Anomaly Candidate Identification and Starting Time Estimation of Vehicles from Traffic Videos

Gaoang Wang[1], Xinyu Yuan[1], Aotian Zhang[1], Hung-Min Hsu[1,2], and Jenq-Neng Hwang[1]

[1] Department of Electrical and Computer Engineering, University of Washington
[2] Research Center for Information Technology Innovation, Academia Sinica, Taiwan
{gaoang, xyuan, hmhsu, hwang}@uw.edu, aotianzheng@gmail.com

Abstract

Anomaly event detection on road traffic has been a challenging field mainly due to lack of training data and a wide variety of anomaly cases. In this paper, we propose a novel two-stage framework for anomaly event detection in road traffic based on anomaly candidate identification and starting time estimation of vehicles. First, we use Gaussian mixture models (GMMs) to generate the foreground mask and background image to identify the anomaly candidates. Foreground mask is used to produce the region of interest (ROI) to filter out the noise from the object detector, YOLOv3, in the background image. Then, we apply the TrackletNet Tracker (TNT) to extract the trajectory of anomaly candidate to estimate the anomaly starting time. Experimental results, with achieved S3 score performance of 93.62%*, on the Track 3 testing set of CVPR AI City Challenge 2019 City Flow dataset, show the effectiveness of the proposed framework and its robustness in different real scenes.*

1. Introduction

Due to the rapid development of computation power, using computer vision techniques for urban traffic optimization receives great attention in recent years. The bad road conditions can be reduced if damage control is efficient. However, anomaly detection in road traffic has been a challenging task. For example, there are only rare labeled anomaly videos in road traffic for training. In addition, a large amount of different complicated anomaly cases is another reason to make the difficulty of anomaly detection. Therefore, it is necessary to develop a method to automatically detect the anomalies on the roads based on the traffic videos.

Previous works of anomaly detection in videos are only for specific purpose [24]. In this work, we propose a two-stage framework for anomaly detection in road traffic, as shown in Figure 1, which performs anomaly candidate identification and starting time estimation of vehicles from traffic videos. For anomaly candidate identification, we use Gaussian mixture models (GMMs) [22, 10] to obtain the foreground and background of each frame. Then the foreground images of each video are added up to generate a traffic flow mask which is treated as the region of interest (ROI) for vehicle detection. For each frame, the vehicles that lie in both the background regions and ROI are detected as anomalies, since they are non-moving vehicles on the main traffic road. Here, YOLOv3 is used as the vehicle detector in the experiment. After the anomaly candidates are detected, two branches are used to determine the starting time of the anomaly in the second stage. First is to detect the exact starting time of small vehicles which are far away from the camera with an unsupervised approach adopted from foreground object extraction. For other cases, based on the detection results, a multi-object tracking (MOT) approach, called TrackletNet Tracker (TNT) is adopted for detection associations among sequential frames. Then the whole trajectory of the anomaly vehicle can be obtained. We formulate the starting time estimation to a curve fitting problem based on the computer vision geometry. A rough starting time is estimated according to the fitting errors. Afterward, the starting time is refined by checking the bounding box overlap along the time of the trajectory. To summarize, we claim the following contributions,

- A novel and effective anomaly detection approach is proposed based on GMMs and automatic ROI generation.
- An effective TNT tracker is used to obtain the anomaly trajectory.
- We formulate the starting time estimation as a two-stage curve fitting problem.

The rest of this paper is organized as follows. We provide an overview of related works in Section 2 and our proposed anomaly detection approach is introduced in Section 3. The experiments and evaluations of our method on the CVPR AI City Challenge 2019 City Flow dataset [25] are shown in Section 4. Finally, the conclusion is drawn in Section 5.

2. Related Works

2.1. Vehicle Detection.

With the fast development of deep learning, convolutional neural network (CNN) based detectors have shown great power in object detection in recent years. Generally, there are two categories of CNN based detectors, i.e., one-stage and two-stage approaches. For one-stage approaches, such as [16, 17, 18, 11], the input images are divided into different small regions or anchor boxes at different scales. Then the classification and regression are processed for each anchor box. Different from one-stage approaches, two-stage approaches [7, 6, 19, 4, 9] adopt region proposal networks (RPN) to detect objects of the input images. Then the feature maps of detected ROI regions are pooled and used for further classification and regression.

In addition to CNN based detectors, background subtraction for vehicle detection is also very useful in static cameras and anomaly detection. For example, [22, 10] use Gaussian mixture models (GMMs) for background subtraction. Similarly, [30, 28] also use GMMs as a pre-processing step for vehicle detection in the anomaly detection task.

2.2. Deep Learning based Anomaly Detection.

Recently, deep learning technologies have been developed for the anomaly detection in computer vision fields [20, 21]. For example, [20] uses cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes. In [21], the authors introduce a generative adversarial network (GAN) [8] based method to detect the anomalies in images, using only normal data to train the models. In addition to traffic videos, there are several attempts to detect human violence or abnormal events in crowd scenes, such as [13, 15, 23, 29, 12]. In [23], a deep anomaly ranking model is proposed to predict high anomaly score in the testing videos. In [29], a double fusion framework, combining the benefits of traditional early fusion and late fusion strategies is introduced to exploit the complementary information of both appearance and motion patterns. In [12], an end-to-end trainable composite convolutional long short-term memory (Conv-LSTM) network is proposed to predict the evolution of a video sequence from a small number of input frames. With the help of neural networks, the anomaly can indeed be detected from end-to-end. However, large scale training anomaly datasets are very difficult to be obtained.

2.3. Tracking aided Anomaly Detection.

To precisely know when and where an anomaly occurs in the videos, object tracking techniques can be also adopted in the anomaly detection [30, 3]. In [30], velocity information is extracted from tracking. When the velocity suddenly changes, there is a high chance that an anomaly happens. In [3], trajectory analysis is performed for anomalous behavior detection from static cameras. Usually, large scale labeled dataset is not required in tracking aided anomaly detection, and it can easily be transferred to other scenarios, which is also one major advantage of this type of anomaly detection.

3. Proposed Method

3.1. Anomaly Vehicle Candidate Identification

First, Gaussian mixture models (GMMs) are applied for background modeling so that we can generate the background image and foreground mask for each frame from a video sequence. Then, we add up all the foreground masks from one video to generate a traffic flow mask. Based on the traffic flow mask, we produce a region of interest (ROI) to remove unreliable detections. The detected vehicles in the background image usually are anomaly candidates since they are non-moving vehicles. The framework of the anomaly candidate identification is shown in the left part of Figure 1 and the details of the methods are described in the following part of this subsection.

Background Modeling. Using GMMs, we can build the background of a video in pixel level. More specifically, each pixel in the video frame during a chosen time period, we can estimate the density of pixel intensity using a mixture of Gaussians with K components as follows,

$$p(x_t) = \sum_{k=1}^{K} w_{k,t} \mathcal{N}(x_t | \mu_{k,t}, \sigma_{k,t}^2), \tag{1}$$

with

$$\mathcal{N}(x_t | \mu_{k,t}, \sigma_{k,t}^2) = \frac{1}{\sigma_{k,t}\sqrt{2\pi}} \exp\left(-\frac{(x_t - \mu_{k,t})^2}{2\sigma_{k,t}^2}\right), \tag{2}$$

where $\mu_{k,t}$ is the mean, $\sigma_{k,t}^2$ is the variance of the k-th Gaussian component and $w_{k,t}$ is non-negative estimated weights which add up to one.

When given a new sample x_{t+1} that can be matched to one of the existing Gaussian, then the weight, mean and variance of that Gaussian are updated according to the following equations [22],

$$\begin{aligned} \mu_{k,t+1} &= (1-\rho)\mu_{k,t} + \rho x_{t+1}, \\ \sigma_{k,t+1}^2 &= (1-\rho)\sigma_{k,t}^2 + \rho(x_{t+1} - \mu_{k,t+1})^2, \end{aligned} \tag{3}$$

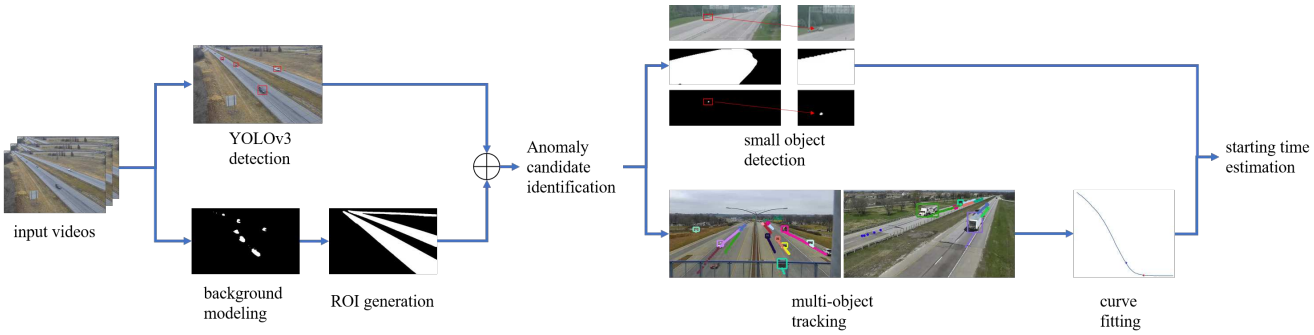

Figure 1. A two-stage framework for anomaly detection in road traffic. In the first stage, anomaly candidate identification, we use GMMs to obtain the foreground and background of each frame. Then the foreground images of each video are added up to generate a traffic flow mask which is treated as ROI for vehicle detection. For each frame, the vehicles that lie in both the background regions and ROI are detected as anomalies, since they are non-moving vehicles on the main traffic road. After that, in the second stage, the TNT is adopted for detection associations among sequential frames. Then the whole trajectory of the anomaly vehicle can be obtained. Finally, the starting time is estimated by solving a curve fitting problem based on computer vision geometry.

where $\rho = \alpha\mathcal{N}(x_{t+1}|\mu_{k,t}, \sigma^2_{k,t})$, α is a learning rate. The prior weights of all Gaussians are adjusted as follows,

$$w_{k,t+1} = (1-\alpha)w_{k,t} + \alpha M_{k,t+1}, \tag{4}$$

where $M_{k,t+1} = 1$ for the matching Gaussian and $M_{k,t+1} = 0$ for all the others.

We then order the Gaussians by the value of $w_{k,t}/\sigma_{k,t}$ and the first B distributions with high supporting evidence and least variance are chosen as the background,

$$B = \arg\min_b \left(\sum_{k=1}^{b} w_{k,t} > P\right), \tag{5}$$

where P is the minimum portion of the image which is expected to be the background. Then we define the background probability as

$$P_{\text{bg}} = \frac{\sum_{k=1}^{B} w_{k,t}\mathcal{N}(x_t|\mu_{k,t}, \sigma^2_{k,t})}{\sum_{k=1}^{K} w_{k,t}\mathcal{N}(x_t|\mu_{k,t}, \sigma^2_{k,t})}. \tag{6}$$

Then the background mask $I_{bg}(x_t)$ of pixel x_t at time t can be described as,

$$I_{\text{bg}}(x_t) = \mathbf{1}_{\{P_{\text{bg}} > 0.5\}}, \tag{7}$$

where $\mathbf{1}_{\{\cdot\}}$ is an indicator function which outputs 1 if the condition satisfies otherwise outputs 0.

Traffic Flow Mask Generation. In order to automatically get regions of interest (ROI), we add up all the foreground masks of a video to generate the final traffic flow mask, i.e.,

$$I_{\text{ROI}} = \mathbf{1}_{\{\frac{1}{T}\sum_{t=1}^{T}(1-I_{\text{bg},t}) > \gamma\}}, \tag{8}$$

Figure 2. Examples of background modeling and traffic flow mask. From the top row to the bottom row: original video frames, foreground masks after background subtraction, traffic flow masks, respectively.

where γ is a pre-defined threshold for ROI generation. To eliminate noise, median filters are adopted. Some examples of background modeling and traffic flow mask are shown in Figure 2.

Anomaly Candidate Identification. For each frame, if the detected vehicle lies in the background region and also in our generated ROI, then this vehicle is very likely to be an anomaly case because it is a non-moving vehicle. We define the detection ROI at time t as the element-wise mul-

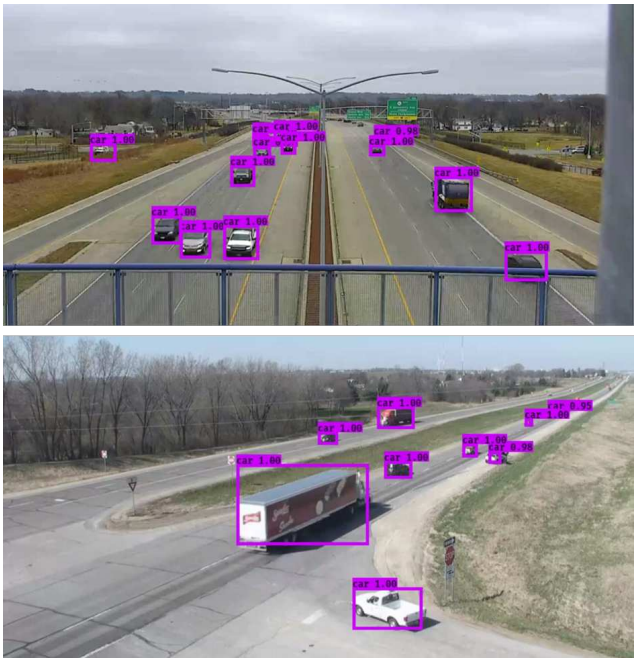

Figure 3. Examples of YOLOv3 detection results.

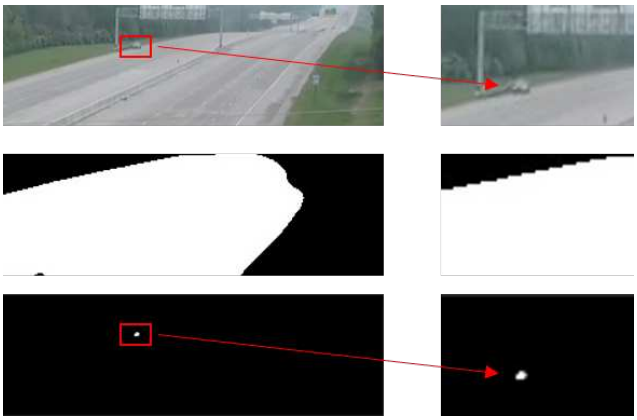

Figure 4. Examples of small vehicle detection. Top row: an example of an anomaly vehicle far away from the camera. Middle row: the generated traffic flow mask. Bottom row: the detected anomaly vehicle using frame subtraction. The figures in the right column are the zoom-ins of the figures in the left column.

tiplication between I_{ROI} and $I_{\text{bg},t}$, i.e.,

$$I_{\text{det},t} = I_{\text{ROI}} \cdot I_{\text{bg},t}, \tag{9}$$

Then the detections that lie in the $I_{\text{det},t}$ mask are treated as anomaly candidates. Some examples of detection results from the pre-trained YOLOv3 detector are shown in Figure 3.

3.2. Anomaly Starting Time Estimation

In this subsection, we describe how to estimate the anomaly starting time with two separate branches. One is based on small object detection. In some cases, Small objects cannot be easily detected by the trained YOLOv3 detector. As a result, an unsupervised approach is adopted for small object detection. The first frame that the small objects are detected is treated as the anomaly starting time. For the second branch, a curve fitting based approach is proposed. First, we perform multi-object tracking using the pre-trained TrackletNet Tracker (TNT). Then the trajectories are obtained for each anomaly candidate. From the computer vision geometry, each trajectory with a constant moving velocity can be represented by an inversely proportional linear function. The anomaly starting time can be estimated by optimizing a curve fitting problem. The details of each part are demonstrated as follows.

Small Object Detection. In some cases, some anomaly vehicles that are far away from the camera are not detected by the YOLOv3 detector. In that case, an unsupervised approach is adopted to detect objects that are far away from the camera. Here, we use frame t to subtract the frame $(t+m)$ in the background region to see if there are strong values, i.e.,

$$\Delta I_{t+m} = g(|I_t - I_{t+m}| \cdot I_{\text{det},t+m}), \tag{10}$$

where $g(\cdot)$ represents median filter operations and m is an empirically selected duration between two frames. Then high responses from ΔI_{t+m} are also identified as anomaly candidates. If there is no small anomaly vehicle in a video, the background pixel value varies little, and the difference between these two pixels should be close to 0. However, if there is an anomaly vehicle appearing in the background, the difference of background pixel of the current frame t and frame $t+m$ will be much bigger than 0. Examples of small vehicles detection are shown in Figure 4. In this case, the first frame that the small objects are detected is treated as the anomaly starting time.

Multi-Object Tracking. We adopt the TrackletNet Tracker (TNT) [27] in Track 3 to extract the trajectories of anomaly vehicles. The tracking system is based on a tracklet graph-based model, as shown in Figure 5, which has three key components, 1) tracklet generation, 2) connectivity measure, and 3) graph-based clustering. Given the detection results in each frame, the tracklets are generated based on the intersection-over-union (IOU) and the appearance similarity between two adjacent frames, which are treated as the nodes in the graph. Between every two tracklets, the connectivity is measured as the edge weight in

the graph model, where the connectivity represents the likelihood of the two tracklets being from the same object. To calculate the connectivity, a multi-scale TrackletNet is built as a classifier, which can combine both temporal and spatial features in the likelihood estimation. Clustering [26] is then conducted to minimize the total cost on the graph. After clustering, the tracklets from the same ID can be merged into one group. Examples of TNT on Track 3 are shown in Figure 6.

The reason we use TNT as our tracking method is due to its robustness dealing with occlusions. More specifically, 1) TrackletNet focuses on the continuity of the embedded features along the time. In other words, the convolution kernels only capture the dependency along time. 2) The network integrates object Re-ID, temporal and spatial dependency as one unified framework. Based on the tracking results from TNT, we know the continuous trajectory of each detected object ID across frames.

Starting Time Estimation by Curve Fitting. For each anomaly vehicle candidate, we can extract the whole trajectory from the results of TNT. Then we can check the 2D bounding box overlap to calculate the anomaly starting time t_{st} by,

$$\begin{aligned} \hat{t}_{\text{st}} &= \arg\min_t t, \\ s.t., &\frac{\text{BBox}(t+\tau) \cap \text{BBox}(t)}{\text{BBox}(t+\tau) \cup \text{BBox}(t)} > \delta, \end{aligned} \tag{11}$$

where $\text{BBox}(t)$ is the bounding box of the candidate vehicle at time t, τ is the time interval for calculating the difference and δ is the IOU threshold for identifying whether the vehicle is stopping. However, there are two drawbacks of this approach, i.e., 1) the performance is easily affected if the vehicle is far away from the camera in some frames. This is because the vehicle is moving quite slowly in the 2D image coordinate even before the anomaly happens if it is far away from the camera. 2) The noise from the detection and tracking also makes the criteria not very reliable. As a result, we propose a two-stage curve fitting approach for choosing the accurate starting time in the following subsection.

From the computer vision geometry, the relation between the 3D coordinate and the 2D image coordinate can be written as,

$$\begin{aligned} x &= fX/Z + x_c, \\ y &= fY/Z + y_c, \end{aligned} \tag{12}$$

where (X, Y, Z) is the 3D coordinate, (x, y) is the 2D image coordinate, (x_c, y_c) is the image center and f is the focal length. If the vehicle is moving with a constant speed, i.e.,

$$\begin{aligned} X(t) &= a_x t + b_x, \\ Y(t) &= a_y t + b_y, \\ Z(t) &= a_z t + b_z, \end{aligned} \tag{13}$$

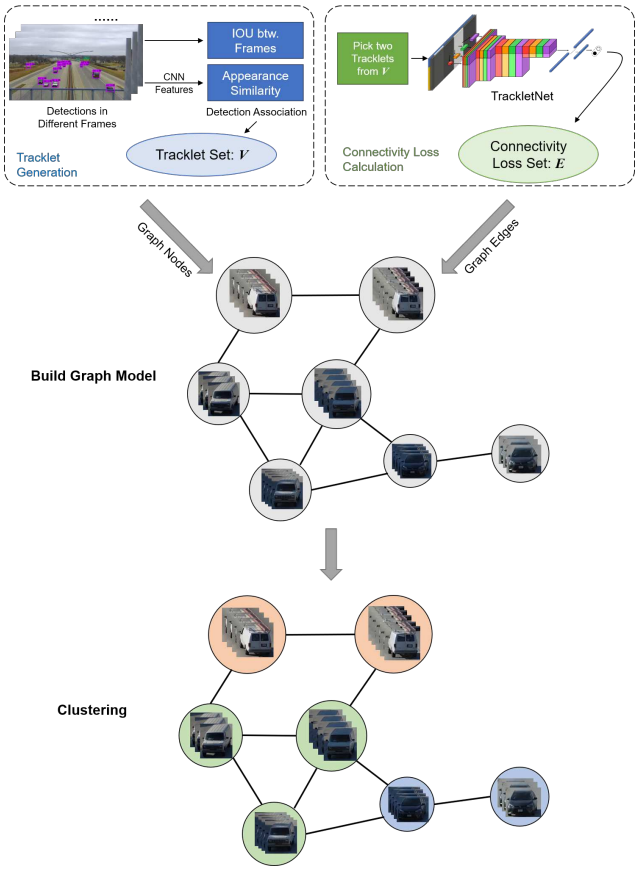

Figure 5. The TNT framework for multi-object tracking. Given the detections in different frames, detection association is computed to generate Tracklets for the Vertex Set V. After that, each pair of two tracklets are put into the TrackletNet to measure the connectivity, which formed the similarity on the Edge Set E. A graph model G can be derived from V and E. Finally, the tracklets with the same ID are grouped into one cluster using the graph partition approach.

then

$$\begin{aligned} x(t) &= \frac{f(a_x t + b_x)}{(a_z t + b_z)} + x_c, \\ y(t) &= \frac{f(a_y t + b_y)}{(a_z t + b_z)} + y_c, \end{aligned} \tag{14}$$

which means the 2D trajectory can be represented by an inversely proportional linear function as,

$$\begin{aligned} x(t) &= \frac{1}{a_1 t + b_1} + c_1, \\ y(t) &= \frac{1}{a_2 t + b_2} + c_2, \end{aligned} \tag{15}$$

where a_1, b_1, c_1 and a_2, b_2, c_2 are the parameters that need to be estimated.

For each trajectory, we formulate the following loss

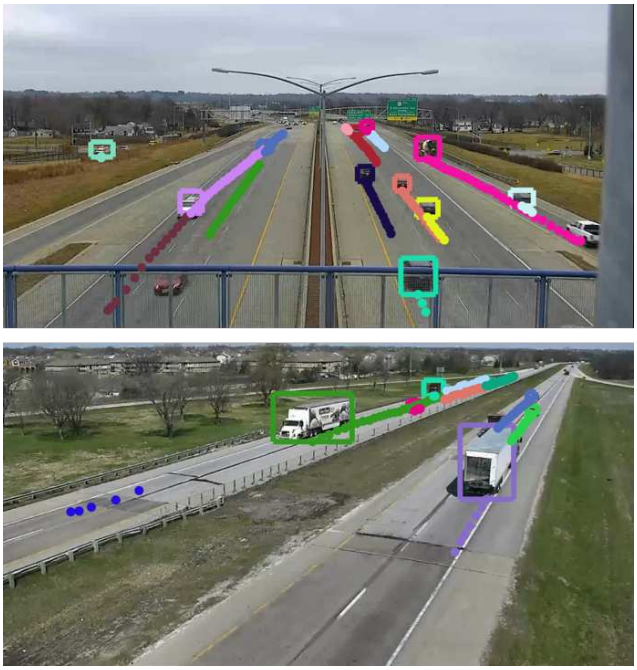

Figure 6. Examples of multi-object tracking with TNT. Each color represents a unique tracked vehicle ID. Dots are vehicle locations detected in the previous frames.

function,

$$\begin{aligned}&\mathcal{L}(a_1, b_1, c_1, a_2, b_2, c_2)\\&= \sum_t \|x_t - (\frac{1}{a_1 t + b_1} + c_1)\|^2\\&+ \|y_t - (\frac{1}{a_2 t + b_2} + c_2)\|^2,\end{aligned} \tag{16}$$

where (x_t, y_t) are the observed 2D coordinates from the tracking results at frame index t. To solve this curve fitting problem, we follow the assumption that the vehicle should have a constant speed in the 3D coordinate. As a result, we need to select the samples roughly before the anomaly happens for the curve fitting. Here we use a simple rule that we exclude the non-moving samples which can be inferred from the trajectory for minimizing the loss function based on the generated background masks. After that, RANSAC [5] algorithm is adopted to further remove outliers for the optimization.

When the anomaly happens, the velocity of the candidate vehicle should have an abrupt change, which will cause a large fitting error with the estimated curve function. As a result, a rough starting time t_1 from the first stage is estimated by the following criteria,

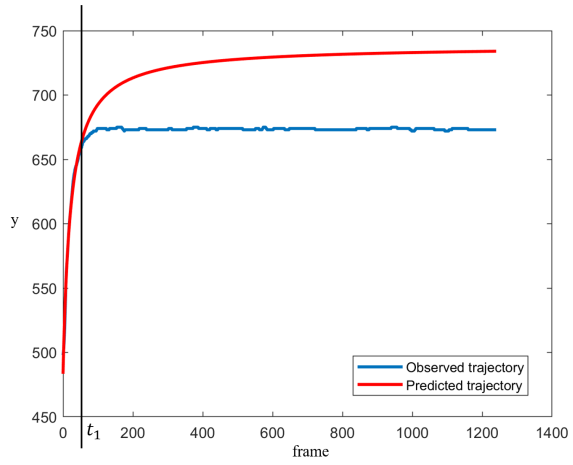

Figure 7. An example of the curve fitting.

$$\begin{aligned}\hat{t}_1 = \arg\max_{t_1} |&\frac{1}{t_1}\sum_{t=1}^{t_1}(\|x_t - (\frac{1}{a_1 t + b_1} + c_1)\|^2\\&+ \|y_t - (\frac{1}{a_2 t + b_2} + c_2)\|^2)\\&- \frac{1}{(N - t_1)}\sum_{t=t_1+1}^{N}(\|x_t - (\frac{1}{a_1 t + b_1} + c_1)\|^2\\&+ \|y_t - (\frac{1}{a_2 t + b_2} + c_2)\|^2)|,\end{aligned} \tag{17}$$

where N is the length of the trajectory. The above equation means that we want to find a time threshold that makes the fitting error before the threshold to be small and the fitting error after the threshold to be large. One example of the curve fitting is shown in Figure 7.

After estimating a rough starting time from above, in the second stage, we estimate the final starting time t_{st} based on the bounding box overlap using the following equation,

$$\begin{aligned}&\hat{t}_{\text{st}} = \arg\min_t |t - \hat{t}_1|,\\&s.t., \frac{\text{BBox}(t+\tau) \cap \text{BBox}(t)}{\text{BBox}(t+\tau) \cup \text{BBox}(t)} > \delta,\end{aligned} \tag{18}$$

which is the final output of our proposed method.

4. Experiments and Results

4.1. Datasets.

The benchmark dataset [1] contains 100 training and 100 testing videos, each approximately 15 minutes in length, recorded at 30 fps and 800×410 resolution. The types of anomalies include vehicle stopping on the road, vehicle losing control and crushes, vehicle crushing with another

Figure 8. Examples of the dataset used for training the YOLOv3 detector.

vehicle, vehicle going off the road and going into the grass.

4.2. Implementation Details.

To balance the performance and the time efficiency of our system, we use YOLOv3 as our vehicle detector. A combination of datasets is used for training YOLOv3. These datasets include labeled videos in the training set of AI City Challenge 2019 City Flow dataset [25], training data from AI City Challenge 2018 [14], traffic flow videos from high ways [2] and also VisDrone dataset [31]. Examples are shown in Figure 8.

Figure 9. Examples of detected anomalies. Left column: the original video frames. Right column: the corresponding processed video frames with detected anomalies shown with red bounding boxes from background regions. Other vehicles are smoothed out using our proposed method.

4.3. Evaluation and Results.

We show some qualitative results in Figure 9. The figures in the left column are from the raw videos. The corresponding right figures are the processed video frames with only the anomaly vehicles remained in the image.

For quantitative results, as described on [1], the evaluation is based on anomaly detection performance, measured by the F1-score, and detection time error, measured by normalized root mean square error (NRMSE). Specifically, the final score S3 is defined as,

$$\mathrm{S3} = \mathrm{F1}(1 - \mathrm{NRMSE}). \tag{19}$$

To compute the F1-score, a true-positive (TP) detection is considered as the predicted anomaly within 10 seconds of

Rank	Team ID	Team Name	S3 Score
1	12	Traffic Brain	0.9534
2	**21**	**UWIPL**	**0.9362**
3	66	Spartans	0.8504
4	53	Desire	0.7598
5	24	Avengers5	0.7562
6	79	Alpha	0.6997
7	48	BUPT-MCPRL	0.6585
8	113	HCMUS	0.6129
9	36	DGRC	0.4337
10	158	TITAN LAB	0.4083

Table 1. The final ranking and S3 score on Track 3. Our team is shown in bold type.

F1	RMSE	S3 Score
0.9577	6.7461	0.9362

Table 2. The F1 sore and RMSE of the proposed method on the testing data.

the true anomaly (i.e., seconds before or after) that has the highest confidence score. Each predicted anomaly will only be a TP for one true anomaly. A false-positive (FP) is a predicted anomaly that is not a TP for some anomaly. Finally, a false-negative (FN) is a true anomaly that was not predicted. Hence, the F1-score is measured by,

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{20}$$

We compute the detection time error as the RMSE of the ground truth anomaly time and predicted anomaly time for all TP predictions. In order to eliminate jitter during submissions, normalization is done using min-max normalization with a minimum value of 0 and a maximum value of 300, which represents a reasonable range of RMSE values for the task. Teams with RMSE greater than 300 will receive an of 1, and thus a score of 0. Hence, NRMSE is defined as,

$$\mathrm{NRMSE} = \frac{\min\left\{\sqrt{\frac{1}{\mathrm{TP}}\sum_{i=1}^{\mathrm{TP}}(t_i - t_i^{\mathrm{GT}})^2}, 300\right\}}{300}. \tag{21}$$

where t_i^{GT} is the ground truth starting time of the anomaly, t_i is the submitted starting time. The final ranking and S3 score are shown in Table 1. We can see that our team is in rank 2, as shown with bold type, which demonstrates the effectiveness of our proposed method. The F1 measure and RMSE of the proposed method are shown in Table 2.

5. Conclusion

In this paper, we propose a novel and effective two-stage approach for anomaly candidate identification and starting time estimation. For the anomaly candidate identification, an unsupervised approach is used to generate the ROI for anomaly detection. Besides, we can even detect vehicles far away from the camera using the background subtraction method. In the second stage, the anomaly starting time is measured with the help of the proposed curve fitting approach. In the evaluation, our proposed method achieves rank 2 place in the challenge and shows promising performance compared with other methods.

Acknowledgement The authors would like to thank many people who helped in the improvement of the performance of the proposed system: Haotian Zhang, Zhichao Lei, Xinyu Zhao, Kelvin Lin, Charles Tung Fang, Huihao Chen, Zexin Li, and also would like to thank the STAR Lab at University of Washington [2] for providing traffic flow dataset for training.

References

[1] Ai city challenge 2019 official website. `https://www.aicitychallenge.org`. Accessed: 2019-02-08.

[2] Website of star lab at university of washington. `https://www.uwstarlab.org`.

[3] Yingfeng Cai, Hai Wang, Xiaobo Chen, and Haobin Jiang. Trajectory-based anomalous behaviour detection for intelligent traffic surveillance. *IET intelligent transport systems*, 9(8):810–816, 2015.

[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[5] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[8] 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.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[10] Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-based surveillance systems*, pages 135–144. Springer, 2002.

[11] 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*, pages 21–37. Springer, 2016.

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

[13] Sadegh Mohammadi, Alessandro Perina, Hamed Kiani, and Vittorio Murino. Angry crowds: detecting violent events in videos. In *European Conference on Computer Vision*, pages 3–18. Springer, 2016.

[14] 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.

[15] Mahdyar Ravanbakhsh, Enver Sangineto, Moin Nabi, and Nicu Sebe. Training adversarial discriminators for cross-channel abnormal event detection in crowds. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1896–1904. IEEE, 2019.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[17] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[20] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, and Reinhard Klette. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. *IEEE Transactions on Image Processing*, 26(4):1992–2004, 2017.

[21] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

[22] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 246–252. IEEE, 1999.

[23] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2018.

[24] Waqas Sultani and Jin Young Choi. Abnormal traffic detection using intelligent driver model. In *2010 20th International Conference on Pattern Recognition*, pages 324–327. IEEE, 2010.

[25] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. *arXiv preprint arXiv:1903.09254*, 2019.

[26] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *CVPR Workshop (CVPRW) on the AI City Challenge*, 2018.

[27] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. *arXiv preprint arXiv:1811.07258*, 2018.

[28] JiaYi Wei, JianFei Zhao, YanYun Zhao, and ZhiCheng Zhao. Unsupervised anomaly detection for traffic surveillance based on background modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 129–136, 2018.

[29] Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117–127, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 145–152, 2018.

[31] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Haotian Wu, Qinqin Nie, Hao Cheng, Chenfeng Liu, et al. Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.