Traffic Anomaly Detection via Perspective Map based on Spatial-temporal Information Matrix

Shuai Bai[1], Zhiqun He[1], Yu Lei[1], Wei Wu[2], Chengkai Zhu[2], Ming Sun[2], Junjie Yan[2]
[1]Beijing University of Posts and Telecommunications
[2]SenseTime Group Limited
{baishuai, he010103,397680446}@bupt.edu.cn
{wuwei, zhuchengkai, sunming1, yanjunjie}@sensetime.com

Abstract

Anomaly detection on the road traffic has vast application prospects in urban traffic management and road safety. Due to the impact of many factors such as weather, viewpoints and road conditions in the real-world traffic scene, anomaly detection still faces many challenges. There are many causes for vehicle anomalies, such as crashes, vehicle on fires and vehicle faults, which exhibits different unknown behaviors. In this paper, we propose an anomaly detection system that includes three modules: background modeling module, perspective detection module, and spatial-temporal matrix discriminating module. The background modeling analyses the traffic flow to obtain the road segmentation results, and the vehicle flow superposition is used to obtain the continuous stationary region. The perspective detection module gets the perspective map by the first detection result, through which the image is cropped to uniform scale for different vehicles and re-detection. Finally, we get all anomalies by constructing spatial-temporal information matrix with the detection results. Furthermore, all anomalies are merged through the non maximum suppression (NMS) and the re-identification model, including spatial and temporal dimensions. The experimental results show that our system is effective in the Track3 test-set of NVIDIA AI CITY 2019 CHALLENGE, which finally ranked first in the competition, with a 97.06% F1-score and 5.3058 root mean square error (RMSE).

1. Introduction

Anomaly detection of traffic accidents plays an important role in urban safety, where abnormal traffic events will greatly reduce traffic efficiency. As more and more traffic cameras are deployed to record road information, it is important to develop an efficient, universal, automated anomaly detection method. However, the real-world traffic cameras record a wide range of viewpoints, weather, and road conditions. These issues make it difficult to design general purpose anomaly detection methods. Therefore, anomaly detection of traffic accidents is a meaningful task that still has many challenges to be solved.

As a basic problem, anomaly detection has been well-studied within diverse research areas and application domains. One definition is to pick out an instance different from other instances in the data and this selected instance is defined as an exception [3]. In recent years, anomaly detection methods based on deep learning have been rapidly developed. For example, supervised deep anomaly detection involves training a deep supervised binary or multi-class classifier, using labels of both normal and anomalous data instances [1].

Normal data is comparably easier to obtain, so semi-supervised anomaly behavior detection has been developed. A common way is using deep auto-encoders [4, 5] to train normal data in a semi-supervised way. Sufficient training samples of normal class auto-encoders would produce low reconstruction errors for normal instances, over unusual events [28, 25]. Despite these methods have achieved great development, especially on some anomaly data sets, such as UCSD [17] and CUHK [15] Avenue, which are homologous data. Most of them are not satisfying when faced with some heterogeneous scenes like Shanghai [16] Tech. So it is inapplicable for the road traffic with more complex and unknown scenarios. In addition, the anomaly data is difficult to acquire so that these methods are hard to cover all the situations in the real-world.

With the rapid development of deep learning in recent years, it has achieved great success in basic computer vision problems, such as image classification [22] and object detection [14], and the accuracy of classification even exceeds that of humans. Object detection has also reached a relatively mature level. In particular, face detection has been widely used. In this task, we use the state-of-the-art de-

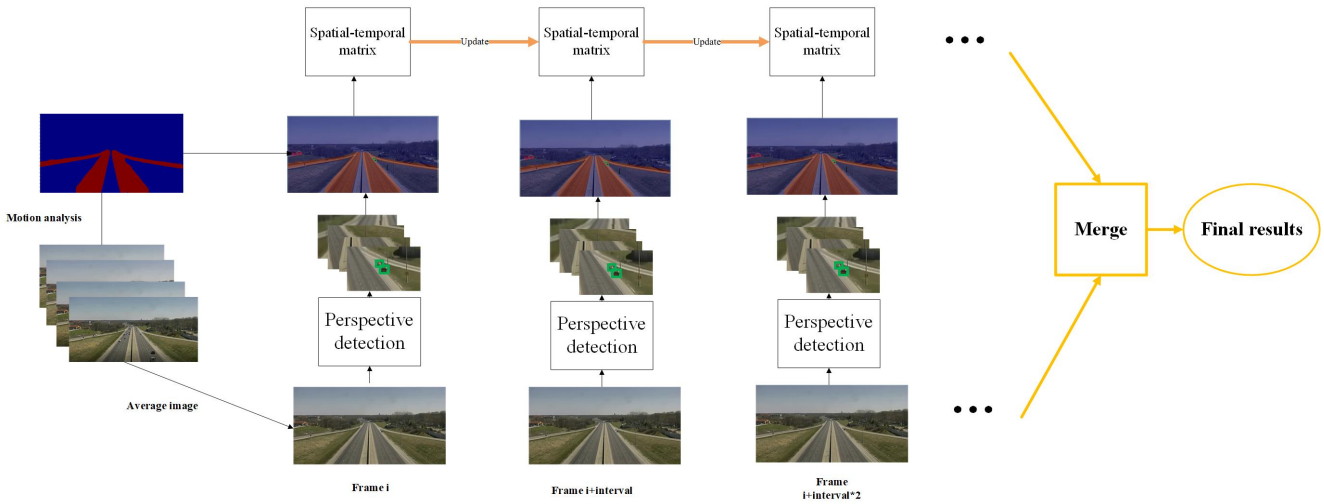


Figure 1. Overview of the architecture of our anomaly detection framework, which consists of background modeling module, perspective detection module, and spatial-temporal matrix discriminating module.

tection model, a ResNet50 [9] based Faster R-CNN model with feature pyramid networks [12] (FPN) and Deformable Convolutional Networks [6] (DCN) to detect vehicles in the image. In order to effectively capture all the vehicles in the video, a perspective detection method is adopted, which is detailed in section 3.2. The proposed system only depends on the detection results, which means a better generalization ability.

The main contributions of this article are summarized as follows:

- We present an unsupervised road segmentation method based on traffic flow analysis. It can effectively eliminate the interference of vehicles outside the road.
- We introduce the perspective relationship into the vehicle detection of the traffic scene, which can effectively improve the recall while ensuring accuracy.
- We design an anomaly discrimination method based on spatial-temporal information matrix, which transforms the analysis of a strip trajectory into an analysis of spatial position. It can not only accurately locate the anomaly start and stop time, but also effectively improve the robustness of the system.

Based on the above technical contributions, we propose a high performance traffic anomaly detection system. We evaluate on the Track 3 test-set of the NVIDIA AI CITY CHALLENGE [19], and the experimental results show the effectiveness of the system in real-world scenarios. We ranked first among the 23 participating teams, and we obtain the F1-score metric at 0.9706 and the RMSE metric at 5.3058.

2. Related work

Vehicle detection, as a basic module of road analysis, plays an important role in anomaly analysis. With the development of deep learning, continuous improvement is achieved in the field of detection, including [21, 13, 24, 6]. Our detection results also benefit from these methods.

The anomaly detection has been well-studied within diverse research areas and application domains. Benefit from the widespread application of convolutional neural networks in computer vision, the current mainstream methods have shifted from traditional machine learning methods to deep learning methods. The current mainstream methods are mainly including: semi-supervised, unsupervised, hybrid models, and one-class neural networks [2]. Semi-Unsupervised mainly includes auto-encoder [4] to fully learn normal samples, which detects anomalies by comparing normal and abnormal results. In addition, the method of generative adversarial [11] has been developed. Unsupervised often uses reinforcement learning to learn anomalies [7, 10], while Hybrid uses a combination of deep learning features and traditional algorithms for anomaly detection [8].

For traffic videos, there are several attempts to detect human violence or anomaly events in crowd scene [18, 20, 26]. In [26], a deep anomaly ranking model is proposed to predict high anomaly score in the testing videos. At the same time, some work is designed to detect a specific anomalous event, but the anomalies in the actual scene are complex and changeable.

The problem of vehicle anomaly detection for road scenes has received wide attention with the development of smart cities. In NVIDIA AI CITY CHALLENGE 2018[19],

[30, 27] use the background modeling method to effectively eliminate the interference of the mobile vehicle, and obtains the location of the static region to analyze, which has achieved competitive results. In this paper, we further analyze the traffic information and construct a dynamically analyzed pipeline based on the spatial-temporal information matrix, which is able to effectively adapt to a changeable environment and can be used in real-world scenarios.

3. Method

We construct a pipeline of anomaly detection framework, which contains three steps. The first step uses the traffic flow analysis to model the background and acquire the perspective detection module to obtain the detection results. Then, we utilize the detection results to construct the spatial-temporal information matrix, and judge the anomaly by the spatial-temporal matrix, and finally backtrack and merge the obtained anomalies to get an accurate time.

3.1. Background and space modeling

In fact, the concerned anomalies always happen in the road. However, the vehicles beside the house or in the parking lot can interfere with our results. But it is very difficult to effectively get the segmentation image of roads using a certain model in such multiple perspectives and complicated situations. A unsupervised segmentation method based on traffic flow analysis is applied, which is simple but effective.

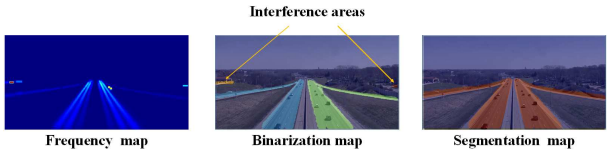


Figure 2. From left to right, it respectively represents the effect of binarization and filtering small connected areas. As shown in the middle figure, the small connected regions are often interference areas such as parking lots or houses.

In order to obtain the traffic information, we continuously weight and overlay the results of the detection result of each frame to the corresponding position, so as to obtain the frequency map of the vehicle in the global position. After normalizing the whole image, we perform binarization to obtain a segmentation map of the traffic flow. For example, as shown in Figure 2, several connected regions can be seen, in which the small connected regions are often the vehicles in parking lots or beside the house. we filter the connected regions with smaller areas to eliminate these disturbance in the final segmentation result.

To obtain the region of background and stationary vehicles, a method of averaging images similar to [30] is adopted, which continuously calculating the weighted sum of the input frames in the whole video, which can enhance the static parts and suppress the moving parts in objects. In order to eliminate the influence of slow traffic as much as possible, we average images in a fixed interval instead of every frame, which can be expressed by the formula,

$$AVG\_image = (1-\alpha) \times AVG\_image + \alpha \times frame_j,$$
$$j = i \times interval + start\_frame. \tag{1}$$

As shown in Figure 1, the image overlay can effectively eliminate the moving traffic and extract the static vehicle.

3.2. Perspective detection

After filtering out the moving vehicles, we need to detect the stationary vehicles. But in the traffic scene, the small targets in the distance are easily missed, which leads to the missed detection of anomaly events. So as to solve this problem, we use the perspective relationship to normalize near and far targets to a smaller range of fluctuations, thus ensuring the recall rate for small targets in the distance.

In this section, we first introduce how to estimate perspective relationship and then crop the local image to an uniform scale.

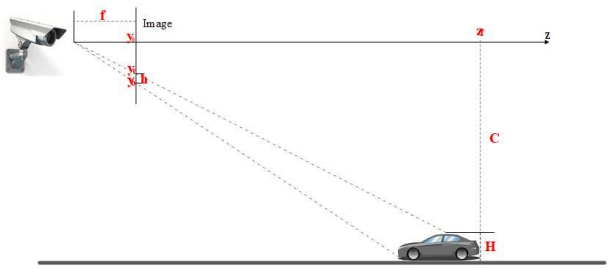


Figure 3. The perspective geometry between vehicles and camera.

Perspective relationship estimation. The perspective is widely used in vehicles, crowds, license plate detection and other scene with obvious perspective [23, 29]. For the perspective, the value of each pixel is defined as the number of pixels that one meters required in the image of the real scene [32], and the corresponding relationship of the size of the objects is similar in the image. Figure 3 shows the perspective relationship of the visual traffic camera. We can solve the perspective by

$$y_t = f(C-H)/z_f,$$
$$y_b = f(C)/z_f, \tag{2}$$

where y_t and y_b respectively represent the top and bottom of the vehicle, so we can get the following mapping relationship. The proportional relationship between the target size in the figure and its in the real world can be expressed

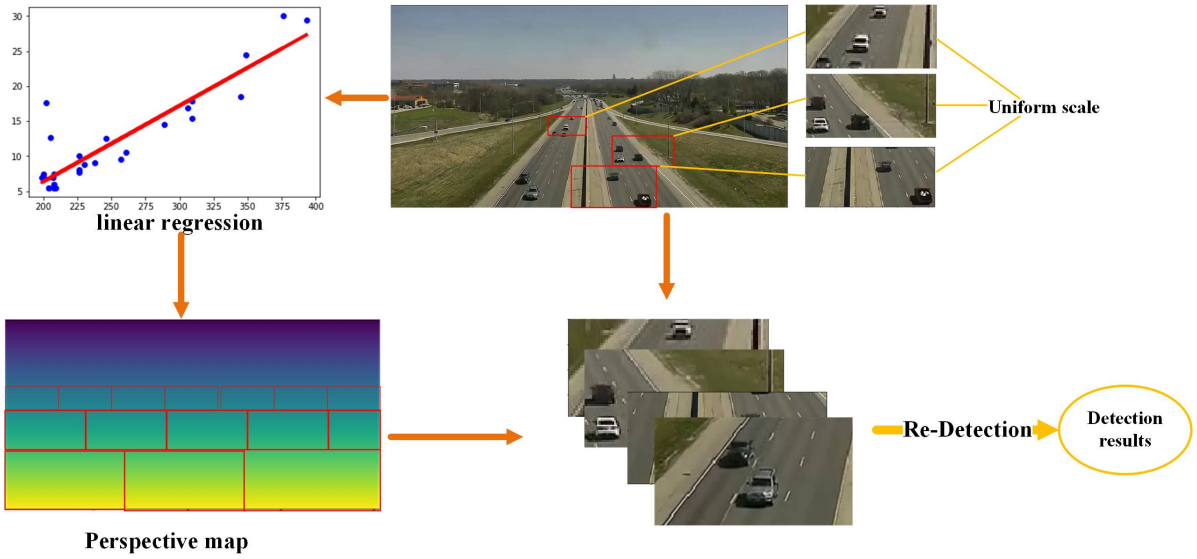


Figure 4. Firstly, according to the initial detection result, the perspective map is solved with linear regression map, and then the area is cropped to ensure that the capacity of each region is consistent. The final detection result is obtained through re-detection. As shown in the upper right corner, after the divided area has been resized, the vehicle maintains a consistent size.

as:

$$\frac{h}{H} = \frac{1}{C-H} \times y_h, \tag{3}$$

Therefore, we assume that the difference between the height of the vehicle is uninfluential. So we use a constant H. Then, for the same camera, $k = \frac{H}{C-h}$ is also a constant, which can be simplified as $h = k \times (y_h - y_0)$, where y_0 is the position where the horizontal line starts. The relationship can be expressed as

$$h = ky + b, \tag{4}$$

where $b = -k \times y_0$ and h is obtained by the initial detection result. We set $h_i = \sqrt{width \times height}$, y is the $y - axis$ position of the corresponding detection box. We obtain k and b by linear regression.

Based on k and b, the scales of different regions in the image are normalized. The area of the region is divided according to the number of targets that can be accommodated in the region. Here, we assume that the height of each cropped region with the same capacity for vehicles,

$$A = \int_{y_1}^{y_2} \frac{1}{h} dy = \int_{y_1}^{y_2} \frac{1}{ky+b} dy, \tag{5}$$

where A donates the capacity of the region between y_1 and y_2.

For too small targets, the details of the contours have been lost. Even if enlarged, it is difficult to detect. So we set the minimum target size that can be detected in the image is $h_0 = 8$. Then the starting point of the integral is solved that $y_0 = \frac{h_0 - b}{k}$. That means, the target with the height of 6 pixels in the images will be not considered. In the experiment, the capacity of the region is set to 3. Then the scale-normalized image is subjected to secondary detection to obtain the final detection result, which improves the detection of small targets. This process is visualized in detail in Figure 4. Besides, we use FPN-DCN network as our detector, where ResNet50 is used as backbone, and finetune it on vehicle detection datasets, which is described in section 4.1.

3.3. Spatial-temporal matrix discrimination

The position and time information of the stationary vehicle is obtained through the perspective relationship detection module. But not all detected stationary vehicles are anomaly, e.g., the vehicle waiting for the red streetlight and some false positive detection. At the same time, the determination of the start and end time for the same anomaly according to the detection result is important but difficult. Object tracking and optical flow are often employed to analyze the trajectory of the same vehicle. But the result of the trajectory reduction is often sensitive to the quality of the video and the interference of other vehicles in the road. In order to locate and identify each anomaly more robustly, we no longer analyze each vehicle and its trajectory, but instead analyze each location.

We judge the anomaly event according to the information of the vehicle dynamics represented by the matrix of the original image size. The process of matrix update is summarized in Algorithm 1.

Algorithm 1 Updating spatial-temporal matrix

Input: The time of the current frame, t; The set of detected boxes B; Six information matrices, V_{old}; The segmentation mask,S;

Output: Six updated information matrices V_{new};

initialize two zero matrices $M_{detected}$ and M_{score};
for each $b \in B$ **do**
  if $score_b > 0.3$ **then**
    $M_{detected}[b] = 1$;
    $M_{score}[b] = maximum(M_{detected}[b], score)$;
  end if
end for
$M_{undetected} = (1 - M_{score}) \cdot S$;
$M_{detected} = M_{detected} \cdot S$;
$M_{score} = M_{score} \cdot S$;
$V_{detected} = V_{detected} + M_{detected}$;
$V_{score} = V_{score} + M_{score}$;
$V_{undetected} = V_{undetected} + M_{undetected}$;
$V_{undetected}[M_{detected}] = 0$;
$V_{start}[V_{detected} == 1] = t$;
$V_{end}[V_{detected} > 0] = t$;
$M_{delay} = V_{end} - V_{start}$
find the location with the most response of M_{delay}, pos;
if $M_{delay}[pos] > threshold_{time}$ **then**
  get the binarized matrix M_{binary} where $V_{detected}[pos] - V_{detected} \leq 1$;
  get a connected region of the peak position as the anomaly region with breadth first search (BFS) of M_{binary};
  if $\frac{V_{score}}{V_{undetected}} > threshold_{score}$ **then**
    start or keep anomaly status;
  end if
  if $V_{undetected}[pos] > threshold_{undetected}$ **then**
    finish the anomaly and output the anomaly information;
  end if
end if
$V_{state}[V_{detected} > threshold_{detected}] = 1$;
$V_{state}[V_{undetected} > threshold_{undetected}] = 0$;
$V_{detected}[(M_{detected} + V_{state}) > 0] = 0$;
$V_{score}[(M_{detected} + V_{state}) > 0] = 0$;
return $V_{detected}, V_{undetected}, V_{state}, V_{score}, V_{start}, V_{end}$;

Specifically, there are a total of six information matrices, where $V_{detected}$ records the number of times each pixel is continuously detected, and $V_{undetected}$ records the number of times each pixel is not detected continuously. V_{state} records the status of each pixel (whether or not this position is in the anomaly region). V_{score} records the total detection score for each pixel. V_{start} records the start time of the most recently detected for each pixel and V_{end} records the time each pixel last remained suspicious. $V_{undetected}, V_{detected}, V_{score}, V_{state}$ are in the form of heat-maps, and V_{start}, V_{end} just continually update the time recorded. The thresholds is described in detail in section 4.1.

The establishment of spatial-temporal information matrix. Through the road segmentation map S obtained in section 3.1, the counting matrix $V_{detected}$ and the scoring matrix $V_{undetected}$ are continuously updated only for the new static detection result of each frame in the road. When the position is continuously detected, the position enters an anomaly suspicious state. The suspicious state is updated by the continuous undetected matrix $V_{undetected}$ while time-related information V_{start}, V_{end} is recorded for each location. A major criterion for anomaly suspend in the road is the length of the suspend time. We only analyze the location that enters the suspicious state and lasts for the longest time at a time. When the duration is too long (greater than 60s), the position is used as the starting point in breadth-first traversal to obtain a connected region with similar count values in $V_{detected}$ as the anomaly region. As an anomaly region, it simultaneously outputs the start time when the position is detected. Until the suspicious state of the position is updated, the end time is output. The detected average score of the position is taken as the score of the anomaly. Each anomaly includes the location of the region, start time, end time and score.

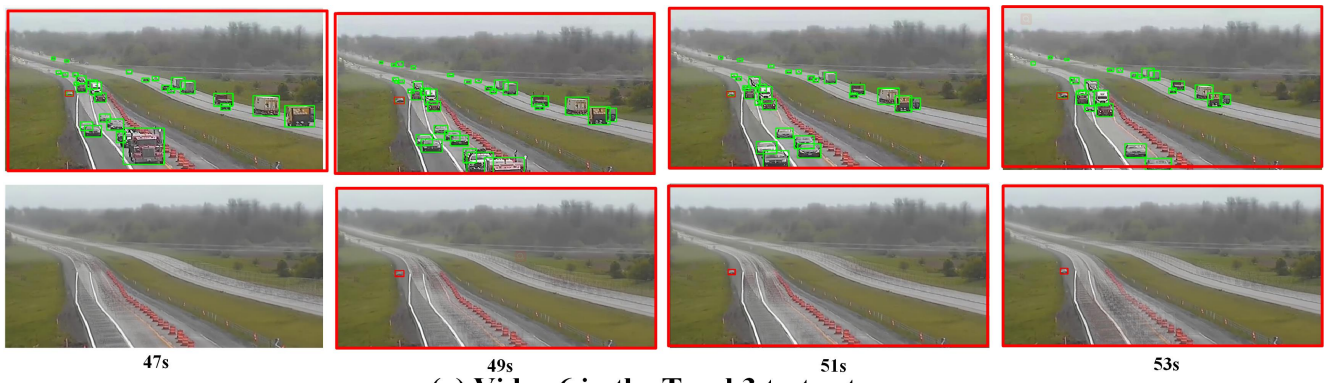
The results merge. For acquiring the anomaly results, we first apply the NMS method to merge the spatial locations. The earliest time for the anomaly update of IOU greater than a certain threshold is the start time, and the latest time is the end time. In particular, for anomalies with a time interval of less than 10s, we use the re-identification model to compare features in the two regions to eliminate the difference in displacement due to accidents. Due to the average image, there is a delay in the time when the static vehicle is detected. We perfume time backtracking for the original image of the anomaly region in fixed time length. When the intersection over-union (IOU) is greater than 0.5, update the start time of the anomaly. At the same time, if the vehicle is continuously detected in this area, we will continue to backtrack until it is not detected.

4. Experiments

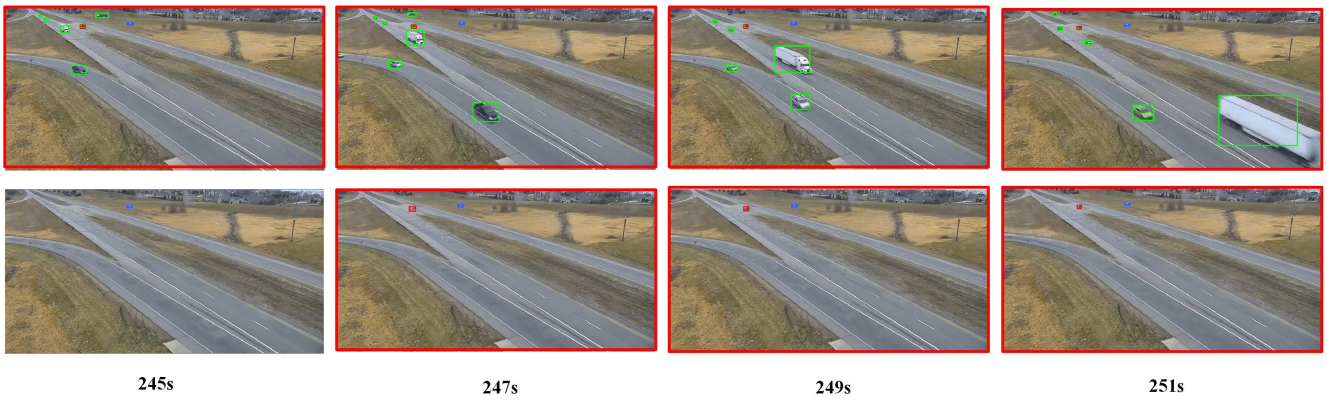
In this section, the experimental results of the proposed method is present in detail, and then introduce the dataset of NVIDIA AI CITY 2019 CHALLENGE Track3, and the corresponding evaluation method. Finally, our system is evaluated on this challenge.

4.1. implementation details

Background and space modeling. We normalize the frequency obtained by superimposing the traffic flow. The threshold of binarization is 0.1. In each part, the filtering



(a) Video 6 in the Track3 test-set



(b) Video 85 in the Track3 test-set

Figure 5. Example results on Track3 test-set. By averaging the image, we can get a delayed start time of the anomaly, and then we get a more accurate time positioning by backtracking the detection result of the original image.

area is 2,000 pixels for a small connected region. The interval for obtaining a static overlay is 4 frames, and the weight of the overlay is 0.05.

Figure 6. Qualitative detection results. Several difficult scenes with obvious perspectives were picked, and our perspective detection module shows great performance.

Perspective relationship detection. According to the perspective relationship, we take an image with an integral length of $y_{i+1} - y_i = 3$ and the minimum target size is $h_0 = 8$. Regarding training the detection network, we use UA-DETRAC [31] and VisDrone [33], part of the AIC train-set, where UA-DETRAC dataset consists of 10 hours of videos captured with a Cannon EOS 550D camera at 24 different locations at Beijing and Tianjin in China. There are 84k frames and more than 578k annotated bounding boxes in this dataset. In order to maintain consistency with AIC, we use the Gaussian blur enhancement mentioned in [27]. The VisDrone [33] is primarily captured by drones and contains a wide range of scenes ,time and weather, as well as a large number of small target objects. The detection network uses ResNet50 as the Backbone, joins the fast RCNN network of FPN and DCN, and trains with three shortest size of [800, 1000, 1200].

Spatial-temporal matrix discriminattion. The threshold for the normal state transition to the suspicious transition state is 6 consecutive frames detected, $threshold_{detected} = 6$, while the threshold for the suspicious transition state to the normal state transition is 8 consecutive frames, $threshold_{undetected} = 8$. The shortest time threshold for output anomaly

	F1	RMSE	S3
Our method	0.9706	5.3058	0.9534

Table 1. Our results on Track3 test-set

$threshold_{time} = 60s$, and the minim score threshold for anomaly $threshold_{score} = 0.8$. In addition the anomaly NMS threshold is 0.8. The Reid model uses in this task is the same as the model we used in NVIDIA AI CITY CHALLENGE TARCK2.

4.2. Track3 dataset

The Track 3 train-set and test-set of NVIDIA AI CITY CHALLENGE 2019 each contain 100 video, every video with a length of about 15 minutes, a frame rate of 30 fps and a resolution of 800×410. It may include anomalies caused by crashes, stalled vehicles. The main goal of the task is to detect these anomalies in the video, and give the start time and confidence score.

Evaluation for track 3 will be based on model anomaly detection performance, measured by the F1-score, and detection time error, measured by RMSE. Specifically,the track 3 score will be computed as

$$S3 = F1 \times (1 - NRMSE), \tag{6}$$

where is the F1-score and is the normalized root mean square error (RMSE). The score ranges between 0 and 1, and higher scores are better. A true-positive (TP) detection will be considered as the predicted anomaly within 10 seconds of the true anomaly. The normalization of RMSE will be done using min-max normalization with a minimum value of 0 and a maximum value of 300.

4.3. Experimental results

We evaluate our method on the Track 3 testing data. As showed in Table 1, we achieve 0.9706 F1-score while detection time error is only 5.3058 seconds, which demonstrates our proposed methods superiority and robustness. Local S3 score is obtained to 0.9534 by Equation 6. The final learderboard results among all the teams are shown in Figure 5, we achieve 0.9534 S3 score and rank the first place among all the participant teams.

5. Conclusion

Stationary vehicles in the road tend to be traffic anomalies. Vehicles moving normally or parking in the parking lots, beside the house and other places outside the road will cause interference to the anomaly detection. We propose an anomaly detection system that can reduce the non-anomaly interference to the maximum extent. Firstly, we quantify the frequency statistics of the spatial traffic, binarize and filter the small independent connected regions to achieve unsupervised segmentation of the road, thereby eliminating the

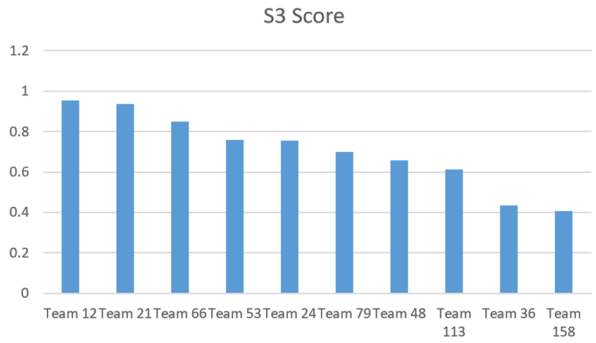


Figure 7. Compared results on the Track 3 test-set from the top 10 on the leaderboard.

interference of factors outside the road. the static objects is enhanced by continuously superimposing the input frames so as to eliminate dynamic traffic disturbance. In order to maximize the accuracy and recall rate of the detector, we utilize a perspective relationship transform both distant and near areas into a uniform scale, and then apply the state-of-art detector to get all vehicles. Furthermore, the spatial-temporal information matrix is used to analyze the anomaly of each position in the space. Finally, the NMS and Reid modules are applied to merge the anomalies to obtain the start and end time of the final results. Results on NVIDIA AI CITY CHALLENGE 2019 show our proposed method is effective on various scenes of traffic videos, which gets a 97.06% F1-score and 5.3058 RMSE. In future work, we plan to replace the detection result with vehicles density map to achieve a more continuous estimate of the vehicle's position and probability distribution, as well as improving the accuracy of time estimation.

References

[1] Raghavendra Chalapathy, Ehsan Zare Borzeshi, and Massimo Piccardi. An investigation of recurrent neural architectures for drug name recognition. *arXiv preprint arXiv:1609.07585*, 2016.

[2] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Outlier detection: A survey. *ACM Computing Surveys*, 14:15, 2007.

[4] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *International Symposium on Neural Networks*, pages 189–196. Springer, 2017.

[5] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR 2011*, pages 3449–3456. IEEE, 2011.

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. 2017.

[7] François de La Bourdonnaye, Céline Teulière, Thierry Chateau, and Jochen Triesch. Learning of binocular fixations using anomaly detection with deep reinforcement learning. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 760–767. IEEE, 2017.

[8] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Chengqiang Huang, Yulei Wu, Yuan Zuo, Ke Pei, and Geyong Min. Towards experienced anomaly detector through reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. Anomaly detection with generative adversarial networks for multivariate time series. *arXiv preprint arXiv:1809.04758*, 2018.

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[13] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[15] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013.

[16] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 341–349, 2017.

[17] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981. IEEE, 2010.

[18] Sadegh Mohammadi, Alessandro Perina, Hamed Kiani, and Vittorio Murino. Angry crowds: Detecting violent events in videos. In *European Conference on Computer Vision*, 2016.

[19] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty, Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, et al. The 2018 nvidia ai city challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 53–60, 2018.

[20] Mahdyar Ravanbakhsh, Enver Sangineto, Moin Nabi, and Nicu Sebe. Training adversarial discriminators for cross-channel abnormal event detection in crowds. 2017.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. 2015.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[23] Miaojing Shi, Zhaohui Yang, Chao Xu, and Qijun Chen. Revisiting perspective information for efficient crowd counting.

[24] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *IEEE Conference on Computer Vision Pattern Recognition*, 2016.

[25] Hongchao Song, Zhuqing Jiang, Aidong Men, and Yang Bo. A hybrid semi-supervised anomaly detection model for high-dimensional data. *Computational Intelligence Neuroscience*, 2017(1):1–9, 2017.

[26] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. 2018.

[27] Jiayi Wei, Jianfei Zhao, Yanyun Zhao, and Zhicheng Zhao. Unsupervised anomaly detection for traffic surveillance based on background modeling. pages 129–136, 2018.

[28] Drausin Wulsin, Justin A Blanco, Ram Mani, and Brian Litt. Semi-supervised anomaly detection for eeg waveforms using deep belief nets. pages 436–441, 2010.

[29] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. Spatiotemporal modeling for crowd counting in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5151–5159, 2017.

[30] Yan Xu, Xi Ouyang, Yu Cheng, Shining Yu, Lin Xiong, Choon-Ching Ng, Sugiri Pranata, Shengmei Shen, and Junliang Xing. Dual-mode vehicle motion pattern learning for high performance road traffic anomaly detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[31] Yan Xu, Xi Ouyang, Yu Cheng, Shining Yu, Lin Xiong, Choon-Ching Ng, Sugiri Pranata, Shengmei Shen, and Junliang Xing. Dual-mode vehicle motion pattern learning for high performance road traffic anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 145–152, 2018.

[32] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 833–841, 2015.

[33] Pengfei Zhu, Longyin Wen, Xiao Bian, Ling Haibin, and Qinghua Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.