Unsupervised Traffic Anomaly Detection Using Trajectories

Jianfei Zhao[1], Zitong Yi[1], Siyang Pan[1], Yanyun Zhao[1,2], Zhicheng Zhao[1,2], Fei Su[1,2], Bojin Zhuang[3]
[1]Beijing University of Posts and Telecommunications
[2]Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications
[3]Ping An Technology Co.,Ltd
{zjfei, yizitong001, pansiyang, zyy, zhaozc, sufei}@bupt.edu.cn
zhuangbojin232@pingan.com.cn

Abstract

Traffic anomaly detection of unsupervised videos has attracted great interests in computer vision field, and this task is very challenging since the scarcity of data and scene diversities. In this work, we present a robust framework for solving unsupervised traffic anomaly detection based on vehicle trajectories. The possible anomalies are detected and tracked from background image sequence of videos. The start time of the abnormal events is located by the decision module based on tracks. In order to better solve the problems of false detections and missed detections caused by the detector, we design a multi-object track (MOT) algorithm suitable for this task. We also present an adaptive unsupervised road mask generation method to filter out false anomalies outside the road area. Our method participated in the evaluation of 2019 AI CITY CHALLENGE Track3 and achieved good result.

1. Introduction

Unsupervised anomaly detection in traffic videos, such as stagnant objects, accidents, and anomalous objects, has been a challenging task due to the lack of annotated data and great diversities of scenes. However, it is worthy to explore robust computer vision methods to solve this problem with the increasing use of cars and the importance of traffic safety.

In previous studies, some researchers believe that it is possible to determine whether an event is an abnormality by reconstructing the characteristics of the normal events, since the scene will change when an abnormality occurs [8][4][16]. In fact, traffic anomalies usually involve large time and space ranges, scene diversities, and small targets at long distances. These are difficult problems that must be faced in traffic anomalies detection in unsupervised videos. So far reconstructing traffic events of unsupervised video has been extremely challenging in the field of computer vision. In despite of the excellent performance of the generative adversarial network(GAN) network [7], the performance of the model is still not robust enough when the scene changes.

In AI CITY CHALLENGE, the video scenes change greatly, and the areas of accidents are often small, some of them are difficult to distinguish with human eyes. In order to solving this problem, [25] exploited background modeling method to extract background images which may include stop vehicles, then detected vehicles from background images, and finally obtained the trajectory of each vehicle with a simple tracking method to determine whether an abnormal event occurs. This method has achieved good results in 2018 AI CITY CHALLENGE, but there are still some problems. First of all, it uses unsupervised detectors for detection, so it definitely produce a lot of FPs and FNs. Although it removes most of the false detections by the classifier after the detection, there are still some false detections remaining. Second, it uses the ReID features to track the vehicles, which may reduce the quality of the tracks in unsupervised videos. Third, this method does not take into account the impact of the vehicles in the parking lot on the outcome.

To address these problems, we make some improvements based on the model [25]. Firstly, since the durations and space ranges of traffic abnormal events are very different, therefore, the method of characteristics reconstruction with a network such as GAN becomes infeasible based on the current computing power of computers. In fact, to solve the detection problem of complex events, it is a feasible solution of event detection in unsupervised videos based on trajectory analysis[31]. In this year, we propose an unsupervised framework for anomaly detection in traffic monitoring videos, mainly based on tracking trajectories. Secondly, according to the characteristics of background image sequence, we design a new MOT algorithm to obtain the tracks from the video background sequence. In order to compensate for the missed detection caused by the detector, a single-object tracker(SOT) is used to supplement the detection results in MOT. And considering the instability of the ReID features, we only use the position and shape features for tracking. Thirdly, in order to eliminate false anomaly detection, we propose an adaptive unsupervised method of road mask generation, which extracts the individual road mask of each video. Finally, we extract the more reliable vehicle tracks on the original video with our detector and tracker. Our detection framework of traffic anomaly can locate the start time of the anomaly event, well filter out the false detection. The flow chart of our method is shown in Figure 1

In this paper, our contributions are as follows: (1) Propose an unsupervised framework for anomaly detection in traffic monitoring videos, which includes modules for detection, tracking, and decision etc.; (2) Design a MOT method that combines SOT results for background image sequences to solve unsupervised anomaly detection problems; (3) An adaptive unsupervised road mask generation method is proposed to filter out false detection events outside the road area in traffic anomaly detection. Further details about our system will be presented in the Section 3.

2. Related Work

2.1. Unsupervised anomaly detection

For unsupervised anomaly detection, only normal datasets can be used for training [2]. Therefore there are some methods to use feature reconstruction to solve this problem, which is based on the assumption that some features of anomalous events cannot be learned from the data of normal events[34]. And [3][8][19]learns the features of normal events through deep neural networks and automatic encoders. [3][17] employ a Convolutional LSTMs Auto-Encoder (ConvLSTM-AE) to model the normal pattern, which hugely promotes the performance of methods based on CAE.

However, for traffic anomalies, the area of the accidents often accounts for a small proportion of the entire image. Therefore, it is very difficult to find the difference between the anomalies and the normal events with the convolutional neural network. [25] and [28] proposes to use background modeling and object detection methods to extract anomalous events, which better solves this problem.

2.2. Object detector

Detection networks based on deep learning have great advantages in accuracy and speed. Among them, the two-stage detector has better performance than the single-stage detector. Mainly because the two-stage detector first generate a set of region proposals and then refine them by CNN networks. In the R-CNN network[6], the proposals are extracted in advance by the selective search algorithm[24], and then each proposal is more accurately corrected by the convolutional network. Faster-RCNN[22] proposes the RPN layer, extracts the possible proposals on the feature map by selecting appropriate anchors, and then passes the ROI pooling layer to the subsequent convolution layer to learn, which greatly reduces the calculation amount and is the first end-to-end detection network. Based on this, Mask-RCNN[9] proposes ROI Align method instead of ROI pooling to further improve the accuracy of regression. FPN[13] proposes a pyramid connection structure from top to bottom. Compared with the image pyramid used by SSD[15], the lower feature maps can also obtain high-dimensional semantic features, which greatly improves the accuracy of small objects.

2.3. MOT and SOT tracker

In recent years, the single-object tracking network had a huge improvement. In particular, the SOT tracker based on the Siamese network has a good performance in terms of accuracy and efficiency. SINT[23] first proposes to use the Siamese network to obtain the similarity estimation between two frames. The SiamFC[1] network uses a full convolutional network to estimate the region-wise feature similarity between two frames. SiamRPN[12] then considers single-target tracking as a one-shot detection task in a local area, and adds a structure to extract the proposal after the Siamese network. On this basis, DaSiamRPN[32] proposes a new method for selecting the optimal bounding box.

For multi-object tracking, most of the methods are based on tracking-by-detection, and use the Hungarian algorithm or the minimum network flow algorithm to solve the matching problem between the tracks and the detections. Deep sort[27] proposes a grading strategy to build a distance matrix. [11] uses LSTM to extract the appearance and motion features for tracking. However, False Positive detections(FPs) and False Negative detections(FNs) due to detector performance and mutual occlusion between objects had a very bad effect on the results of MOT algorithm. To solve this problem, some MOT algorithms[5][30] use a SOT tracker to track the tracks in a short-term. And in [5], the author designs a quality function to control whether the SOT tracker is tracked, and achieve state-of-art results on the MOT challenge[20].

3. Method

In this section, we introduce our unsupervised anomaly detection system. As can be seen from Figure 1, our system is mainly composed of the following five parts, background modeling, detection, tracking, road modeling and the final decision module.

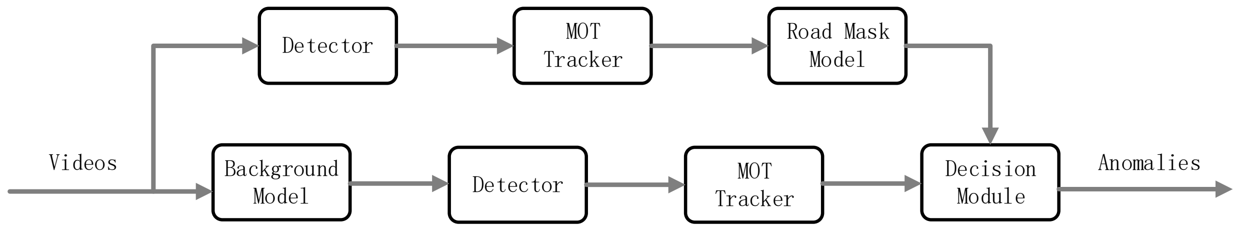

Figure 1. **Flowchart of our system.** The videos are sent to two parallel branches. The above branch generates the road mask, and the bottom obtains the tracks of candidate abomaly tracks. Then they are passed to the decision module to get the final result.

3.1. Background Model

Figure 2. **Background Modeling.** The left frames are extracted from original test video(1.mp4), and the right frame is the background image generated from the left ones.

To detect traffic anomalies in the monitoring videos, a robust tracking algorithm is necessary to track the vehicles in convention. And then vehicle trajectories are used to analyze whether vehicles are running normally or not, so as to detect abnormal vehicles. However it is difficult to obtain accurate and complete tracking trajectories since vehicles are occluded from each other on the road. It is worth noting that normal vehicles never stay on the road except for an abnormality occurring. When an abnormal event occurs, the related vehicles have to stop in scene. If the background model of video scene is constructed, the anomalous vehicles certainly appear in the background image. On the contrary normal vehicles never appear in the background image, which eliminates a large number of vehicles unrelated to traffic anomalies. Based on this essential feature of anomalous events, it is the best shortcut that we detect anomalous vehicles from the background images of the traffic videos. Like [25], MOG2[33] algorithm is used for background modeling. And for reducing the calculations, we don't perform background modeling for each frame, the videos are sampled at intervals of T.

Moreover, there is the problem of frame congestions in videos, it will cause the vehicle to stay on the background for a long time. Therefore, before performing background modeling, we calculate the inter-frame difference of videos with Equation1 and remove the video frame with small difference.

$$A_i = abs(F_{i-T} - F_i), \tag{1}$$

where F_i represents the image of the ith frame. When A_i is less than the threshold T_a, we think that the video has been stuck between the two frames. Then the ith frame is eliminated in background modeling. As show in Figure2, after such processing, the anomalies are easy to find on the background images.

3.2. Detection Module

We use Faster-RCNN[22] as our detection network and use ResNet-101[10] as the backbone of the network. And we also use the FPN[13] structure to improve the detection performance of small objects, and use the ROI-align layer [9]instead of the ROI-pooling layer. To detect abnormal vehicles as much as possible from the background images, we threshold the detection results with lower score. Further, in order to detect small abnormal vehicles in the distance, we use a sliding window to divide the image into 2*2 partial overlap areas and detect each area separately as results. Although producing many FPs, most of them will be eliminated by our tracker3.3 and decision module3.5

3.3. Tracking module

After detecting candidate abnormal vehicles, we use the MOT algorithm to track these vehicles. In convention, MOT algorithms [27][5] use the convolutional neural network to extract the ReID features from the detections to solve the occlusion problem between objects. However the ReID features rely heavily on the dataset. Therefore in the unsupervised problem, low-quality ReID features have a bad effect on the tracking. Considering that our background model has filtered out a lot of occlusion from normal vehicles, the ReID features are not necessary in our tracking algorithm, and we just use the features of position and shape to get stable tracks.

For a frame F_i of the video, let $B_i\{b_1, b_2 ... b_m\}$ be the set of all detections on the current frame, and $T_{i-1}\{t_1, t_2, ... t_n\}$ is the set of all the tracks obtained before the current frame.

And like the deep sort algorithm[30], we also assume that the tracks hav three possible states, namely 'confirmed', 'tentative' and 'deleted'.Different from the deep sort algorithm, we calculate the distance matrix M_i with the position and shape informations of detections and tracks. The distance between the detection d_j and the track t_k can be calculated by the following formula:

$$D_{j,k} = 1 - iou(b_j, t_k) * \frac{min(S_j, S_k)}{max(S_j, S_k)}, \tag{2}$$

and $iou(b_j, t_k)$ denotes intersection over union of b_j and t_k, calculated by:

$$iou(b_j, t_k) = \frac{S_{j,k}}{S_j + S_k - S_{j,k}}, \tag{3}$$

where $S_{j,k}$ denotes the intersecting area between the detection b_j and the track t_k, S_j represents the area of b_j, and S_k represents the area of t_k. We assume that Tc_{i-1} is a set of all tracks in T_i except tracks which are 'deleted' state. According to Equation2, we can construct the distance matrix M_i between B_i and T_{ci-1}, and find the most suitable pairs of tracks and detections on M_i with Hungarian algorithm[21]. When a track t_k and a detection b_j match successfully, the position of t_k is updated to the position of b_j. When a track continues n frames matched by the distance matrix, its state will be set to 'confirmed'. For the detections without matching, we think that these are newly generated tracks, the state of tracks is 'tentative', and then put them into the T_i set.

In detection of abnormal vehicles from background images, the detector may miss vehicles due to occlusion among the objects, which also affects multi-target tracking. To solve this problem we improve MOT with SOT tracker. For the track without matching, we use the DaSiamRPN model[32] to search the tracked object to find its possible position on the lost frame. But a problem of the SOT tracker is that it is hard to tell when to stop tracking if the target is lost. So like [5], a quality function is designed to control whether the SOT tracker continues to track. The function can be expressed as:

$$Q_i = \begin{cases} Score_d, & \text{if track matched by} M_i \\ Q_{i-1} * Score_s, & \text{elif } Score_s > T_s \\ 0, & \text{else} \end{cases} \tag{4}$$

where $Score_d$ indicates the score of the detection which matched by distance matrix M_i, and $Score_s$ indicates the score of the SOT when the track unmatched from the distance matrix M_i, and T_s is a threshold that removes SOT results with low scores. When Q_i is less than a threshold T_q, we set the state of the track to 'tentative'. We mark a track as 'deleted' if its states are continuosly 'tentative' for N frames. As shown in Figure3, the tracking results are more stable by fusing SOT tracker.

3.4. Road mask model

Currently, false detection of detectors is inevitable. The false objects within the scene may be incorrectly identified as abnormal vehicles. In fact, vehicles can only travel on roads, and the abnormal vehicles are also within road areas. In order to eliminate false detections outside roads with road mask, we present an unsupervised adaptive algorithm1 to generate a road mask with detection boxes of tracks in a scene.

Algorithm 1 Road Mask Model

Input:
The size $w * h$ of every frame in video,
The set of vehicles' tracks T,
The set of bounding boxs $B_t\{b_1, b_2...b_n\}$ for each $t \in T$
Output:
The road mask M

```
1:  Initialize w * h road mask M with 0
2:  for each t ∈ T do
3:      b_s = 1.5 * b_1
4:      d = D_iou(b_s, b_n)
5:      if d < 1 then
6:          continue
7:      end if
8:      for each b ∈ B_t do
9:          M[b] = M[b] + 1
10:     end for
11: end for
12: M = Norm(M)
```

In algorithm1, the set T is got from our MOT tracker without SOT. And the inputs of MOT tracker are the detections which generated by our detector in the original videos. In the detections, there must be some FNs and some vehicles which are stopped on the park. Therefore, the tracks which move very slow are eliminated at step 3 to 6. At step 3, b_s is a bounding box which width and height is 1.5 times of b_1 and has the same center with b_1. Then the IoU distance calculated by Equation:

$$D_{iou}(b_s, b_n) = 1 - iou(b_s, b_n), \tag{5}$$

where $iou()$ is the same as Equation3. Then tracks whose iou distance is lower than 1 are eliminated. For those remaining tracks, the area contained in the bounding box of each frame is plus by one on M. Finally to indicating the posibility whether one pixel of mask image M is belong to scene road, M is normalized using the median of M, which can be expressed as:

$$Norm(M) = min(\frac{M}{median(M)}, 1), \tag{6}$$

(a)

(b)

Figure 3. **MOT results.** (a) and (b) are the tracking results of two different MOT algorithms. (a) is the result of using SOT in MOT algorithm. And (b) is the result without using SOT.

where function $median()$ represents the median of non-zero values in matrix M. And Figure4 shows the road masks we have modeled by this method.

Figure 4. Some road masks got from our algorithm. The left images are the road mask of the right videos.

During modeling the road masks, the possible sizes of the vehicles is recorded at each pixel. Anomalies outside this size range are eliminated by the decision module3.5. By this way, we further filter out false detections such as large traffic signs.

3.5. Decision module

In this decision model, we can get the exact time when the anomaly occurrs by the tracks and the road masks which we obtain. The overall algorithm can be summarized as follows:

- Step 1. Select a track t from the set T which get from 3.3. If the set T is empty, execute the Step 7.
- Step 2. If the length of t is less than L, remove t from T, return to the first step, if not, execute Step 3.
- Step 3. Suppose the starting position of t is b_s and the ending position is b_e. Then we initialize two different SOT trackers with b_s and b_e as template frames, respectively, denote them as S_s and S_e. Then we use S_s and S_e to track temporal forward and backward on the original video for the track t. For each SOT tracker, we judge whether its starting position is on the road by our road mask getting from 3.4. If the position is on the road, execute Step 4, otherwise, execute Step 5.
- Step 4. Select a region A centering on the starting position of the SOT tracker. Keep tracking by SOT tracker, and when the center of the track position is not in the area A or the track is interrupted, the tracking is stopped. Record the stopped frame, and execute Step 6.
- Step 5. Keep tracking by SOT tracker, and when the track position is on the road, record the stopped frame. And if the track is stopped before returning to the road, record the stopped frame as -1. Execute Step 6.
- Step 6. If the stopped frames returned by S_s and S_e aren't both less than 0, and the length of the updated track is lower than the length of video, put it into set T_c. Return Step 1.
- Step 7. Merge all tracks in set T_c on the timeline until the interval between any two tracks is less than 2 minutes Then the startting frame of all remaining tracks is taken as the start time of the anomaly.

In Step 2, we remove the tracks whose length is less than L. One purpose is to eliminate a small amount of FPs, and another is that we want to remove some events as traffic jams and vehicles waiting for traffic lights. And the SOT trackers S_s and S_e are used for forward and backward tracking of the tracks in Step 5 and Step 6. This is because the tracks obtained from background images do not represent the true duration of an anomaly, so we track them on the original videos to find the exact start time of anomalies. And a function f is designed to determine if the trajectory is on the road, as Equation 7.

$$f(b_i) = \frac{sum(b_i)}{S_{b_i}}, \tag{7}$$

where b_i denotes the track's bounding box, and $sum()$ denotes the sum of the values of the points included in b_i on the road mask. And S_{b_i} denotes the area of b_i. When $f(b_i)$ is greater than threshold T_f, it represents that b_i is on road. When the track obtained from background images isn't on the road mask, it is considered that the anomaly has just started or has ended if this track can return to the road by SOT tracker, otherwise it is a FP by detector or the vechiles stop on the park. When the trajectory got from background imges is on the road, as [25], we define an area to judge the start and end time of the anomaly.

At the same time, when using SOT tracker, like3.4 a quality function is set for the tracker to determine whether the tracker results are correct. However, unlike 3.4, since the detection result is not used for tracking, when the tracker score is greater than 0.7, we set the quality to 1.

4. Experiments

In this section, we first introduce the evaluation dataset. Then, we describe the implement details. Finally, we present the performance of our method in the Challenge dataset.

We test and evaluate our system on the Track 3 testing set of 2019 AI CITY CHALLENGE. It aims to detect anomalies in traffic monitor videos. The Track 3 dataset contains 100 training and 100 test videos, each approximately 15 minutes in length, recorded at 30 fps and $800 * 410$ resolution. The test dataset contains the real scene videos with diverse scenes, light condition, weather. Therefore, it is a quite challenging dataset.

4.1. Implement details

In this section, we introduce our implement details and show some results of 2019 AI CITY CHALLENGE Track 3 test dataset in Figure5.

Data processing. Instead of performing background modeling with each frame of the video, we sample 1 frame every 10 frame as input to the background modeling algorithm, and then extract the background image every 30 input frames. And, when calculating the interframe difference, the threshold T_a is set to $2 * 10^6$.

Detector. We use the open source Mask-Rcnn code[18] and trun off the mask path to train our detectors. The model is trained on 2*1080Ti and the number of batch size is set to 2. We used UA-DETRAC[26] and partially hand-labeled YOUTUBE video as the training and validation dataset. The Gaussian filtering and brightness-changing method also processed on the UA-DETRAC dataset, as shown in the Figure6. In the validation dataset our model can reach 90.5% mAP.

MOT and SOT Tracker. We make some changes on the open source Deep SORT and DaSiamRPN code. We use the pre-trained DaSiamRPN model on OTB dataset[29] in our experiment. In MOT tracker, if the distance between the detection and the track is greater than 0.9, the distance value is removed in distance matrix M. The quality function threshold T_q of SOT is 0.3 and T_s set 0.1. When the score of the quality function is lower than 0.3, the state of the trajectory is set to 'tentative'. When a track is matched for 3 consecutive frames, its status will become 'confirmed'. If a track does not match for 20 consecutive frames, its status is updated to 'deleted'.

Road mask. To generate the road mask, the detections and tracks are obtained by our detector and MOT tracker. When obtaining the vehicle detections of the original video vehicle, we used pre-trained Faster-RCNN model on the coco dataset[14]. And we select the detections which label in car, truck, bus and van and score greater than 0.7. SOT tracker is used to track 10 frames for each detection, and both results are processed using NMS with a threshold of 0.3. Then in MOT algorithm, we simply use the IOU distance to get the tracks. Final, the masks are generated by these tracks.

Decision model. In the decision model, we think that the shortest length L of the track is 90. When judging whether the track is on the road, the threshold T_f is set to 0.9. Then unlike multi-target tracking, the threshold T_q of the SOT quality function is used 0.1. The score of anomaly is represented by the quality score of the trajectory.

4.2. Evaluation on Track 3 testing set

Evaluation for the Track 3 testing set 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:

$$S3 = F1 * (1 - NRMSE), \tag{8}$$

Here, the detection time error is the RMSE between the ground truth anomaly time and predicted anomaly time for all TP predictions. NRMSE is the normalized RMSE score across all teams, obtained via min-max normalization given all team submissions.

Table 1. Our result on Track3 testing set.

	F1	RMSE	Local S3
Our result	0.7164	24.2689	0.6585

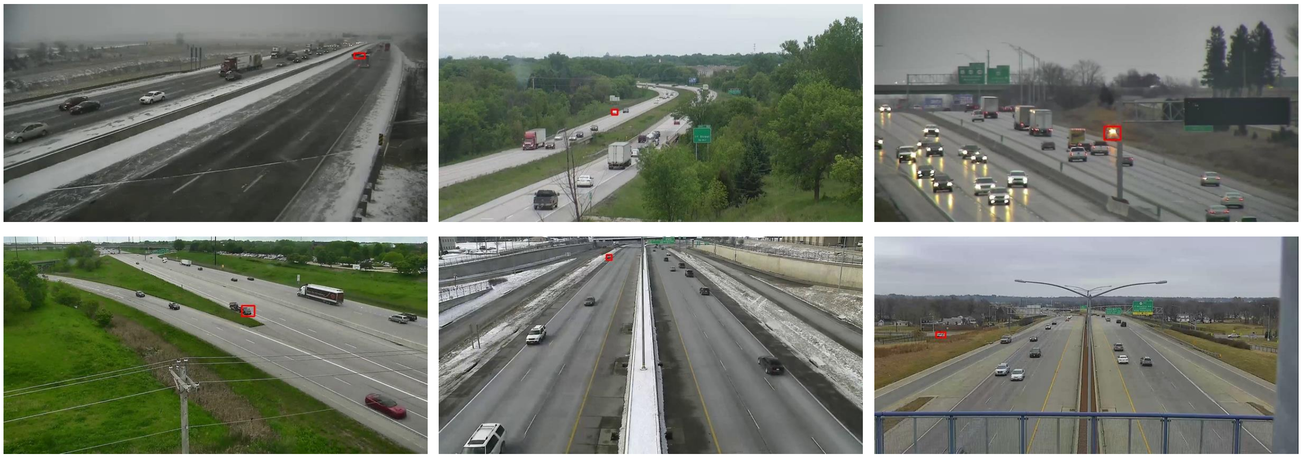

Figure 5. Some anomalies detected by our system on 2019 AI CITY CHALLENGE Track3 test dataset.

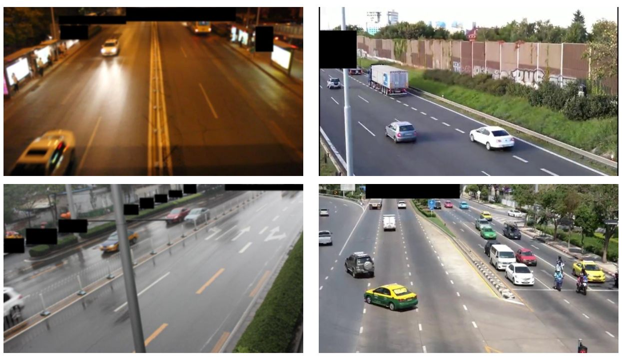

Figure 6. The left two images are the samples of UA-DETRAC dataset and the right two images are the samples of YOUTUBE dataset.

We evaluate our method on the Track 3 testing data and obtain the best result as shown in Table 1. As you can see, we achieve 0.7164 F1-score while detection time error is only 24.2689 seconds, which demonstrates our proposed methods superiority and robustness. Local S3 score is obtained to 0.6585 by Equation 8.

5. Conclusions

In this paper, we propose a framework to determine whether an abnormal event has occurred based on the tracks in unsupervised traffic anomaly detection task. In our method, we first obtain the tracks from the background images of videos with Faster-RCNN and MOT algorithms, and then adaptively generate the road mask for each video. Finally, we combine these information to determine the possibility of anomalies by our decision model and get the precise start time of the anomalies. In the 2019 AI CITY CHALLENGE we achieved the 7th results.

6. Acknowledgments

This work is supported by National Key R&D Program of China (2017YFC0803804).

References

[1] Luca Bertinetto, Jack Valmadre, Joo F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, 2016.

[2] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *Acm Computing Surveys*, 41(3):1–58, 2009.

[3] 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.

[4] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456. IEEE, 2011.

[5] Weitao Feng, Zhihao Hu, Wei Wu, Junjie Yan, and Wanli Ouyang. Multi-object tracking with multiple cues and switcher-aware classification. 2019.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2014.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[8] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 733–742. IEEE, 2016.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[11] Chanho Kim, Fuxin Li, and James M. Rehg. Multi-object tracking with neural gating using bilinear lstm. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[12] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[13] Tsung Yi Lin, Piotr Dollr, Ross Girshick, Kaiming He, and Serge Belongie. Feature pyramid networks for object detection. 2016.

[14] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. 2014.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, 2016.

[16] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2720–2727. IEEE, 2013.

[17] Weixin Luo, Wen Liu, and Shenghua Gao. Remembering history with convolutional lstm for anomaly detection. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, pages 439–444. IEEE, 2017.

[18] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark`, 2018.

[19] Jefferson Ryan Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

[20] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, Mar. 2016. arXiv: 1603.00831.

[21] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1):32–38, 1957.

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

[23] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders. Siamese instance search for tracking. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2016.

[24] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[25] Jia Yi Wei, Jian Fei Zhao, Yan Yun Zhao, and Zhi Cheng Zhao. Unsupervised anomaly detection for traffic surveillance based on background modeling. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[26] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv CoRR*, abs/1511.04136, 2015.

[27] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple Online and Realtime Tracking with a Deep Association Metric. *arXiv e-prints*, page arXiv:1703.07402, Mar 2017.

[28] 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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[29] Wu Yi, Lim Jongwoo, and Yang Ming-Hsuan. Object tracking benchmark. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(9):1834–1848, 2015.

[30] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, and Ming Hsuan Yang. Online multi-object tracking with dual matching attention networks. 2019.

[31] Yandong Zhu, Kaihui Zhou, Menglai Wang, Yanyun Zhao, and Zhicheng Zhao. A comprehensive solution for detecting events in complex surveillance videos. *Multimedia Tools and Applications*, 78(1):817–838, 2019.

[32] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. 2018.

[33] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.

[34] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.