Unsupervised Anomaly Detection for Traffic Surveillance Based on Background Modeling

JiaYi Wei, JianFei Zhao, YanYun Zhao, ZhiCheng Zhao
Beijing University of Posts and Telecommunications
{weijiayi8888, zjfei, zyy, zhoazc}@bupt.edu.cn

Abstract

Most state-of-the-art anomaly detection methods are specific to detecting anomaly for pedestrians and cannot work without adequate normal training videos. Recently, there is a growing demand for detecting anomalous vehicles in traffic surveillance videos. However, the biggest challenge in this task is the lack of labeled datasets for training supervised models. By examining the resemblances of anomalous vehicles, we find it reasonable to label a vehicle as anomaly if it stays still in the video for a relatively long time. Utilizing this property, in this paper we introduce a novel unsupervised anomaly detection method for traffic surveilliance based on background modeling, which shows great potentials in handling heterogeneous scenes as well as extremely low resolution videos recordings without the dependence on labeled data. In the proposed system, we first employ background modeling using MOG2 to remove the moving vehicles as foreground while keeping the stopped vehicles as part of the background. Then we use Faster R-CNN to detect vehicles in the extracted background and decide if they are new anomalies under certain conditions. All information is updated on a frame basis until the end of the video which contains the final results. In this way, we make full use of the characteristics that abnormal vehicles stay in the scene for a relatively long time and reduce the difficulty of vehicle anomaly detection. Eventually, we can detect almost every anomaly in the NVIDIA AI CITY CHALLENGE track-2 dataset except for several extremely complex cases with a 81.08% *F1-score and 10.2369 RMSE.*

1. Introduction

Detecting anomalies in surveillance videos, *e.g*. stalled objects, accidents and abnormal objects, has been a challenging task because of the shortage of annotated or labeled data and the variable video scenes. Therefore, it is almost infeasible to acquire the orbits of every object in the videos and then judge whether the orbit should be classified as anomaly or not. However, it is a truly valuable task due to its potential application in real world.

The main trend in this area is to design or learn a feature representation for videos clips with no anomalies, such as[5][17][8][4][22]. Recently, thanks to the development of Generative Adversarial Network(GAN)[7], video prediction has been used on anomaly detection[14]. Despite of the great success of the previous works, we argue that almost all existing methods do not have the capacity to handle the real cases for two main reasons. First, most of them can just work on datasets ,*e.g*. UCSD[20] and CUHK Avenue[17], which are captured in homogeneous scenes, rather than in dataset with heterogeneous scenes, like Shanghai Tech [19]. Even if method like [14] experiments on [19], their performance is not satisfying enough. Second, the demand for anomaly detection in traffic surveillance is to be competent in all kinds of scenes, which is barely achieved by any existing methods. All these challenges exist in the NVIDIA AI CITY CHALLENGE track-2. This competition aims to promote the techniques that rely less on supervised approaches and can be applied in real life to make our transportation system safer. These are also motivation for our work.

To deal with the above challenges, we tackle the anomaly detection problem of traffic surveillance from a creative perspective. As we can tell from human eyes, vehicles running on the road have the similar patterns. Hence, we decide to design a system based on vehicle detection. But, it is extremely hard to detect every vehicle, especially with low resolution and severe congestion. After observing the videos with and without anomalies, we find that whenever an accident happens it leads to at least one stopped vehicle. It means that the backgrounds of videos before and after accidents are different, since the stopped vehicle becomes a part of the new background, as illustrated in Figure 1. Then, we can pick out a successively detected vehicle appearing in the background modeled video, which stays longer than the time duration when the traffic light turns red, as an anomaly.

In our system, we use the state-of-the-art detection model, Faster R-CNN[23], to detect vehicles cross frames.

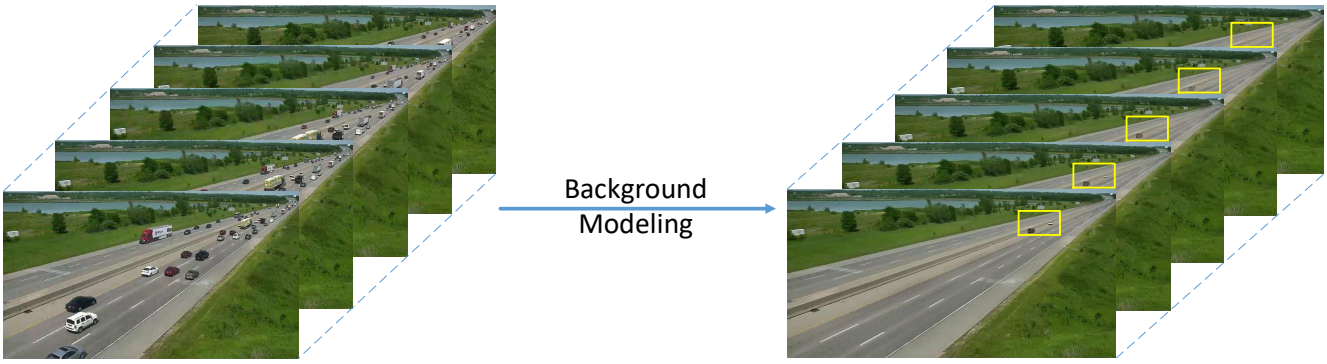

Figure 1. **Background Modeling.** The left frames are extracted from original evaluation video (2.mp4), while the frames on the right side are the background images generated from the left ones. Yellow rectangles indicate area containing stalled vehicles. We can easily notice that the stalled vehicles become parts of the background and the moving vehicles totally disappear in the background at the same time.

To find as many vehicles as possible in images, multi-scale detection is used in our experiment. Then, we implement a VGGNet[25] as a powerful classifier to reduce the false positive detection rate coming with multi-scale detection. All detected bounding boxes are delivered to the decision module to wait for the final judgment. Some cases can affect our decision module, *e.g*. the vehicles waiting for traffic light at the same position is one type of interference. The other type of interference is that the detected abnormal vehicle can show up in different scenes camera's perspective changes. Our strategy is to compare the similarity between detected vehicles. We employ the ResNet[9], trained with triplet loss, to accomplish similarity comparison. The performance of our method on track-2 illustrates its capacity that it can perform well in various scenes. To the best of our knowledge, it is the first work to use such pipeline for anomaly detection in traffic surveillance.

Our main contribution is our novel method, which combines the traditional background model and deep learning network. Through experiments, we show that the proposed method possesses several outstanding advantages over the other state-of-the-art anomaly detection methods: i) *Unsupervised:* Our method does not require any training data for specific scenes. ii) *Robust:* The system reduces the difficulty of anomaly detection and finishes the task in a simple and elegant way, with robustness against complex scenes. Due to the well designed decision logic of the final module, even some false or missed detection in the second module cannot affect the final judgment. Our method can detect almost all anomalies in the dataset provided by NVIDIA AI CITY CHALLENGE track-2 which includes abundant scenes. iii) *Generalizable:* Because of the nature of our method, it is able to detect accident in any scenes without special modification. Further details about our system will be present in the Section 3.

2. Related Work

Numerous efforts have been taken for anomaly detection [5][8][14]. However, because of the lack of datasets about the abnormal event in traffic surveillance, most of these work are made for anomaly detection in the crowd. Based on the strategies used, all the existing methods can be categorized into two categories: i) feature reconstruction for normal training data based methods [5][4][19][10]). ii) video frames prediction based methods[14].

2.1. Feature Reconstruction Based Methods

Generally speaking, the most works in computer vision solve the problem with a framework of reconstructing training data. They use either the hand-crafted features [5][17] or the features learned by a deep neural network with Auto-Encoder to reconstruct normal events with small reconstruction errors [4][8][22]. Sparse coding or dictionary learning is a popular approach to encode the normal event [5][17][27]. The Gaussian mixture model is used by Mahadevan *et al*. [20] to fit a mixture of dynamic textures (MDT). Also, in order to be more efficient both in training and testing phase, Lu et al [17] raises to discard the sparse constraints and learn multiple dictionaries to encode normal scale-invariant patches.

With the advances in deep learning area, lots of works [8][4][19][18] begin to utilize various architectures of Deep Neural Network (DNN) to detect anomaly. Hasan *et al*. proposes to use 3D convolutional auto-encoder (CAE) to model regular frames [8]. Furthermore, motivated by the strong capacity of Convolutional Neural Networks (CNN) in spatial feature learning, Recurrent Neural Network (RNN) and its long short term memory (LSTM) variant has been widely applied for sequential data modeling. Therefore, by taking advantage of CNN and RNN, [4][18] employ a Convolutional LSTMs Auto-Encoder (ConvLSTM-AE) to model the normal pattern, which hugely promotes the performance

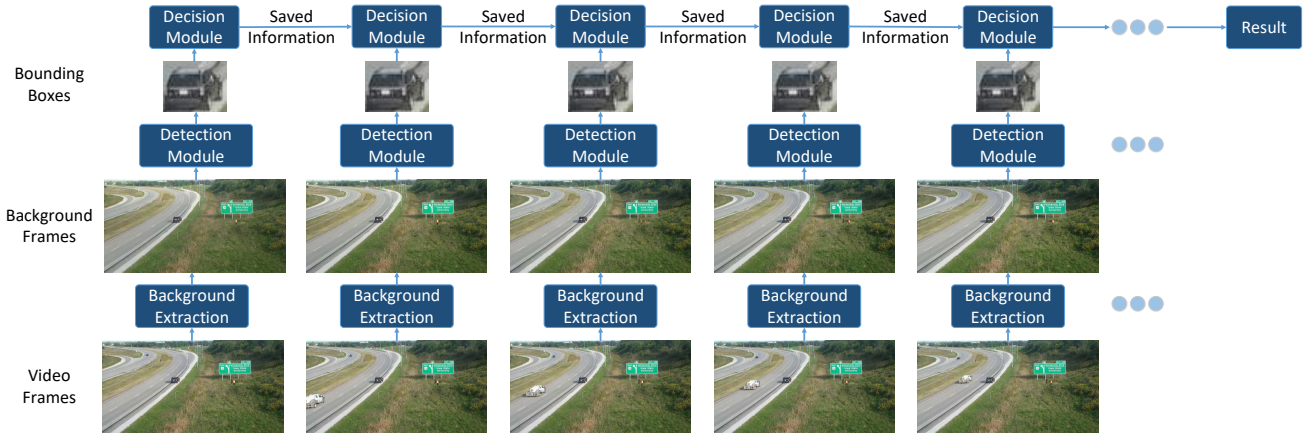

Figure 2. **Framework of our proposed system.** Every frame of a video is sent to background extraction module first. Then, the detection module detects vehicle in background images. Each detected vehicle from former module is passed to decision module. This module updates result in real time. When the video finished, our system gains the final prediction.

of methods based on CAE. In [19], Luo *et al*. proposes a method based on temporally coherent sparse coding which can map to a stacked RNN framework.

All above methods, aiming to detect anomalies, are based on the reconstruction of regular training data and the surmise that abnormal events could correspond to larger reconstruction errors. But features, whether hand-crafted or learned by deep learning model, cannot guarantee a large enough reconstruction error for abnormal incidents, which means that these methods are not really robust in complex scenes. Hence, the difference of reconstruction errors between normal and abnormal events can be small, leading to low capacity in discrimination.

2.2. Frame Prediction Based Method

Recently, prediction learning is attracting more and more attention due to its possible application in unsupervised feature learning for video representation [21]. [21] proposes a multi-scale network with adversarial training, which could generate more natural future frames in videos. Considering the identification of events that do not meet the expectation, [14] designs a frame prediction system based on U-Net [24] and compares the prediction with its ground truth for anomaly detection. However, methods based on frame prediction rely more on the dataset, which makes it almost infeasible to leverage such system in reality, because we cannot build separate models for each scene.

The shared shortage of all existing method is that they are not generalizable, which has been shown by their performance on the challenging dataset ShanghaiTech [19]. For [14], it can only achieve 72.8% AUC (Area Under Curve), while it can reach 83% AUC on the CUHK Avenue dataset [17] and about 90% AUC on the UCSD Pedestrian dataset [20]. It means that we need to train these models with data containing no anomaly when deploying them in a new circumstance. However, it is contrary to the nature of this challenge and impossible to achieve in real life. The purpose of unsupervised anomaly detection is to build a system which can be used in surveillance video without large modification. Unfortunately, none of these existing methods could achieve this goal.

3. Methods

Our traffic anomaly detection system, as shown in Figure 2, is composed of three modules. The first module extracts the background images of every frame (Figure 1), using MOG2. The detection module illustrated in Figure 3 is made up of the Faster R-CNN [23] detector and the classifier based on VGGNet [25]. Faster R-CNN is responsible for detecting as many vehicles as possible using multi-scale detection. Because the training data used to train the detector extremely differs from the data in track-2, we acquire plenty of false detections. Therefore, we utilize VGGNet as a classifier to eliminate the wrong detection results. Next, in order to determine if there is an accident based on the results obtained in the second module, we design a decision module. It can i) define a detected vehicle as anomaly according to the duration it stays in the background, ii)provide the timestamp when the anomaly happens, iii) eliminate the effect of scenes switching and traffic light using ResNet trained with triplet loss.

3.1. Background Extraction

For anomaly detection in traffic surveillance videos, some traditional methods attempt to track every vehicle and obtain its track. Then they analyze the track to determine whether the vehicle is abnormal or not. Unfortunately, light,

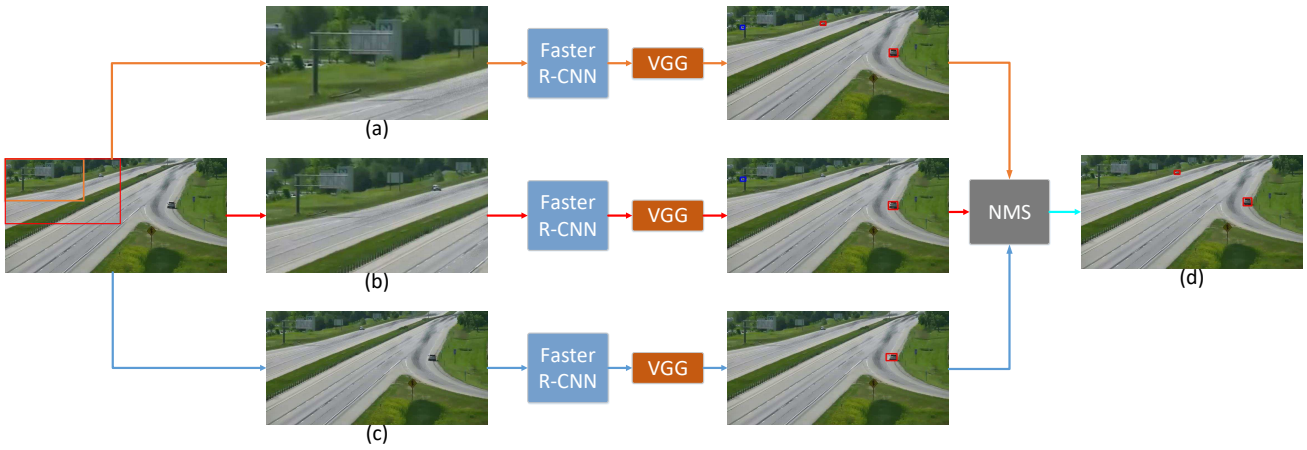

Figure 3. The pipeline of detection. The branch (a) represents the detection process on $3\times$ size image, and the same for branches (b) and (c). The red and blue bounding boxes indicate the positive and negative result, respectively. After NMS, we acquire the final result as (d).

weather and the quality of camera can all affect the quality of videos, which makes detecting and tracking vehicles accurately in various scenes almost impossible. In fact, once an abnormal event occurs, the relevant vehicles should stop and become part of the background. On the contrary, the non-relevant vehicles will not remain in the background since they keep moving. Thus, we exploit the background modeling to extract background images from video frames. If there is a stalled or crashed vehicle, it should appear in the background image. In this way, we can transform the complex problem into a simpler detection problem.

There are several algorithms for the purpose of background modeling. In our work, we utilize the MOG2, which is introduced by Andrew [28] and [29]. One important characteristic of this algorithm is that it selects the appropriate number of components of GMM for each pixel and provides better adaptability to varying scenes.

In traffic surveillance, severe congestion can cause numerous vehicles to stay in background, which severely affect the accuracy of anomaly detection. In order to resolve this problem, we need to choose a reasonable time period T for updating GMM parameters. In MOG2, the update rate is α, where $\alpha = 1/T$. A larger T can make MOG2 to adapt to the gradual change better. In our work, we set the parameter T to $120 frames$, which corresponds to $4s$ for test video (frame rate is $30 fps$). As a result, all normal moving vehicles are removed from the frames and all stopped vehicles stay in the background.

3.2. Detection Module

After removing moving vehicles from the frames, we now detect vehicles in the background images in this detection module. It contains a Faster R-CNN [23] detector and a VGGNet [25] classifier, as illustrated in Figure 3. In surveillance videos, it is hard to accurately detect the vehicles with a single detector since vehicles show up in different sizes. To deal with this issue, [13] and [3] try to detect objects at different scales to obtain a more accurate result. In this paper, we test every image in $1\times$, $2\times$ and $3\times$ size.

Due to the low resolution of the data provided by AI CITY CHALLENGE, the detector mistakenly identifies many non-vehicle objects as vehicles in our scaled images. To correct false detections, we leverage VGGNet to determine whether or not the candidates provided by Faster R-CNN are eligible. Then we use NMS to remove the overlapping bounding boxes coming from different branches of detection module. Our architecture is not as deep compared with *e.g.* ResNet [9] and DenseNet [11], but it has desirable ability to meet our needs.

3.3. Decision Module

In our decision module, we determine anomalous vehicles based on the bounding boxes produced by the former module. We also manage to eliminate the impact of traffic lights and the movements of the cameras. The procedure of determining abnormality is summarized in Algorithm 1.

It is worth mentioning that every saved vehicle information contains the following terms: $begintime$ (the time when the vehicle first appears), $endtime$ (the last time the vechile appears), $frequency$ (the total number of the vehicle's occurrences), $anomaly$ (whether the vehicle is abnormal) and $score$ (the confidence of this predicted anomaly).

As illustrated in Algorithm 1, for the bounding boxes in every frame produced by the detection module, we feed them to the decision module. When the current bounding box does not match any saved information, we save it as a new vehicle. On the other side, if the current detected bounding box shares the same position with a previ-

Algorithm 1 Decision module

Input:
The time t corresponding to the current frame;
The set of bounding boxes B detected in current frame;
The set of all the saved vehicle information I;
Output:
The set of updated vehicle information I_{new};

for each $b \in B$ **do**
  $i_{tmp} = \emptyset$
  for each $i \in I$ **do**
    if $(t - i[end] < 5s$ and $IoU(b, i) > 0.5)$ or $(t - i[end] > 15s$ and $Similar(b, i) < 0.9)$ **then**
      $i[end] = t$, $i[position] = b$, $i[f] += 1$;
      $i_{tmp} = i$;
      remove i from I
      break;
    end if
  end for
  if i_{tmp} is $\emptyset$ **then**
    $i_{tmp}[begin] = i_{tmp}[end] = t$,
    $i_{tmp}[position] = b$,
    $i_{tmp}[anomaly] = False$,
    $i_{tmp}[score] = 0$;
  end if
  if $i_{tmp}[anomaly]$ is $True$ **then**
    update $i_{tmp}[score]$;
  else if $i_{tmp}[end] - i_{tmp}[begin] > 30s$ and $i[f] > 25$ **then**
    $i_{tmp}[anomaly] = True$,
    update $i_{tmp}[score]$;
  end if
  $I_{new} \leftarrow i_{tmp}$
end for
$I_{new} \leftarrow I$

ously stored vehicle within $5s$, we treat the two as the same anomaly and update the saved vehicle with the current information: end time, position, frequency and the most recent time of occurence. We note that in cases where the perspective of a camera changes occasionally, all the previoulsy stored information will not be updated after the camera shifts and the same anomalies will be detect again as new anomalies. To match the anomalies before and after camera moves, we compare the similarities between the current bounding boxes and all saved vehicles if all stored information is not updated within 15s. For similarity comparison, we extract the features of a vehicle through ResNet50 trained with triplet loss and then compute the $L2$ distance to features of other vehicles. When the $L2$ distance is less than 0.9, we consider these two vehicles as same one.

If a vehicle has stayed in the background for more than $30s$ since its beginning time, the system labels it as a potential anomaly and starts to give it a score of being an anomaly which is updated afterwards in real time. We choose a $30s$ time window to avoid the interference when vehicles stop to wait at the red traffic light. The anomaly score of each piece of saved information is updated in real time. The equation of anomaly score is as follow:

$$score = frenquency/(end\_time - begin\_time) \quad (1)$$

Equation 1 indicates the proportion of frames containing the vehicle to the total number of frames from its appearing to disappearing. The nature of our method is that a vehicle staying in the background long enough should be detected as an anomaly. Thus, we employ the proportion of occurrences as the score of confidence. If the score is lower than 0.3, we think it cannot be an anomaly vehicle. And if the time span from beginning to end is more than $120s$, our system gives it a 100% confidence about anomaly. Because no matter if the vehicle is waiting for traffic light or stuck in traffic jam, it cannot stay in the same place more than 120 seconds.

In order to obtain a more accurate timestamp, we go back to track the abnormal vehicle A in the previous frames in the original video after we are certain about it being an anomaly, scoring 0.3 or higher. We set a $7W \times 7H$ size area centered at the anomaly vehicle, where $W \times H$ is the vehicles size. Then we rerun our detection module in this area to detect the vehicle A' from begintime conversely. When the IoU between previously detected A and redetected A' is more than 0.5, we consider them as the same vehicle and set the time when the abnormal vehicle A' disappears in the $7W \times 7H$ area in the original video as the timestamp of this anomaly.

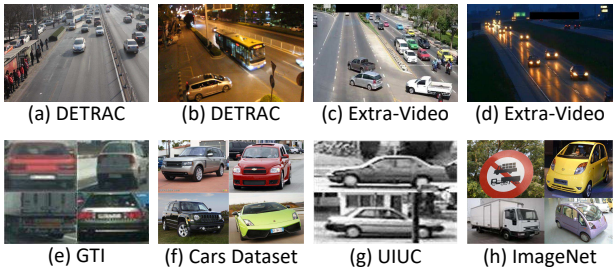

Figure 4. Some examples in the UA-DETRAC, Extra-Video, GTI, Cars Dataset, UIUC and ImageNet datasets.

3.4. Dataset

In this section we introduce the data used in our experiments. Samples are shown in Figure 4.

Data for detection We mainly use training sequences from UA-DETRAC [26] dataset to fine-tune our pre-trained Faster R-CNN model. There are about 84k frames and more than 578k annotated bounding boxes in this dataset. The

UA-DETRAC dataset contains video with large variations in scale, pose and illumination, occlusion, and background clutters, which makes it suitable for our work. Videos in UA-DETRAC have high resolution which differs greatly from the test datasets from NVIDIA AI CITY CHALLENGE track-2. To guarantee the performance on the test datasets, we add Gaussian blur on the training set (UA-DETRAC) to make up for the huge difference in the video resolution. The Gaussian filter on each dimension is as follows:

$$G_i = \alpha * e^{-(i-(K-1)/2)^2/(2*\sigma^2)} \tag{2}$$

where $i = 0, \ldots, K-1$ and α is the scale factor chosen such that $\sum_i G_i = 1$. The size of the filter is $K \times 1$. In our work, we set parameters as follows: $K = 5$, $\sigma X = 5$, $\sigma Y = 5$. Furthermore, we modify the lightness of images in training set as:

$$I_m[i,j][c] = a \times I_o[i,j][c] + b \tag{3}$$

where i, j donate the corresponding position of pixels, c indicates the channels, and $a = 0.7, b = 10$.

Due to the lack of night scenes and certain vehicle types (*e.g.* truck) in UA-DETRAC, we have collected 5 more video clips on the Internet for a total of 6 minutes and 10 seconds which we refer to as Extra-Video.. Then, we manually annotate 1012 frames with about 16k annotated bounding boxes, as shown in Figure 4. We modify the lightness of images in this dataset using Equation 3, where $a = 0.8, b = 40$.

Data for classification The data used to train our VGGNet comes from the following sources: i) ImageNet [6] (categories: car, truck, sign, road, snow and traffic light), ii) UIUC Car Detection [1], iii) GTI dataset [2], iv) Cars Dataset [12], v) Images of vehicles and non-vehicles randomly captured from the Extra-Video. For vehicle images, we crop the bounding boxes if provided and randomly crop 80% of the original size during training. For non-vehicle images, we make a crop of random size (10%, 30%, 50%, 100%) of the original size during training. All training images are resized to 64×64 and are rotated a certain degree randomly chosen from $[-10°, 10°]$.

Data for similarity We use the dataset ViRe [15][16] to train our ResNet50 [9] with triplet loss. This dataset contains over 50000 images of 776 vehicles captured by 20 cameras covering an area of $1.0km^2$ in 24 hours. This dataset aims to promote the research in vehicle Re-Id. It also meets our need to compare the similarity between vehicles.

4. Results

In this section, we evaluate the properties of different components in our proposed system and present the final results.

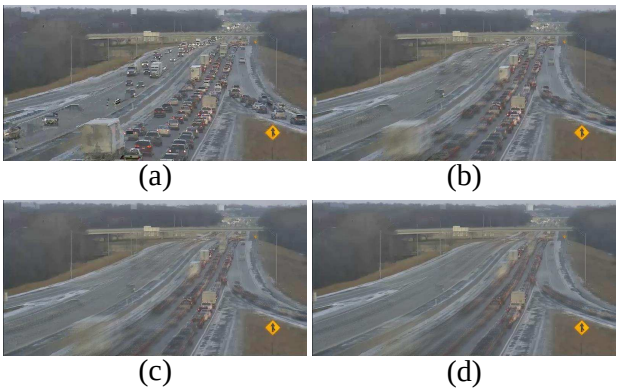

Figure 5. The effects of background extraction with different parameter T. All these images are about a certain frame of 14th video in test data. (a) is from the original video, and (b)(c)(d) are respectively from background extracted videos with $T = 30, 60, 120$.

MSD	VGG	F1(%)
×	×	48.65
×	√	56.25
√	×	32.26
√	√	81.08

Table 1. Test results about the capacity of detection module. "MSD" and "VGG" presents the multi-scale detection and the classifier. The √ and × means with or without corresponding section in the experiment. The performance is measured by F1.

Our expectation for background extraction module is not only to eliminate the effects of moving vehicles but also to reduce the impact of traffic jam. In real scenes, even if in most traffic jams, vehicles can still move at very low speed. Therefore, we want to set an appropriate T, which corresponds to the number of frames that can affect background modeling. As shown in Figure 5, we test the effects of background modeling with different values for T. With $T = 120$, we can eliminate the impact of severe congestion and provide good condition for the detection module.

In order to show the effectiveness of multi-scale detection and the classifier in the detection module (see Section 3.2), which we denote as MSD and VGG respectively, we examine the performance of detection module with different settings. From Table 1), we can see that our detection module work best with both MSD and VGG.

As shown in Table 1, our proposed method performs well in the challenging dataset containing various scenes. It can detect almost all anomalies in the dataset of track-2 and send back the precise positions of the detected anomalies, Figure 6. The results demonstrate the capabilities of our system to achieve robustness and generalizability against complex scenes without additional modification or training.

Figure 6. **Example Results on NVIDIA AI CITY CHALLENGE track-2.** Red bounding boxes show the accident vehicles detected by proposed method, which demonstrate the robustness and generality of this method.

In the NVIDIA AI CITY CHALLENGE track-2 traffic anomly detection competition, we detect a decent amount of anomalies with 0.8108 F1-score and 10.2369 RMSE. Among all the participated teams, our proposed method in this paper ranks in the 2nd place with S2-score 0.7853 on the official evaluation datasets.

5. Conclusion

Since normal moving vehicles cause great interference to anomaly detection for traffic surveillance, we propose a system which can reduce the effects of non-abnormality. We use MOG2 to extract background and eliminate the moving objects. In order to detect as many abnormal vehicles as possible, we utilize multi-scale detection and classification with the help of Faster R-CNN and VGGNet. With our decision module, we can greatly reduce the impact of traffic light waiting time, traffic jam and camera movement. Results on NVIDIA AI CITY CHALLENGE show the potential of our method to work on various scenes of traffic surveillance videos without special training on these scenes, which almost all existing methods can not achieve. Meanwhile, the main drawback our method is that it can provide only a rough rather than precise estimation of beginning time of an anomaly. To perfect our work, more focuses on the dynamic process of anomalies are needed in further work.

6. Acknowledgments

Thanks Xu Si for encouragement during the challenge and helpful advice on paper writing.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *European conference on computer vision*, pages 113–127. Springer, 2002. 6

[2] J. Arróspide, L. Salgado, and M. Nieto. Video analysis-based vehicle detection and tracking using an mcmc sampling framework. *EURASIP Journal on Advances in Signal Processing*, 2012(1):2, 2012. 6

[3] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016. 4

[4] Y. S. Chong and Y. H. Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *International Symposium on Neural Networks*, pages 189–196. Springer, 2017. 1, 2

[5] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456. IEEE, 2011. 1, 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 6

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[8] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 733–742. IEEE, 2016. 1, 2

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 4, 6

[10] R. Hinami, T. Mei, and S. Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. *arXiv preprint arXiv:1709.09121*, 2017. 2

[11] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017. 4

[12] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 6

[13] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan. Scale-aware fast r-cnn for pedestrian detection. *IEEE Transactions on Multimedia*, 2017. 4

[14] W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection–a new baseline. *arXiv preprint arXiv:1712.09867*, 2017. 1, 2, 3

[15] X. Liu, W. Liu, H. Ma, and H. Fu. Large-scale vehicle re-identification in urban surveillance videos. In *Multimedia and Expo (ICME), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016. 6

[16] X. Liu, W. Liu, T. Mei, and H. Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European Conference on Computer Vision*, pages 869–884. Springer, 2016. 6

[17] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2720–2727. IEEE, 2013. 1, 2, 3

[18] W. Luo, W. Liu, and S. Gao. Remembering history with convolutional lstm for anomaly detection. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, pages 439–444. IEEE, 2017. 2

[19] W. Luo, W. Liu, and S. Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. *ICCV, Oct*, 2017. 1, 2, 3

[20] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1975–1981. IEEE, 2010. 1, 2, 3

[21] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 3

[22] J. R. Medel and A. Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016. 1, 2

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 3, 4

[24] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3, 4

[26] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *arXiv preprint arXiv:1511.04136*, 2015. 5

[27] B. Zhao, L. Fei-Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3313–3320. IEEE, 2011. 2

[28] Z. 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. 4

[29] Z. Zivkovic and F. Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006. 4