Fast Unsupervised Anomaly Detection in Traffic Videos

Keval Doshi
University of South Florida
4202 E Fowler Ave, Tampa, FL 33620
kevaldoshi@mail.usf.edu

Yasin Yilmaz
University of South Florida
4202 E Fowler Ave, Tampa, FL 33620
yasiny@usf.edu

Abstract

Anomaly detection in traffic videos has been recently gaining attention due to its importance in intelligent transportation systems. Due to several factors such as weather, viewpoint, lighting conditions, etc. affecting the video quality of a real time traffic feed, it still remains a challenging problem. Even though the performance of state-of-the-art methods on the available benchmark dataset has been competitive, they demand a massive amount of external training data combined with significant computational resources. In this paper, we propose a fast unsupervised anomaly detection system comprising of three modules: preprocessing module, candidate selection module and backtracking anomaly detection module. The preprocessing module outputs stationary objects detected in a video. Then, the candidate selection module removes the misclassified stationary objects using a nearest neighbor approach and then uses K-means clustering to identify potential anomalous regions. Finally, the backtracking anomaly detection algorithm computes a similarity statistic and decides on the onset time of the anomaly. Experimental results on the Track 4 test set of the NVIDIA AI CITY 2020 challenge show the efficacy of the proposed framework as we achieve an F1-score of 0.5926 along with 8.2386 root mean square error (RMSE) and are ranked second in the competition.

1. Introduction

One of the most important, challenging and time-critical tasks in automated traffic video monitoring is the detection of abnormal events such as traffic accidents, violations and crimes. Hence, video anomaly detection has become an important research problem in recent years, especially because of its applications in intelligent transportation systems. Anomaly detection in general is a vast, crucial, and challenging research topic, which deals with the identification of data instances deviating from nominal patterns.

Given the important role that video anomaly detection can play in ensuring safety, security and sometimes prevention of potential catastrophes, one of the main outcomes of a video anomaly detection system is the real-time decision making capability. Events such as traffic accidents, robbery, and fire in remote places require immediate counteractions to be taken in a timely manner, which can be facilitated by the real-time detection of anomalous events. Despite its importance, a very limited body of research has focused on online and real-time detection methods. Regarding the importance of timely detection in video, as [11] argues, the methods should also be evaluated in terms of the average delay, in addition to the commonly used metrics such as true positive rate, false positive rate, and AUC.

A vast majority of the recent state-of-the-art video anomaly detection methods depend on complex neural network architectures [18]. Although deep neural networks provide superior performance on various machine learning and computer vision tasks, such as object detection [5], image classification [10], playing games [16], image synthesis[15], etc., where sufficiently large and inclusive data sets are available to train on, there is also a significant debate on their shortcomings in terms of interpretability, analyzability, and reliability of their decisions [8]. For example, [13, 17] propose using a nearest neighbor-based approach together with deep neural network structures to achieve robustness, interpretability for the decisions made by the model, and as defense against adversarial attack.

Motivated by the aforementioned domain challenges and research gaps, we propose a hybrid use of transfer learning based neural network and statistical clustering based approaches for unsupervised anomaly detection. In summary, our contributions in this paper are as follows:

- We propose a novel framework composed of a nearest neighbor and K-means clustering to detect anomalies without any training.
- We significantly reduce the testing computational overhead and completely remove the training overhead.
- We extensively test our algorithm on the benchmark

dataset without access to external data and yet perform comparatively well.

2. Related Works

Semi-supervised detection of anomalies in videos, also known as outlier detection, is a commonly adopted learning technique due to the inherent limitations in availability of annotated and anomalous instances. This category of learning methods deals with learning a notion of normality from nominal training videos, and attempts to detect deviations from the learned normality notion. [4, 7]. There are also several supervised detection methods, which train on both nominal and anomalous videos. The main drawback of such methods is the difficulty in finding frame-level labeled, representative, and inclusive anomaly instances. To this end, [18] proposes using a deep multiple instance learning (MIL) approach to train on video-level annotated videos, in a weakly supervised manner. Although training on anomalous videos would enhance the detection capability on similar anomaly events, supervised methods typically suffer from unknown and novel anomaly types.

One of the key components of the video anomaly detection algorithms is the extraction of meaningful features, which can capture the difference between the nominal and anomalous events within the video. The selection of feature types has a significant impact on the identifiability of types of anomalous events in the video sequences. Many early video anomaly detection techniques and some recent ones focused on the trajectory features [1], which limits their applicability to the detection of the anomalies related to the trajectory patterns, and moving objects. For instance, [6] studied detection of abnormal vehicle trajectories such as illegal U-turn. [12] extracts human skeleton trajectory patterns, and hence is limited to only the detection of abnormalities in human behavior.

Particularly, in previous NVIDIA AI CITY Challenges [21, 9, 3, 2] use the background modeling method to effectively eliminate the interference of the mobile vehicle, and obtains the location of the static region to analyze, which has achieved competitive results. However, all the above mentioned algorithms required significant amount of external training data and computationally expensive detection models.

3. Proposed Method

The purpose of this challenge is to design a practical framework which is capable of detecting anomalies in traffic videos. While existing works have shown remarkable results on the benchmark dataset, they have a high computation overhead. For example, the state of the art algorithm proposed in [2] requires training the detection model on external datasets [21, 23] and uses the computationally heavy ResNet-50 model which requires 311 ms per frame for object detection on a reasonable GPU. In [22, 19], a vehicle tracking approach is proposed. However, given the high number of vehicles detected in a traffic video, tracking each vehicle also becomes computationally inefficient.

While such algorithms achieve superior performance, they are difficult to implement in a practical setup. Thus, we propose a more heuristic approach based on how humans detect an anomaly in the video. First, we propose to focus only on the stationary objects that we see in the video, specifically cars and trucks. Then, using a nearest neighbor approach, we remove the missclassified vehicles by removing those objects which occur only a few times or occur throughout the video. Then, using clustering we detect regions where a potential anomaly might have occurred. Finally, in the anomaly detection stage, given the region of interest, we locate the first instance where an anomalous vehicle is detected using a backtracking algorithm.

In the following subsections, we describe in detail the proposed three-stage method for fast anomaly detection. We begin by discussing the preprocessing stage, composed of background modelling, road segmentation and object detection. Then, we explain the second stage, the potential candidate selection and localization technique. Finally, the anomaly detection algorithm, which enables timely and accurately detection of the onset of anomalies, is the third stage in the proposed framework. The entire algorithm is given in Algorithm 1.

3.1. Preprocessing

Background Modelling: While most recent works[22, 19, 9] propose using some kind of object detection for vehicle tracking, in this paper we focus on the anomalies related to stationary objects. Hence, similar to [21, 2], we use an averaging technique to emphasize the stationary objects in the video and suppress the moving vehicles. For a given video V with N frames $F^1, \ldots, F^N$, we continuously compute the weighted sum given by:

$$F^t_{avg} = (1-\alpha)F^t_{avg} + \alpha F^{t+m} \tag{1}$$

where F^t_{avg} is the averaged image at time $t = 100, \ldots, N$, α is the update rate and m is a fixed interval. To reduce the complexity for averaging, we use a sampling period of 100, i.e., only consider 1 frame every 100 frames. In this work, we set $\alpha = 0.1$ and $m = 30$.

Road Segmentation: Given that the primary objective in this work is detecting stationary vehicles on the highway, any stationary vehicle detected in a parking lot in the background might cause false positives and need to be effectively ruled out. One way to do this is by extracting the segmentation maps of the roads by using an unsupervised approach. Once we detect moving vehicles in a video, we continuously update the frequency map for the image.

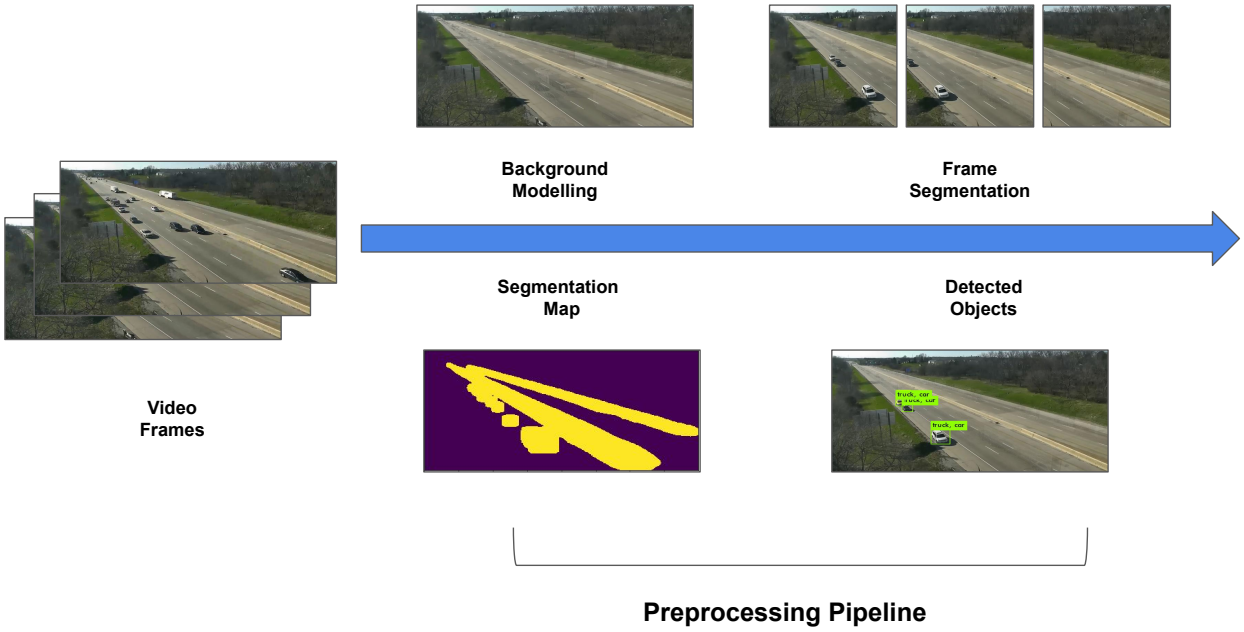

Figure 1. The preprocessing stage of the proposed method.

Then, similar to [2] we normalize the image and perform binarization to extract the segmentation map (S) as shown in 1.

Object Detection: While most existing algorithms also employ some form of object detection, they train specialized models on huge datasets. While such frameworks perform well on the benchmark dataset, we believe that it is a very specific approach and would not be practical because of the immense computation cost associated with retraining such models on new data and the time required to make a decision per observed frame. To this end, we propose leveraging transfer learning to detect objects using a real-time object detection system such as You Only Look Once (YOLO) [14] pretrained on the MS-COCO dataset to obtain the location of potential anomalies.

The advantage of YOLO is that it is capable of processing higher frames per second on a GPU while providing the same or even better accuracy as compared to the other state-of-the-art models such as SSD and ResNet. Speed is a critical factor for detecting anomalies in traffic videos, so we currently prefer YOLOv3 in our implementations.

For each detected object in image F^t_{avg}, we get a bounding box (location) along with the class probabilities (appearance). We remove overlapping boxes by using non-maximum suppression (NMS) if the Intersection of Union is greater than 0.3. Then, for each video V, we build a set C_{XY} consisting of the center (c^t_{xi}, c^t_{yi}) of each object i detected at time t and a set L_{XY} consisting of the corresponding width and height (w^t_i, h^t_i). In this work, we only consider a few classes corresponding to vehicles such as cars, buses, trucks, etc. and ignore the rest of the bounding boxes. Due to the perspective geometry between the vehicles and the camera, most of the vehicles at a distance are small and hard to detect even for humans. Hence, we set a low threshold h of 0.1 for YOLO.

3.2. Candidate Selection

While setting a low threshold of confidence for object detection helps in detecting very small objects, it also leads to misclassification of objects in the background like traffic signals and road signs. Also, while the background model suppresses most of the moving vehicles, some slow moving objects are not completely removed and get detected by the detection algorithm. By observing a single frame, it is very difficult if not impossible to remove such misclassifications without leveraging external datasets. However, by including the temporal information, i.e., by combining all the bounding boxes detected in the video, we see some patterns emerge. The objects in the background which are misclassified tend to occur frequently at the same location from the beginning of the video till the very end and form a high density cluster. Conversely, the slow moving objects do not occur frequently at the same location and thus look like outliers. Hence, we implement a nearest neighbor

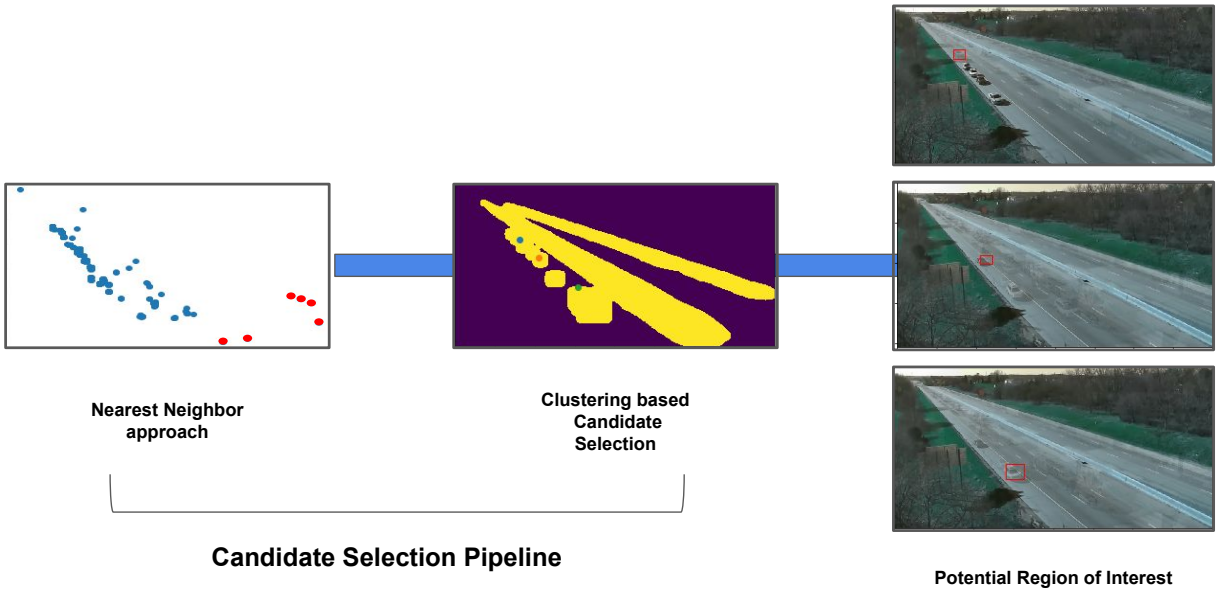

Figure 2. The candidate selection stage of the proposed method

based approach to rule out such cases.

Once we perform object detection for the entire video, we map the center (c^t_{xi}, c^t_{yi}) of the bounding box for an object i detected at each time instance t to a two dimensional plane. Then for each point (c^t_{xi}, c^t_{yi}), we compute the k-Nearest-Neighbor (kNN) distance $d^t_{xi,yi}(k)$ to its neighboring points. Specifically, we consider a point (c^t_{xi}, c^t_{yi}) as misclassified if

$$d^t_{xi,yi}(k_1) \leq l_1 \tag{2}$$

where $k_1 \gg l_1$ and as a slow moving vehicle if

$$d^t_{xi,yi}(k_2) \geq l_2 \tag{3}$$

such that $k_2 \ll l_2$.

As shown in Fig. 2, the nearest neighbor approach is efficiently able to remove the outliers. Following the nearest neighbor implementation, we employ a K-means clustering algorithm with a segmentation map overlay to localize potential "hot spots", i.e., locate regions where stationary objects were detected. This step provides us with K centroids $(m_1, n_1), \ldots, (m_K, n_K)$ or potential spatial locations in the video where an anomaly might have occurred. Here, K is chosen by first computing the within-cluster sum of squares for a range of values and then using the elbow method to find the optimal value of K. Finally, we iteratively look for the first time instance $t_{K\alpha}$ where we detect an object at each of the K locations (centroids) or potential regions of interest. Since we only detect objects every 100 frames and due to the delay caused by a small α in (1), we use a backtracking algorithm which monitors a similarity score to find the true onset time of anomaly. In the next subsection, we discuss our backtracking algorithm in detail.

3.3. Backtracking Anomaly Detection:

Given the potential anomaly onset time $t_{K\alpha}$ for K centroids and region of interest (w^t_i, h^t_i) extracted from the Set L_{XY}, we compute the structural similarity (SSim) [20], between the region of interest at time $t_{K\alpha}$ and each instance t between the start of the video, i.e. $t = 0$ and $t_{K\alpha}$. Ideally, when there is no stalled vehicle at the location, the structural similarity is very low and almost close to zero. As soon as a stalled vehicle appears in the frame, there is a dramatic increase in the structural similarity, which we detect by setting a threshold on the similarity statistic. To remove increases caused due to noise, we apply a Savitzky-Golay filter to the similarity statistic. Specifically, we focus on whether the increase is *persistant* over several frames or occurs only over a couple of frames. Finally, we declare t as the onset time of the anomaly (δ_t) when the similarity statistic crosses the threshold. The efficacy of our algorithm is shown in Fig. 3, where we are satisfactorily able to detect a stopped car with minimum detection delay. Also, it is interesting to note that in our entire pipeline, YOLO requires the maximum computation time which on an average is 19 ms. In Algorithm 1, we summarize our entire pipeline.

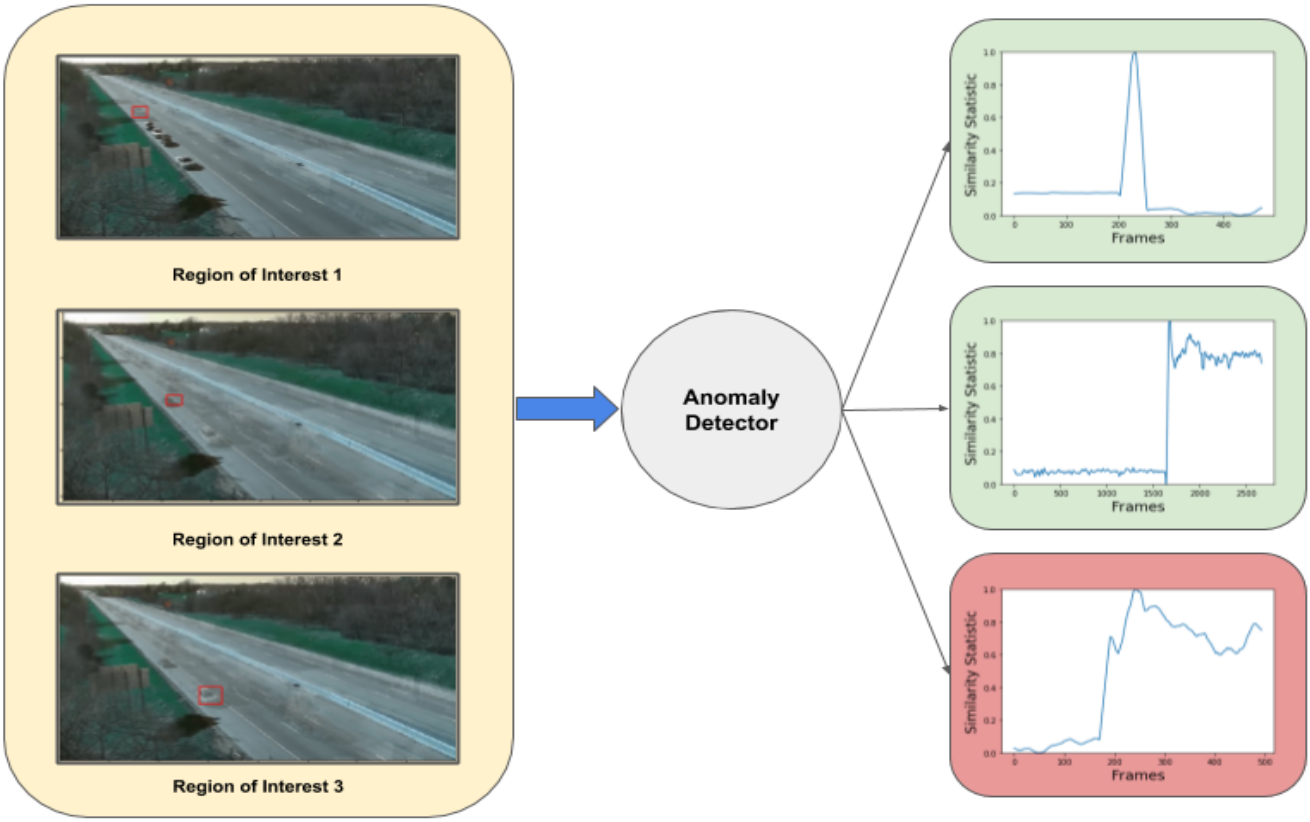

Figure 3. Backtracking Anomaly Detection pipeline for the proposed framework. We monitor the structural similarity for each region of interest and decide to raise an alarm when the first time it crosses a threshold.

4. Experiments

In this section, we first introduce the benchmark dataset we use for evaluating our framework. Then, we present the performance of our method on Track 4 of the AI CITY Challenge.

4.1. Track 4 dataset

The Track 4 training and testing set of the NVIDIA AI CITY Challenge 2020 consists of 100 videos each with a mean video length of about 15 minutes, an excellent frame rate of 30 frames per second and a decent resolution of 800 x 410. The anomalous behaviors mainly constitute of vehicles driving off the road, stalled vehicles and crashes. However, due to many factors such as a range of weather conditions, lighting conditions, viewing angles etc, each video presents a unique challenge. The main objective of the task is to detect the anomalies in videos with a low detection delay and a high F_1 score. As compared to previous year challenges, this task is considerably more difficult as no external dataset was allowed.

The evaluation for track 4 had two major criteria, namely detection delay measured by the root mean square error (RMSE) and the detection performance measured by the F_1 score. Specifically, the final statistic was termed as S_4 and was computed as

$$S_4 = F_1(1 - NRMSE) \tag{4}$$

where NRMSE is the normalized version of the root mean square error. The range of scores was from 0 to 1 with 1 signifying the best performance that could be achieved. A detection was considered as a true positive if it was detected within 10 seconds of the true anomaly. The maximum RMSE that could be achieved was 300 which led to a S_4 score of 0.

4.2. Performance Evaluation

As shown in Table 1, we achieve a F_1 score of 0.5926 and a RMSE score of 8.2386. The final S_4 score computed using 4 was 0.5763, which placed us second in the challenge. Considering that no external data was used and the framework had zero training computational overhead, it shows the generalizing capability of our proposed algorithm. In Table 2, we show the results among all teams.

Algorithm 1: Proposed anomaly detection algorithm

Stage: Preprocessing
Input: $F^1, F^{100}, \ldots, F^N$
Output: $(c_{xi}^1, c_{yi}^1), (c_{xi}^{100}, c_{yi}^{100}), \ldots, (c_{xi}^N, c_{yi}^N)$

for $t = 1, 100, \ldots, N$ **do**
  Obtain the averaged image F_{avg}^t using (1).
  Determine bounding box for each detected object i.
  Remove overlapping boxes using NMS.
  Build set C_{XY} and L_{XY}.
  Compute segmentation map (S).
end for

Stage: Candidate Selection
Input: Set C_{XY}, Set L_{XY} and Segmentation Map S
Output: Centroid $(m_1, n_1), \ldots, (m_K, n_K)$

for $t = 1, 100, \ldots, N$ **do**
  Remove misclassified objects using (2).
  Remove slow moving vehicles using (3).
end for
for $K = 1, \ldots, 15$ **do**
  Compute within-cluster sum of squares.
end for
Select K using elbow method.
if Centroid (m_k, n_k) not in S **then**
  Remove (m_k, n_k)
end if
for $t = 1, 100, \ldots, N$ **do**
  for $k = 1, \ldots, K$ **do**
    if $c_{xi}^t - 5 \leq m_k \leq c_{xi}^t + 5$ **then**
      if $c_{yi}^t - 5 \leq m_k \leq c_{yi}^t + 5$ **then**
        Declare t as potential anomaly onset time $t_{K\alpha}$ for centroid k.
        Declare (w_i^t, h_i^t) as potential region of interest for centroid k.
      end if
    end if
  end for
end for

Stage: Backtracking Anomaly detection
Input: Potential anomaly onset time $t_{k\alpha}$ for centroid k, region of interest (w_i^t, h_i^t) and Set L_{XY}.
Output: True Anomaly onset time δ_t

for $t = 1, 10, \ldots, t$ **do**
  for $k = 1, \ldots, K$ **do**
    if SSim (ROI^t,$ROI^{t_{k\alpha}}$) > threshold) **then**
      Declare true anomaly onset time δ_t.
    end if
  end for
end for

	F1	RMSE	S4
Our Method	0.5926	8.2386	0.5763

Table 1. Our performance

1	113	Firefly	0.9695
2	51	SIS Lab	0.5763
3	106	CETCVLAB	0.5438
4	72	UMD_RC	0.2952
5	91	HappyLoner	0.2909
6	26	Orange-Control	0.2386
7	49	PapaNet	0.1703
8	132	Team_Gaze_NSU_UAP	0.0958

Table 2. Result comparison on the Track 4 test set from the top 8 on the leaderboard.

4.3. Conclusion

In this work, we attempted to tackle the anomaly detection in traffic video challenge. For video anomaly detection, we presented a fast algorithm which consists of a deep learning-based object detection module and two statistical decision making module. We show that as compared to the other state-of-the-art methods, we have negligible training computational overhead and are able to generalize well to different scenarios without access to any external data. Our future work would include density estimation for the K-means algorithm and a continual learning based model capable of learning different type of anomalies.

References

[1] Nadeem Anjum and Andrea Cavallaro. Multifeature object trajectory clustering for video analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1555–1564, 2008. 2

[2] Shuai Bai, Zhiqun He, Yu Lei, Wei Wu, Chengkai Zhu, Ming Sun, and Junjie Yan. Traffic anomaly detection via perspective map based on spatial-temporal information matrix. In *Proc. CVPR Workshops*, 2019. 2, 3

[3] Kuldeep Marotirao Biradar, Ayushi Gupta, Murari Mandal, and Santosh Kumar Vipparthi. Challenges in time-stamp aware anomaly detection in traffic videos. *arXiv preprint arXiv:1906.04574*, 2019. 2

[4] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. Video anomaly detection and localization using hierarchical feature representation and gaussian process regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2909–2917, 2015. 2

[5] 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. 1

[6] Zhouyu Fu, Weiming Hu, and Tieniu Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *IEEE International Conference on Image Processing 2005*, volume 2, pages II–602. IEEE, 2005. 2

[7] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7842–7851, 2019. 2

[8] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in neural information processing systems*, pages 5541–5552, 2018. 1

[9] Pirazh Khorramshahi, Neehar Peri, Amit Kumar, Anshul Shah, and Rama Chellappa. Attention driven vehicle re-identification and unsupervised anomaly detection for traffic understanding. In *Proc. CVPR Workshops*, pages 239–246, 2019. 2

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[11] Huizi Mao, Xiaodong Yang, and William J Dally. A delay metric for video object detection: What average precision fails to tell. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 573–582, 2019. 1

[12] Romero Morais, Vuong Le, Truyen Tran, Budhaditya Saha, Moussa Mansour, and Svetha Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11996–12004, 2019. 2

[13] Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018. 1

[14] 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. 3

[15] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 1

[16] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017. 1

[17] Chawin Sitawarin and David Wagner. Defending against adversarial examples with k-nearest neighbor. *arXiv preprint arXiv:1906.09525*, 2019. 1

[18] 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. 1, 2

[19] Gaoang Wang, Xinyu Yuan, Aotian Zhang, Hung-Min Hsu, and Jenq-Neng Hwang. Anomaly candidate identification and starting time estimation of vehicles from traffic videos. In *AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California*, 2019. 2

[20] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4

[21] 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. 2

[22] Jianfei Zhao, Zitong Yi, Siyang Pan, Yanyun Zhao, and Bojin Zhuang. Unsupervised traffic anomaly detection using trajectories. In *Proc. CVPR Workshops*, 2019. 2

[23] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018. 2