carpool, and the way transit and park-and-ride are provided in this experiment.

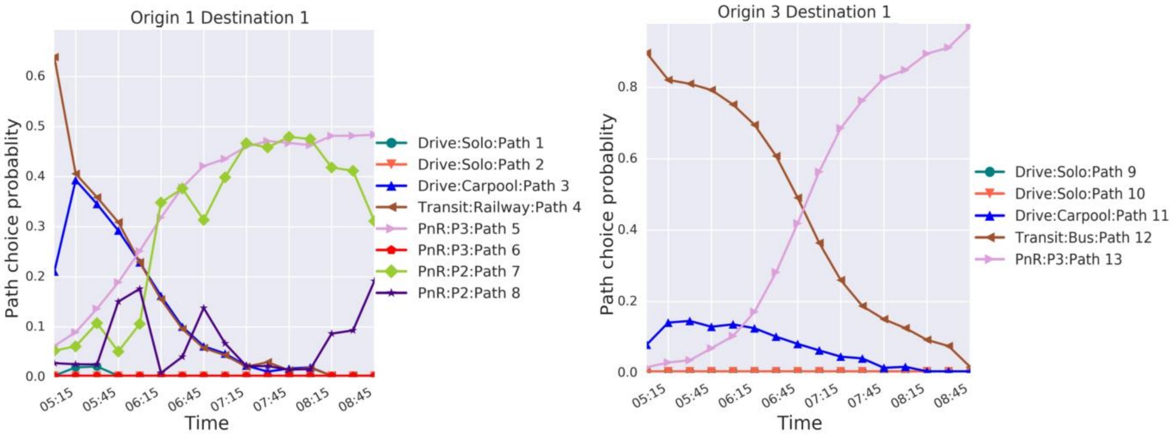


Figure 3 Time-varying path choice for OD pair (1,1) and (3,1).

Sensitivity analysis on total demand level

We change the total demand from 75% to 125% of the O-D demand in the original problem settings, keeping other parameters the same. We solve the MMDUE under different demand levels and plot the average user cost for different modes and the route choice for OD pair (1,1) in Figure 4.

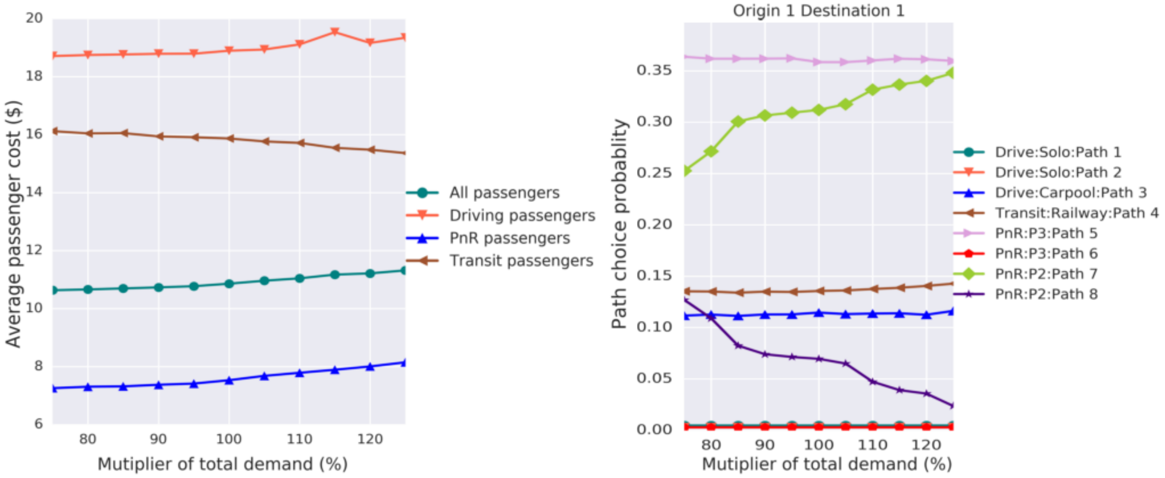


Figure 4 The influence of total demand on average cost per traveler and the path choice for OD pair (1,1).

It shows clearly the average passenger cost increases with respect to the demand level, and the influence of demand level to the average cost is approximately linear. The cost of driving passengers and park-and-ride passengers increase since the network become over-saturated, while the cost of transit passengers almost remain the same, because of constant railway cost. Also, the increasing demand will generally force travelers to avoid driving in routes with high congestion levels. Hence for OD (1,1), the park-and-ride flow drops on Path 8 and increases on Path 7, as a result of more and more congested highways/freeways. Figure 4 can be used to determine the marginal travel time/cost at different demand levels, which supports the urban planning for policymakers.

Sensitivity analysis on parking and bus fare

We study the joint influence of bus fare and parking fare. We change the bus fare and parking fare by applying a multiplier to the fare in the original settings, which ranges from 1 to 5. We solve the MMDUE under different bus and parking fares, and plot the average cost for all travelers, driving travelers, transit travelers and park-and-ride travelers in Figure 5.

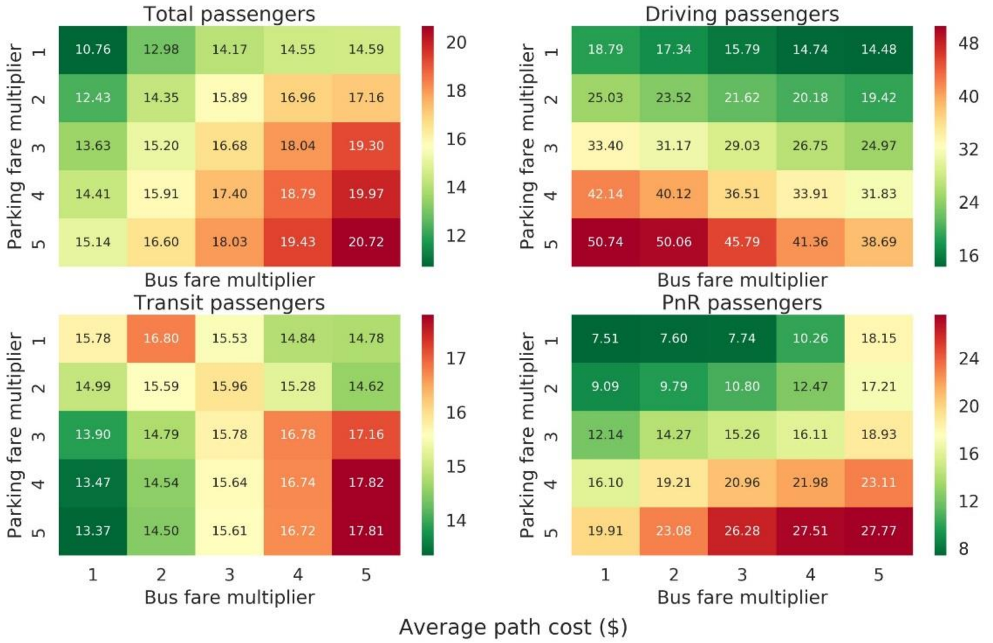


Figure 5 Heatmap of average cost for total traffic and different modes.

When the bus fare increases, the average traffic cost and average park-and-ride cost increase. This is because the park-and-ride cost includes bus fare and park-and-ride is one of the most preferred modes for all travelers. When the parking fare increases, the average costs of driving and park-and-ride increase substantially, because in both modes the travelers need to pay for the parking fee. In this case, the average traffic cost also increases. One interesting observation is that the average transit cost is relatively stable, which implies that transit cost has little change when bus fare and parking fare is changing. One important reason is that transit passengers do not pay the parking fare, and another reason is that many transit passengers can switch from bus to railway. Travel cost incurred on the railway model does not change as much regardless of parking or bus fare.

Influence of inconvenience (impedance) cost

In the original settings, the inconvenience (impedance) costs for carpool, transit, and park-and-ride are assumed to be at a very low level for testing the MMDUE model and the algorithm. As a result, the driving flow is marginal in the MMDUE solutions, which is deviated from the real situation in the study area. In this section, we will explore the impact of inconvenience cost on travelers' mode choice. We set the inconvenience costs for carpool, transit, and park-and-ride to be \$5 under MMDUE. The path choice probability over the morning peak for OD pairs (3, 1) and (4, 1) is presented in Figure 6. Generally, more travelers in O_3 and O_4 choose solo-driving before 7 AM, due to higher (inconvenience) costs for transit/park-and-ride. To conclude, the inconvenience costs has a considerable impact on travelers' mode choice. Therefore, public transit with great accessibility effectively attracts travelers. In future studies, a more fine-grained model should be developed to estimate the values of inconvenience costs, for example, there should be different inconvenience costs for groups of travelers with different demographic or traveling characteristics. This kind of further study is going to be significant in improving the accuracy of MMDUE.

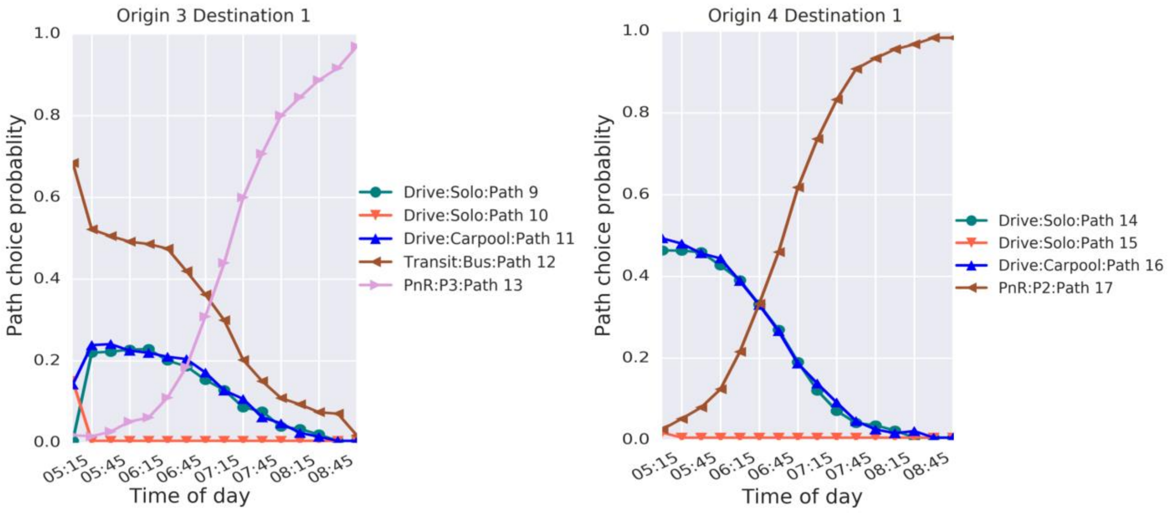


Figure 6 Time-varying path choice for OD pairs (3,1) and (4, 1) with high inconvenience costs.

Parked cars detection

As can be seen from the discussions above parking statistics are an important input to traffic models. Such statistics are readily available for parking garages but are harder to come by for on-road parking. Since cameras are widespread, we want to investigate if videos taken from vehicles can be analyzed to count parked cars.
Car detection itself is now a standard tool. Neural nets like Faster-RCNN (Figure 7) have been trained with large data sets and the resulting models and software are freely available [7].

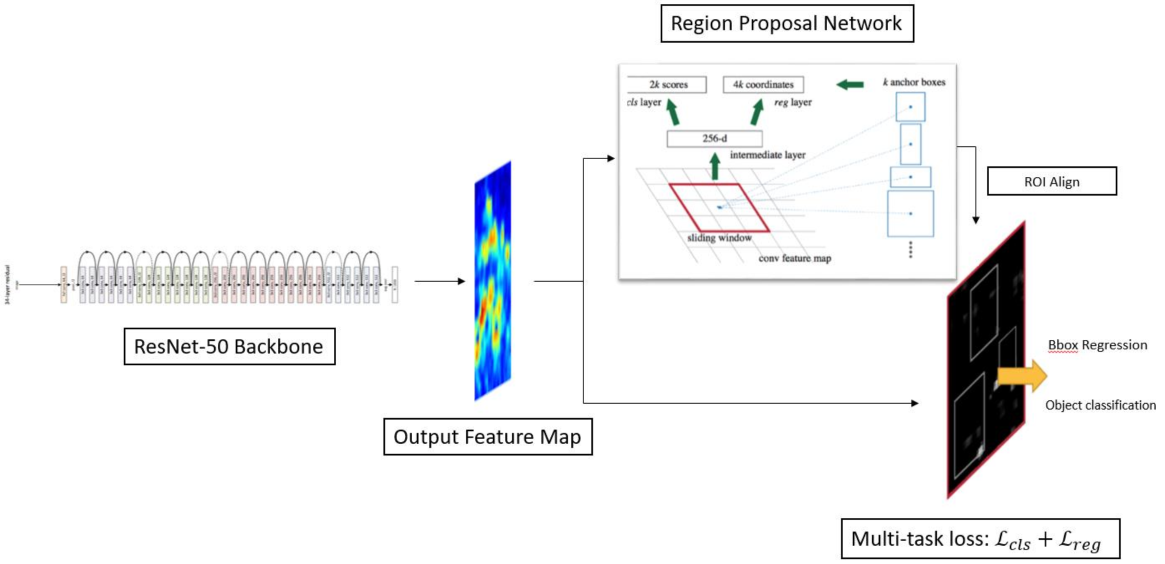


Figure 7 Architecture of Faster-RCNN.

We first tested if the faster-rcnn could distinguish between parked and moving cars, taking cues like distance to curb or proximity to other cars. We labeled a data set with the two classes “moving vehicle” and “parked vehicle” and trained the network with only these two classes. It turned out not to be very accurate. Therefore, we extended the classifier to not only take the RGB into account, but also optical flow. For each WxHx3 RGB image we create an optical flow volume (OFV) of the same size from a sequence of images (Figure 8).

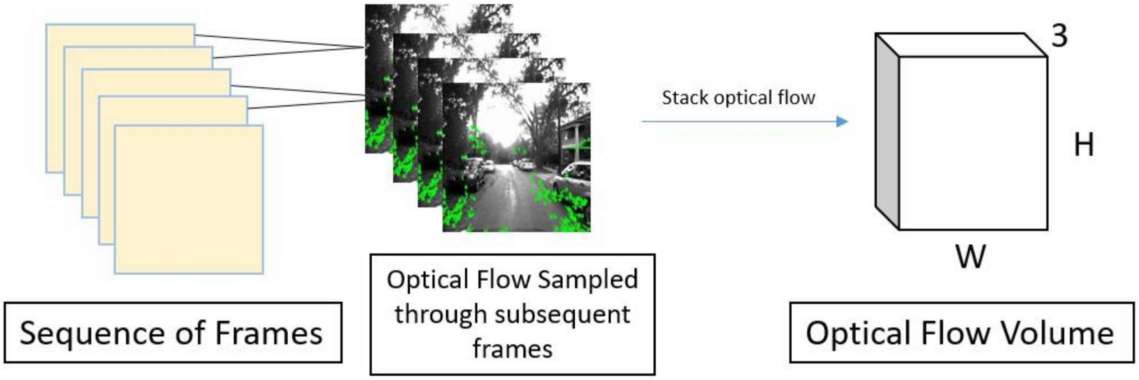


Figure 8 Construction of Optical Flow Volume.

In Figure 9 is an example of an RGB image and one corresponding optical flow layer.

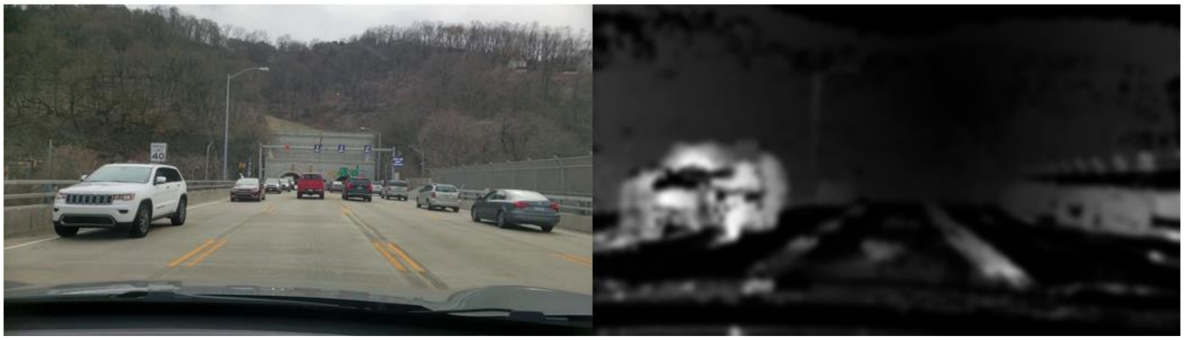


Figure 9 RGB image and corresponding optical flow image.

We modified the Faster-RCNN architecture to incorporate the OFV. The OFV goes through a parallel but otherwise identical backbone (Figure 10). The output from the RGB backbone is fed into the RPN as before. The outputs from both backbones are combined and used for the classification step.

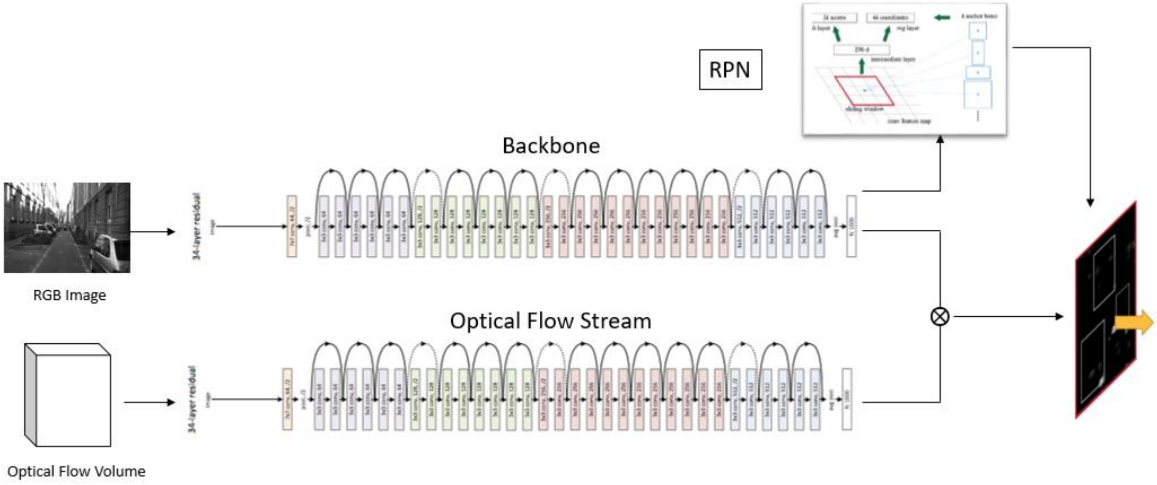


Figure 10 Modified Faster-RCNN architecture to incorporate the optical flow volume.

To train this new parallel network we used 3000 RGB-OFV pairs. The pre-trained weighs of the original backbone and RPN are frozen, whereas the OFV backbone is trained from scratch. We trained again for the two classes, “moving vehicle” and “parked vehicle”. The resulting precision-recall curves are shown in Figure 11.

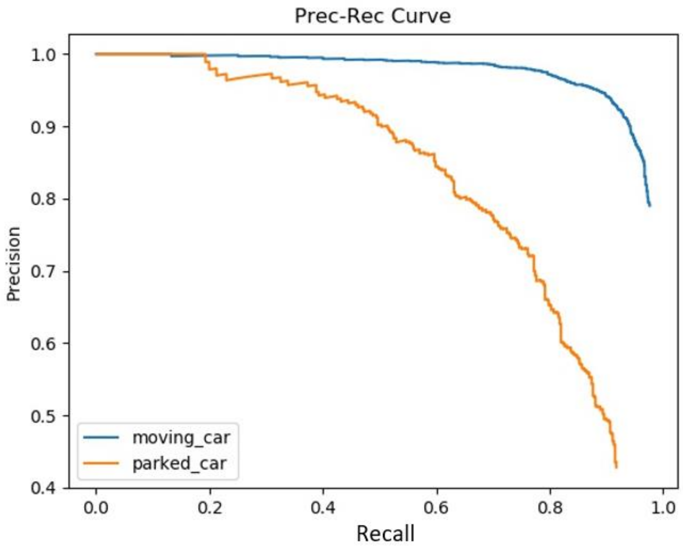


Figure 11 Precision-Recall curve for moving and parked cars.

The corresponding average precision (AP) are: Moving_car AP = 0.87 and Parked_car AP = 0.73. Figure 12 shows correct, false, and missed detections. Also shown are cases where the vehicles are parked on the edge of the road and off the road. Depending on the application, it is also desirable to distinguish between these two classes.

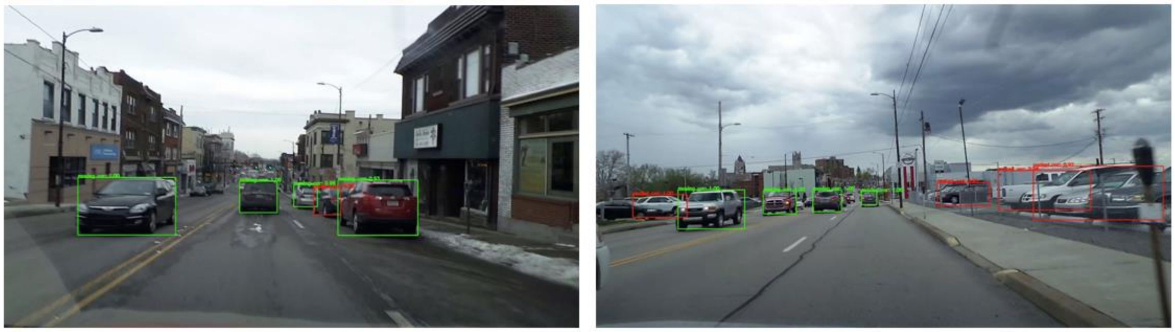
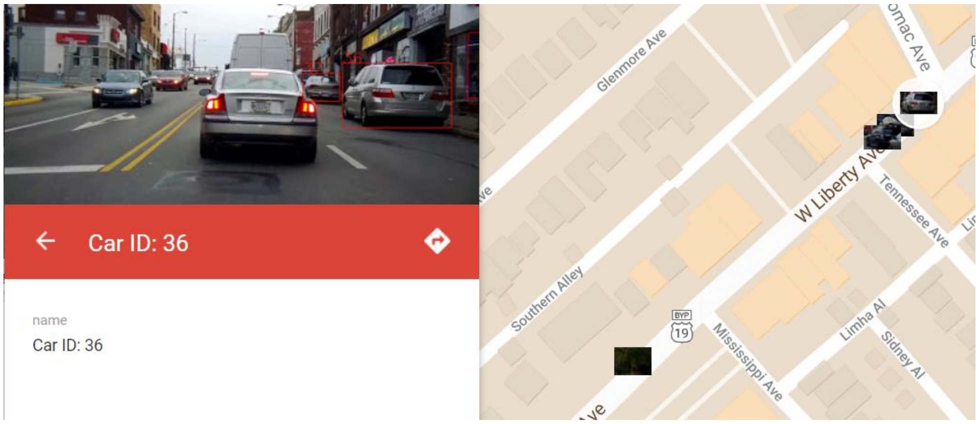


Figure 12 Examples of moving (green) and parked (red) vehicle detection. In the left image are two parked vehicles that are mis-classified as moving and some others in the background that are not classified at all. On the right side one can see many vehicles that are parked off the road.

To visualize the results on a map we created kml files that can be displayed on Google^{TM} Maps (Figure 13).



← Car ID: 36

name

Car ID: 36

Figure 13 Visualization of detection results: On the right cutouts of the detected vehicles are placed on a map. By click with the mouse opens the original image with a box overlaid indicating the detected vehicle and car ID.

Tracking

The individual parked or moving vehicles appear in many frames of the videos. Tracking the vehicles through all the frames they appear in will make the detection much more robust. Outlier classifications can be rejected as they are inconsistent with the detection in other frames. Furthermore, we want to count the individual vehicles only once and not recount them in each frame they appear.

Originally we used a naïve matching algorithm to associate detections frame-by-frame using intersection-over-union (IoU) as our only metric. This method only works well enough for dense detections and high frame rates. In actuality, detections are noisy, with many false negatives and false positives. Upon visual inspection, we can see that this often causes ID switches and fragmentations (Figure 14).

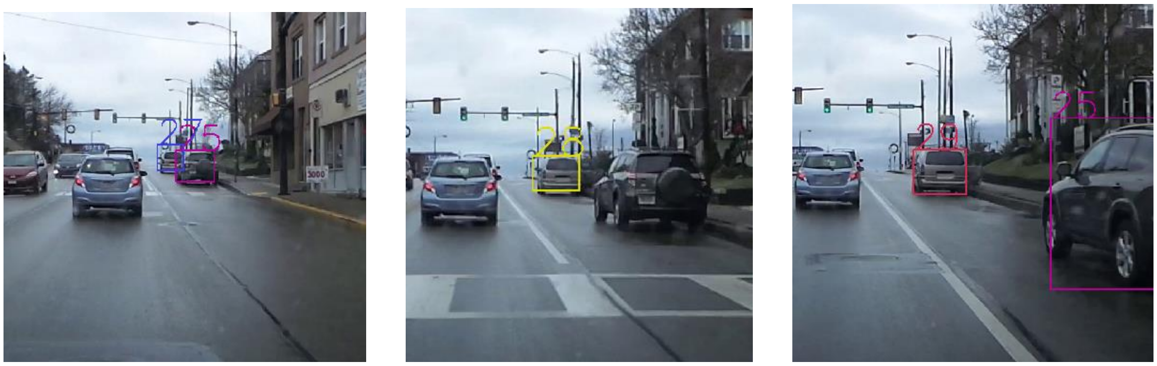


Figure 14 Examples of an ID switch due to missing detections and small IoU (Left to right, 27 -> 28 -> 29)

Due to increasing capabilities of object detection networks, tracking-by-detection has become the mainstream method for Multiple-Object-Tracking (MOT). Instead of the naive tracking we therefore use the DeepSORT [8] algorithm which incorporates a motion model along with an appearance model to perform tracking. The motion model uses a Kalman filter with a constant velocity motion assumption to find IoU of detections over frames. The appearance model uses a lightweight feature extractor (residual network, 9 layers, trained on large-scale person re-identification dataset) to extract embeddings, and calculates the cosine distance in feature space across embeddings.

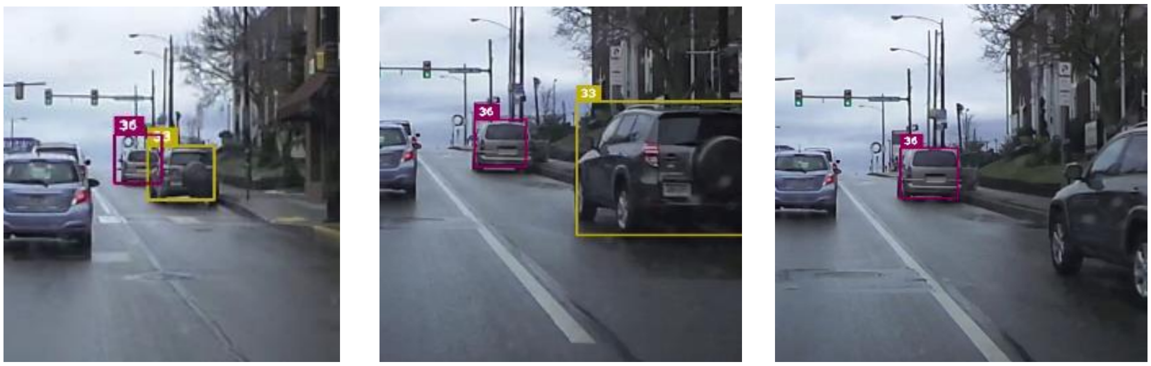


Figure 15 DeepSORT results, no ID switch (Left to right, 36 -> 36 -> 36)

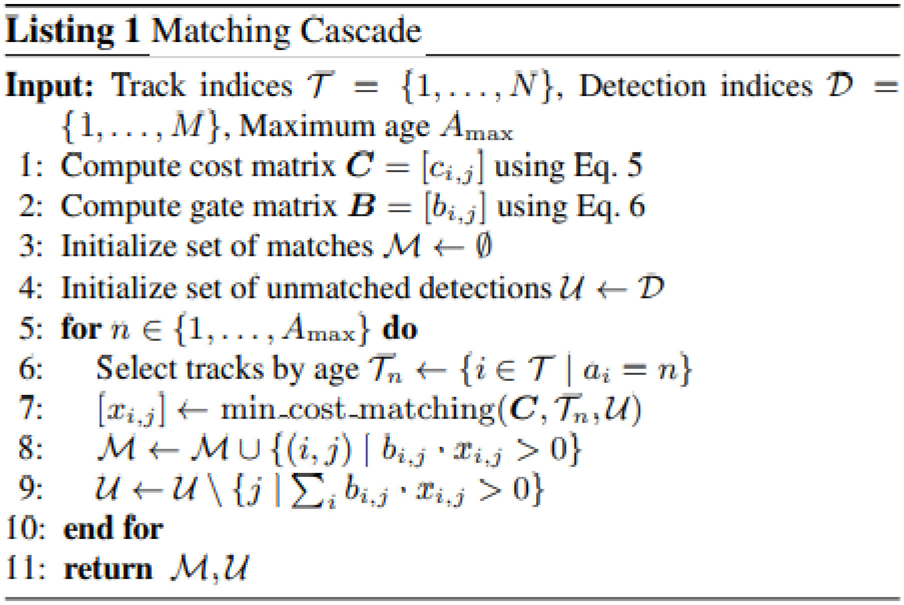
A weighted sum of these two metrics is then added and used for Hungarian matching:

$$c_{i,j} = \lambda d^{(1)}(i,j) \; + \; (1-\lambda) d^{(2)}(i,j)$$

$c_{i,j}$ denotes the weighted (weight λ) sum of distances $d^{(1)}$ and $d^{(2)}$ between detection *i* and *j*. The motion model accounts well for short-term scenes with little movements, while the appearance model accounts for long-term tracking by taking occlusions and large shifts in appearance into consideration.

DeepSORT performs significantly better than the naive method. It uses heuristics for adding/deleting tracks, namely, a track must be accurately detected for at least three frames before being considered a valid track, and tracks that have not been observed for a maximum time are deleted. A threshold is also used to gate metrics.

To handle occasional missed detections or occlusions, a “matching cascade” is implemented:



Listing 1 Matching Cascade

```
Input: Track indices $\mathcal{T} = \{1, \dots, N\}$, Detection indices $\mathcal{D} = \{1, \dots, M\}$, Maximum age $A_{\max}$
Compute cost matrix $\boldsymbol{C} = [c_{i,j}]$ using Eq. 5
Compute gate matrix $\boldsymbol{B} = [b_{i,j}]$ using Eq. 6
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
for $n \in \{1, \dots, A_{\max}\}$ do
    Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
    $[x_{i,j}] \leftarrow$ min_cost_matching($\boldsymbol{C}, \mathcal{T}_n, \mathcal{U}$)
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i, j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
end for
return $\mathcal{M}, \mathcal{U}$
```

Tracks are matched by age, and newer detections are given more priority. Detections which have not been associated are then passed to the next matching iteration.

Below we show some failure cases (Figure 16 and Figure 17), mainly ID switches. This can occur for a number of reasons:

(1) There are false negatives (missed detections), causing matching failures, which in turn create new tracks.
(2) Camera motion results in tracks overlapping with each other.
(3) Appearance model cannot distinguish well enough between cars of similar build.

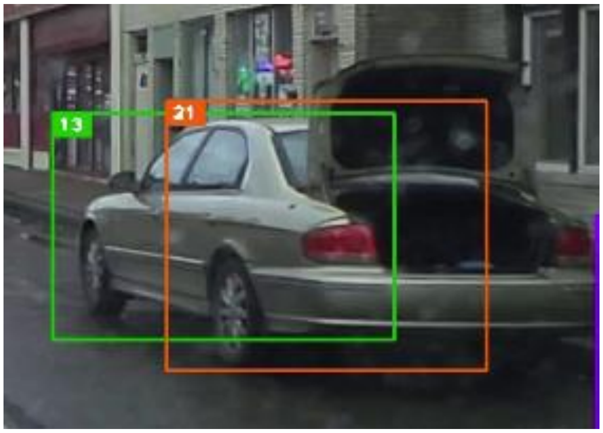


Figure 16 Overlapping detections, creating two tracks of same object.

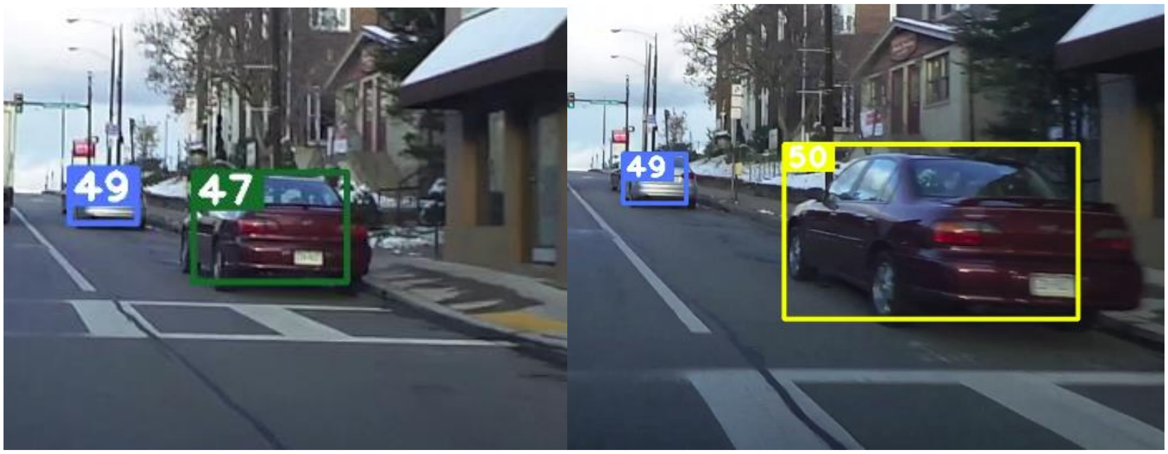


Figure 17 ID switch due to missing detections in between (Left to right, 47 -> 50)

Training on three classes

We seek to robustly count parked vehicles from simple camera feed, to aid with traffic and parking analysis. To this objective, we initially train an object detector with two classes (moving/parked). Often we encounter occurrences of parked cars in other areas besides the road, like parking lots of a car dealership or off-road parking for a shopping center. Depending on the application, these parked cars might not be of interest. E.g. if one wants to study the compliance to roadside parking restrictions during rush hours, the off-road parking is not relevant.

We thus extend our detector by another class, namely, "parked cars/side" with following detection performances:

Moving_car AP : 0.88
Parked_car AP : 0.68
Side_car AP : 0.66
mAP : (0.88 + 0.68 + 0.66) / 3 = 0.74

As is expected when the number of classes are increased, the AP of some classes decrease. With two classes Parked_car AP was 0.73 (see earlier section). Overall the performance is still good.

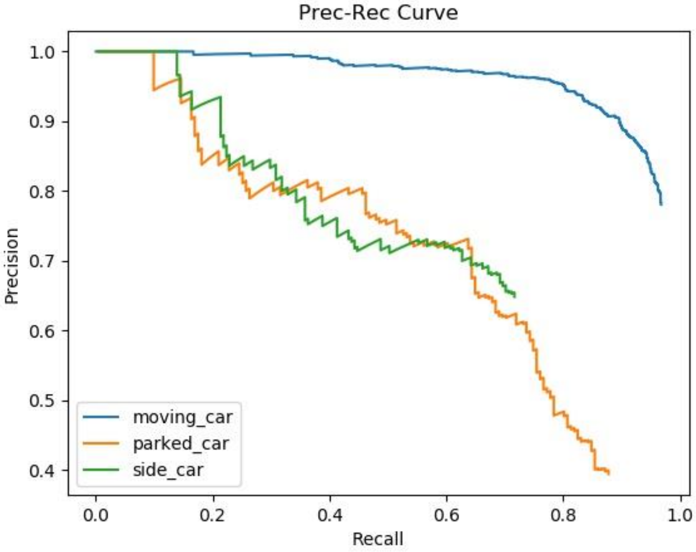


Figure 18 Precision-Recall Curve after training with 3 classes.

Though the detector does have some errors distinguishing between the three classes, we find that after running DeepSORT on these detections, that noisy detections are filtered out, and tracks that are detected for a majority of the time remain. This aligns with the goal of our specific application, where we want to count the vehicles parked on the side of the road. Figure 19 shows many successfully tracked cars. Moving cars and cars parked on the left or right side of the roads are counted, while vehicles in adjacent parking lots are ignored.

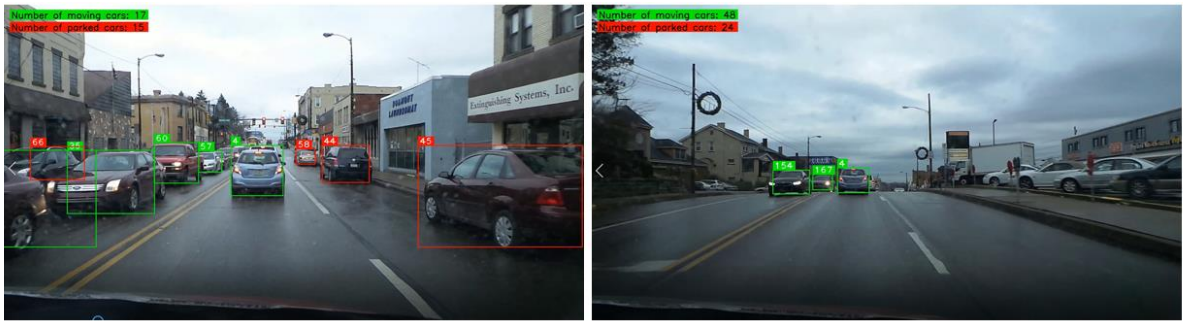

Number of moving cars: 17
Number of parked cars: 15
66
35
60
57
4
58
44
45
Extinguishing Systems, Inc.
Number of moving cars: 48
Number of parked cars: 24
154
167
4

Figure 19 The two images show detection and tracking of moving (green) and parked cars (red). Only cars parked on the road are counted, vehicles in a parking lot on the side are ignored. The boxes around the cars show their ID, the total number of detected and tracked cars are shown in the upper left corner.

Evaluation Results

In order to quantitatively evaluate the performance of our system we analyzed 60 seconds of a video, focusing on parked cars. The following table lists the experimental details:

Evaluation Video Details	
Video duration	60 seconds
No. of frames	703
Location	W Liberty Ave
Distance traveled	341 m
No. of vehicles	16

The MOT was evaluated with the metrics from [9] and the results are listed in following table:

MOT Evaluation Results	
IDF1 : 95.3%	ID F1 Score. The ratio of correctly identified detections over the average number of ground-truth and computed detections.
IDP : 97.5%	ID measures: global min-cost precision.
IDR : 93.2%	ID measures: global min-cost recall.

Recall : 94.4%	Number of detections over number of objects.
Precision : 98.8%	Number of detected objects over sum of detected and false positives.
GT : 16	Number of ground truth tracks
MT : 9	Mostly tracked targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span.
PT : 5	Number of objects tracked between 20 and 80 percent of lifespan.
ML: 2	Mostly lost targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.
FP : 20	The total number of false positives.
FN : 95	The total number of false negatives (missed targets).
IDs : 5	The total number of **identity switches.**
FM : 9	The total number of times a trajectory is fragmented (i.e. interrupted during tracking).
MOTA : 92.9%	**Multiple Object Tracking Accuracy**. This measure combines three error sources: false positives, missed targets and identity switches.
MOTP : 93.7%	Multiple Object Tracking Precision. The misalignment between the annotated and the predicted bounding boxes.

Conclusion and future work

This project allows us to collect time-varying parking data that could be used, in addition to existing traffic data, to understand travel behavior among various travel modes, such as driving (including solo-driving and carpooling), public transit, and park-and-ride. The travel behavior of all travelers in the network is then simulated to assess system performance and features of passenger/vehicular flow. The results are used to assess policies and operational strategies that target congestion reduction. Those policies and strategies are integrated into the behavioral model and simulation to estimate how passenger/vehicular flow would change comparing to the baseline, so that we can deploy the right policies and strategies to ensure their efficiency in reducing congestion. As one example, we found that when changing the parking prices from $1 to $5 per trip in downtown Pittsburgh, the use of public transportation increases by 11% and the vehicular traffic decreases by 7%.

With the development of the parked car counter we are now able to measure street parking and include it in the model. The overall performance of the parked car counter of MOTA = 92.9% is very good.

In the near future we want to apply the parked and moving car counter on videos recorded from transit buses. These transit buses regularly travel their routes and we will therefore be able to obtain traffic statistics with high temporal resolution.

References

[1] Pi, X., Ma, W. and Qian, Z.S., 2019. A general formulation for multi-modal dynamic traffic assignment considering multi-class vehicles, public transit and parking. Transportation Research Part C: Emerging Technologies, 104, pp.369-389.
[2] Ma, W. & Qian, S. (2015), Traffic impact of the Greenfield bridge closure. The public works of Pittsburgh technical report, Civil and Environmental Engineering, Carnegie Mellon University.
[3] Ma, W., Pi, X. & Qian, S. (2019), 'Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs', arXiv preprint arXiv:1903.04681
[4] Ma, W. & Qian, Z. S. (2018), 'Estimating multi-year 24/7 origin-destination demand using high-granular multi-source traffic 720 data', Transportation Research Part C: Emerging Technologies 96, 96–121.
[5] Tobin, R. L. & Friesz, T. L. (1988), 'Sensitivity analysis for equilibrium network flow', Transportation Science 22(4), 242–250.
[6] Patriksson, M. (2004), 'Sensitivity analysis of traffic equilibria', Transportation Science 38(3), 258–281.
[7] https://github.com/facebookresearch/detectron2
[8] https://github.com/nwojke/deep_sort
[9] Bernardin, K. & Stiefelhagen, R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. Image and Video Processing, 2008(1):1-10, 2008.