Road Violation Detection

Mingrui Wu [1]
Yao Lu [1]
Jianghang Lin [1]
Chaoyang Zhu [1]
Jing He [1]
Zhenjie Yu [1]
[1]School of Informatics, Xiamen University, China

Abstract

Road violation events detection is a road traffic management problem, which aims to help traffic administrators deal with numerous traffic violations.However, most existing methods focus on only one violation event detection and they rarely take the speed of the entire pipeline into consideration, which may fall short in practical.In this work, we propose a road violation detection framework in actual scenarios to auto-detect road violation events, including parking event, jam event, illegal lane change event, retrograde event, pedestrian event, and spill event.Our method is known as tracking by detection, which achieves real-time and high performance.

Introduction

The last few years have witnessed a new wave sweeping across every sector of our society. Brought by deep learning, from research area including neural network design, video surveillance, anomaly detection to practical applications such as face recognition system installed just at the entrance, everything is automated, by that we mean no human intervention in between. Unsurprisingly, detecting road violation has also become automated. However, during the process towards automation, several key challenges still remain open and unresolved. For example, how to classify each of the road event and form a taxonomy is a non-trivial task, for the reason that multiple violation could occur simultaneously. Or, under only one violation case, say, illegal lane change event, there exist a possibility that multiple cars may violate this rule in one single frame. Above are distinct problems in road violation detection, besides these, it also suffers from the robustness and accuracy of object detection methods.

Based on these unsolved problems above, we proposed some methods to address them. First we captured the cars in the road for observing their behaviors by YOLOv3. The core idea of YOLO series algorithms is to solve the object detection problem as a regression problem. YOLO chooses to directly input the entire image into the network, and then directly return to the position of the border and the category of the object. Because of the performance and the efficiency of YOLOv3, we used it as our basic object detection algorithm. And then we proposed some methods for different kinds of road behavior respectively to address them, such as the nearest neighbor method and GMM , etc.

In order to verify the effectiveness of the methods used for different road behaviors, we carried out experimental tests respectively. From the experimental results, the proposed methods can complete the task well.

Our main contributions can be summarized as following:

- Research the application of YOLOv3(Redmon and Farhadi 2018) in road violation detection tasks.
- Propose a solution to the misjudgment of events caused by insufficient object detection accuracy of YOLOv3.
- Explore real-time vehicle tracking algorithms.All the work we do is trying to make it practically used in real scenarios.

The rest of the paper is organized as follows, Section II is the related work and briefly clarify how the state-of-the-art solve the event detection problem. Section III introduces our newly proposed method to solve the problem.

Related Work

Object Detection: Given a frame, the task demands to localize the objects and return the bounding boxes and class labels of objects. It is essentially a compound task, i.e., regression sub-task and classification sub-task combined. In recent years, the object detection algorithm has made great breakthroughs. According to its problem-solving ideas, it can be divided into two categories. One is the two-stage R-CNN series of algorithms, including R-CNN(Girshick et al. 2014), Fast R-CNN(Girshick 2015), Faster R-CNN(Ren et al. 2017), etc. These algorithms need to use heuristic methods to generate Region Proposal, and then perform object classification and position regression on the candidate frame. The another is one-stage algorithms, including SSD(Liu et al. 2016), YOLO series algorithms, etc., which use only one CNN network for direct prediction. In recent years, anchor-free one-stage object detectors are developed, The detectors of this sort are CenterNet (Duan et al. 2019),Cornernet(Law and Deng 2018),FCOS(Rezatofighi et al. 2019),etc.The two-stage scheme has high accuracy

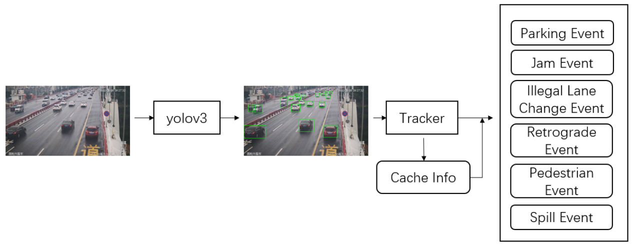

Figure 1: An overview of the method.Cahe info includes bboxes,classes,socres,IDs from yolov3 and Tracker in the historical frame.The Tracker output the current frame information and then input to the following detection logic unit together.

and slow speed, and the one-stage algorithm is fast, but the accuracy is lower. Considering performance,detection speed and model complexity, we chose YOLOv3 instead of YOLOv4(Bochkovskiy, Wang, and Liao 2020) as the basis for object detection.

Object Tracking: More and More modern trackers follows the trend of tracking-by-detection.One SORT (Bewley et al. 2016) uses a Kalman filter and associates each bounding box with its highest overlapping detection in the current frame using bipartite matching to track bounding boxes. Deep SORT (Wojke, Bewley, and Paulus 2017) proposes the overlap-based association cost in SORT with appearance features from a deep network.More recent approaches focus on increasing the robustness of object association.BeyondPixel (Sharma et al. 2018) track vehicles with using additional 3D shape information to.ARCF-HC(Huang et al. 2019)proposes a novel approach to repress the aberrances happening during the detection process by enforcing restriction to the rate of alteration in response maps generated in the detection phase, the tracker can evidently suppress aberrances and is thus more robust and accurate to track objects.These methods have two drawbacks. First, the data association discards image appearance features or requires a computationally expensive feature extractor. Second, detection is separated from tracking. In our project using CenterTrack (Zhou, Koltun, and Krähenbühl 2020), association is almost free. Association is learned jointly with detection. Also,it can be trained on the labeled video sequence, or through data enhancement of static images.

Proposed Solution

Our method is based on the yolov3 object detector and a tracker based on the nearest neighbor method.First,We use yolov3 to detect car boxes and person boxes.Second, We use the tracker to track the boxes.Then we add event judgment logic to determine violations.An overview of the method is shown in Figure 1.

Parking Event

We judge the parking event based on the distance the car has moved in a certain period of time.When within a period of time, the distance between a vertex of the box with the same ID in two frames is less than the threshold, we judge it as parking.

Concretely, we use the current frame information and the cache frame information to judge it.We look for a car with the same id in the cache frame before the set time threshold t_{park} according to the id of a certain car in the current frame.Then, when the Euclidean distance of their top left corner vertex is less than the threshold d_{park}, we judge it as parking.The threshold d_{park} changes adaptively according to the distance of the car body to avoid the influence of the distance of the car in the frame. The threshold d_{park} is calculated as follows:

$$d_{park} = f_{park} \cdot min(r - l, b - t) \tag{1}$$

where f_{park} is a distance factor and r, l, b, t are the right,left,bottom,top of the car box of the current frame.In addition, in order to avoid the impact of very small cars in the frame, we will not consider cars whose bbox area is less than the set threshold.

Jam Event

We judge the jam event based on the count of the cars c_{jam} whose movement distance less than the threshold over a period of time.Judging a car as jam is same as the parking event.The difference is that its distance factor f_{jam} is larger than the f_{park}.When the count of the cars judged as jam in the current frame is greater than the set threshold, we judge the current frame as jam.Additionally, we set a jam event duration.When the jam event happened, we will not judge the parking event.In order to avoid repeated detection of the jam event, we only judge it as a jam event when the previous frame is in a non-congested state and the current frame is in a congested state.

Illegal Lane Change Event

First, manually preset the lane line, and then we calculate the center point of the current detection frame to determine whether the center point changes left and right in the lane line.We use the list $l_{k,b}$ to describe the solid line of each lane's slope and offset. Traverse the list $l_{k,b}$ for each detection box to determine which marked line the y value of the center point of the current frame detection box belongs to. If it meets the y range of a solid line in the k,b list, use $y = kx' + b$ to calculate the x' of the solid line, and then determine whether the distance between the x' at the center of the detection frame and the above-calculated x' is less than the threshold h. If it is less, the detection is a lane change. It should be noted here that, since the farther the camera is, the larger the monitoring picture, so the threshold is adjusted according to the distance of the camera.

Retrograde Event

First of all, we preset the driving directions of the vehicles on different lanes, and then record the initial positions of all vehicles to determine whether the vehicles are driving in the reverse direction within the lane for a certain period of time. Algorithm 1 describes how the proposed method works for judging the retrograde event. We judge whether it is on the right side of the solid line on the left of the reverse direction and on the left side of the solid line on the right according to the initial location x of the current detection box to judge the retrograde event.

Algorithm 1 Judge Retrograde Event

Input:
The definete of the distance threshold, d;
The center point x', y' of the current detection box, C_n;
The record of the first position center point x, y for all track id, R_{id};
The record of the direction(forward and backward) of and the corresponding solid(left and right), R_{dict};
Output:
The result of the event (type: bool). True or False.
The process of algorithm.
for each center point$(x'_(i), y'_(i))$ **do**
  if $|y' - y| < d$ **then**
    get the direction of y' to y
    for each backward direction of the right of the left solid for R_{dict_i} **do**
      if x' in the right of the left solid for R_{dict_i} and in the left of the right solid for R_{dict_i} **then**
        return True
      end if
    end for
  end if
end for
return False

Pedestrian Event

When the detector detect a person in the frame,we judge it as pedestrian.

Spill Event

We use the Gaussian Mixture Model(GMM) background substraction method to detect spill.Concretely,we use GMM to find out the objects in motion and then subtract the detected objects(i.e. vehicle and person objects) obtained by yolov3.However, the moving objects found in the previous step are coarse due to the insufficient detection accuracy of yolov3, not all of which are spill (a large part of them belong to the vehicle or part of the vehicle), we need to refine these objects. Therefore, we train a vehicle parts classifier using SVM due to its fast inference speed in advance, and use it to filter out those objects belonging to the vehicle parts, and the rest is considered to be spill.

Experiments

This section introduces our experiment results, implementation details and further analysis.

Datasets

For this work, we evaluate on a real data which contain 500 training and 100 testing surveillance video data drawn

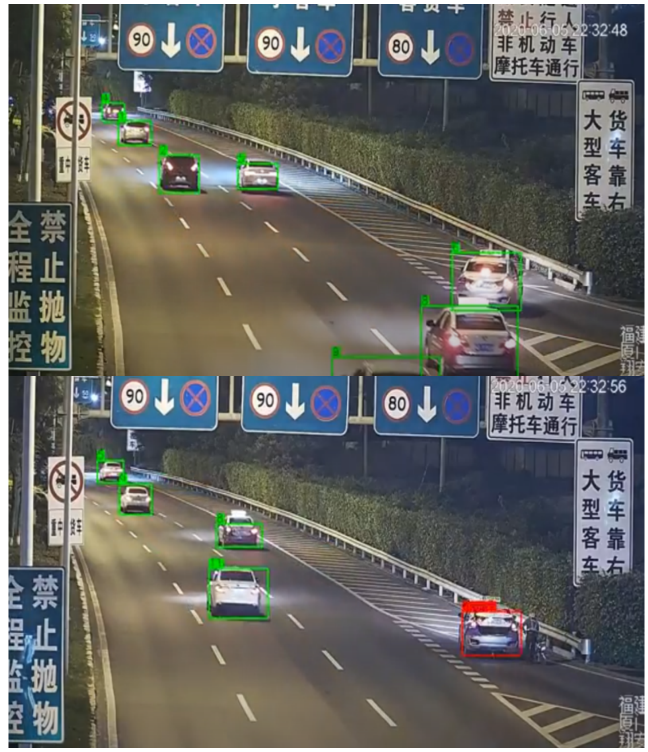

Figure 2: an instance of park.

from six classes,respectively.These videos describe the traffic state of highways, tunnel entrances and exits, and bridge decks. We detect different traffic incidents through preset rules.

Implementation Details

We set input size to be 512, min bbox area to be 4000, c_{jam} to be 5, duration of the jam event to be 200(8 seconds), t_{park} and t_{jam} are both set to 50(2 seconds), f_{park} and f_{jam} are set to 0.1 and 1.5.We use a single GPU GTX 980M to test the model.

Experiment Result

In this section, we report results on six tasks, including parking violation, jam event,illegal lane change event,retrograde event,pedestrian event and spill event.Figure 2 and Figure 5 show the instances of the park violation and the jam violation.We can see that our system is very capable of detecting illegal parking vehicles. The red detection box indicates illegal parking vehicles, and the green indicates normal vehicles.When multiple green detection frames appear in a certain range, we judge the speed of these detected objects to detect whether an jam event has occurred. it proves that our algorithm can work very well.   Vehicle lane change is the main task of our system detection.In Figure 3,Our detector detected an illegal traffic behavior in the tunnel that changed lanes and it will not misjudge normal vehicles.Vehicle retrograde detection is a very difficult goal to achieve, and the experimental results are difficult to observe from the pictures.

Figure 3: an instance of illegal lane change event.

Figure 5: an instance of jam event.

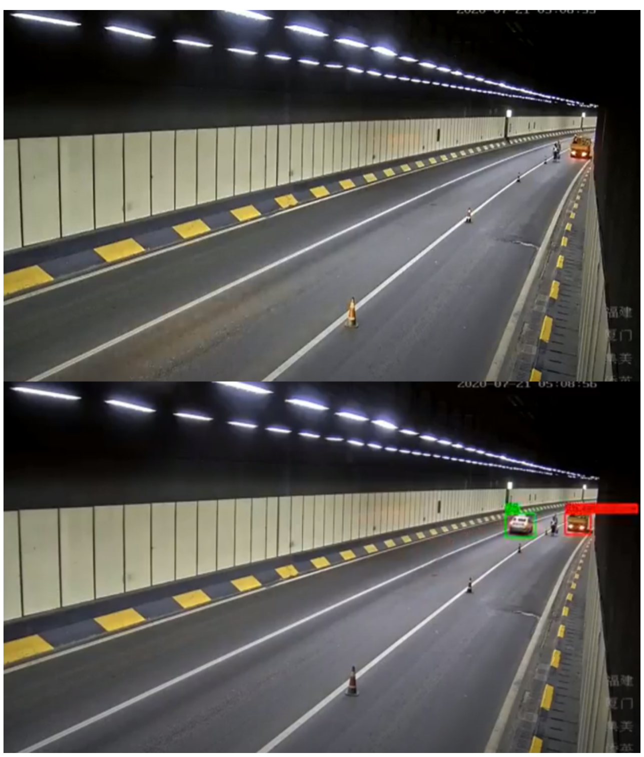

Figure 4: an instance of retrograde event.

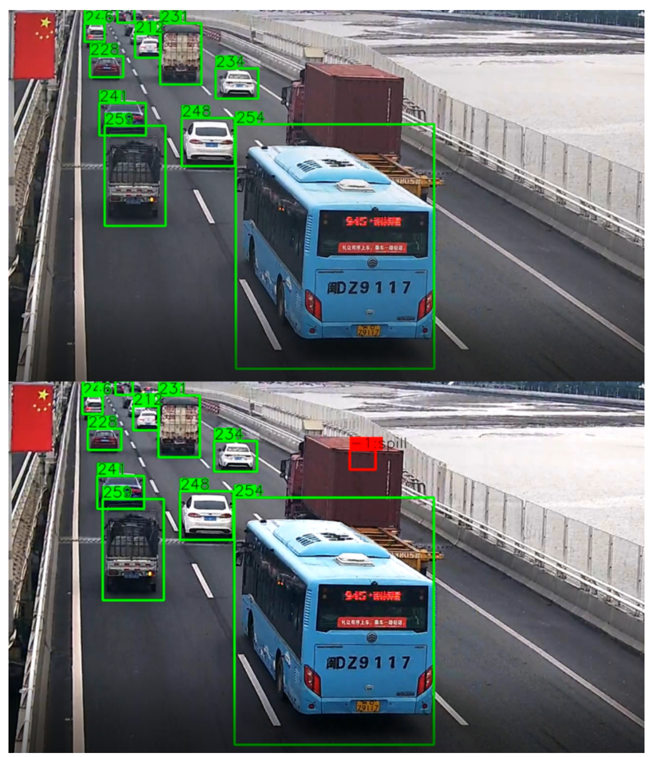

Figure 6: an instance of SVM model.

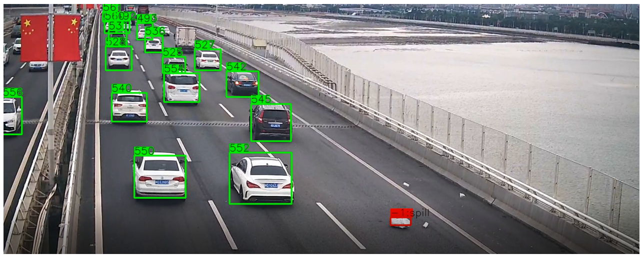

Figure 7: an instance of spill event.

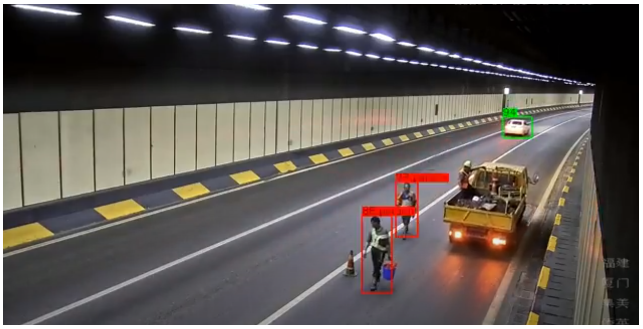

Figure 8: an instance of pedestrian event.

We have prepared relevant videos that will be displayed in subsequent project reports. But we still give the effect as shown in Figure 4.We also did pedestrian detection in real scenes. In the Figure 8, our system detected two pedestrians and marked them in the tunnel scene with a red detection frame.Figure 7 shown the spill event detected by our method.

Analysis

Our system basically achieved the performance we planned. Especially in illegal lane changes, pedestrian detection, and road congestion monitoring, great results have been achieved, which proves the effectiveness of our algorithm.We showed some good results achieved by our system in the experimental results section.But this is not enough to illustrate the performance of our system through pictures. We will prove the performance of our system in the form of video in the following class report.Of course, our project also have some disadvantage.In retrograde event, We found that the tracking window sometimes disappears that it means that the red tracking window is used to track the object, but the red tracking window will disappear in some frames in the test video.The results may be due to the slow speed of the retrograde vehicle, because we didn't consider this extreme situation in our detection algorithm which is our next step to improve.In the spill event as Figure 6,due to the birth defects of Yolov3 which can't detect small objects well,our system can't achieve perfect performance,so we use many way to assist like SVM and GMM to improve the results.we also find some disliking happening as Figure 2 that there were the pedestrian and parking on the white line.Under normal circumstances, our system is supposed to detect two event,pedestrian and park.but only one window appear.In short,when two or more than two event occurred within a very short distance,our detector can't very well even a little bad.The same situation appeared in Figure 8.In fact, the system should detect the pedestrian event and the retrograde event that a yellow car is retrograding but no red window tracking it.We will find a way to solve it as soon as possible.It is worth mentioning that our system is still very fast at detecting events. This is because we mainly use Yolov3, a very fast object detection algorithm, which ,of course, will lose some accuracy in a certain aspect, but the real-time performance in exchange is our needs.

Conclusion

The detection of road violation events is greatly affected by the significant changes in the complex environment.In this paper, we focused on real-time and not restricted scenes road violations events detection.We propose a real-time and generalized road violation detection framework, which achieves auto-detect parking event, jam event, illegal lane change event, retrograde event, pedestrian event, and spill event.In spill event, we innovatively propose the method that combines GMM with SVM to get more fined objects to compensate for the poor performance of yolov3 which truely improve the spill detect result with a little time consumption.Experiments and analysis demonstrate the effectiveness and efficiency of our approach to a wide range of violation detect tasks.

References

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. 2016. Simple online and realtime tracking 3464–3468.

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-Y. M. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934* .

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 6569–6578.

Girshick, R. 2015. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–1448. doi: 10.1109/ICCV.2015.169.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587. doi: 10.1109/CVPR.2014.81.

Huang, Z.; Fu, C.; Li, Y.; Lin, F.; and Lu, P. 2019. Learning aberrance repressed correlation filters for real-time uav tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 2891–2900.

Law, H.; and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 734–750.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science* 21–37. ISSN 1611-3349. doi:10.1007/978-3-319-46448-0_2. URL http://dx.doi.org/10.1007/978-3-319-46448-0_2.

Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* .

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6): 1137–1149. doi: 10.1109/TPAMI.2016.2577031.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 658–666.

Sharma, S.; Ansari, J. A.; Murthy, J. K.; and Krishna, K. M. 2018. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 3508–3515. IEEE.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, 3645–3649. IEEE.

Zhou, X.; Koltun, V.; and Krähenbühl, P. 2020. Tracking Objects as Points. *arXiv preprint arXiv:2004.01177* .