Multi-Granularity Tracking with Modularlized Components for Unsupervised Vehicles Anomaly Detection

Yingying Li[1], Jie Wu[2]*, Xue Bai[1]*, Xipeng Yang[1], Xiao Tan[1], Guanbin Li[2],
Shilei Wen[1], Hongwu Zhang[1], Errui Ding[1]
[1] Department of Computer Vision Technology (VIS), Baidu Inc. [2] Sun Yat-sen University.
{liyingying05, baixue06, yangxipeng01, tanxiao01, wenshilei, zhanghongwu, dingerrui}@baidu.com
wujie23@mail2.sysu.edu.cn, liguanbin@mail.sysu.edu.cn

Abstract

Anomaly detection on road traffic is a fundamental computer vision task and plays a critical role in video structure analysis and urban traffic analysis. Although it has attracted intense attention in recent years, it remains a very challenging problem due to the complexity of the traffic scene, the dense chaos of traffic flow and the lack of fine-grained abnormal labeled data. In this paper, we propose a multi-granularity tracking approach with modularized components to analyze traffic anomaly detection. The modularized framework consists of a detection module, a background modeling module, a mask extraction module, and a multi-granularity tracking algorithm. Concretely, a box-level tracking branch and a pixel-level tracking branch is employed respectively to make abnormal predictions. Each tracking branch helps to capture abnormal abstractions at different granularity levels and provide rich and complementary information for the concept learning of abnormal behaviors. Finally, a novel fusion and backtracking optimization is further performed to refine the abnormal predictions. The experimental results reveal that our framework is superior in the Track4 test set of the NVIDIA AI CITY 2020 CHALLENGE, which ranked first in this competition, with a 98.5% F1-score and 4.8737 root mean square error.

1. Introduction

Anomaly detection of traffic accidents plays a critical role in urban traffic analysis and potential down-stream applications like evidence investigation. With the rapid development of computer vision in recent years, anomaly detection in road traffic has attracted more attention, an effective and automated anomaly detection method can promote effective and efficient traffic management. As shown in Figure1, due to the complexity of traffic conditions, the diversity of weather and the size of vehicles, there are great challenges in the detection of traffic anomalies.

Figure 1. Overview of anomaly detection of traffic accidents. Complex road scenes make the task very challenging. The red boxes denote the abnormal locations in the picture.

In recent years, deep learning based anomaly detection methods have been developed rapidly, but it remains a very challenging problem due to the serious imbalance between normal and abnormal samples, the serious lack of fine-grained labeling data about abnormal events and the ambiguity about the concept of abnormal behaviors. In contrast, normal data is easier to obtain, previous studies [16, 2, 11, 15, 5, 3] generally leverage normal training samples to model abnormal concepts, and identify the distinctive behaviors that deviate from normal patterns as anomalies. However, these works are not accessible to abnormal videos, which may incorrectly classify some normal behaviors with abrupt action as abnormal ones. On the other hand, Anomaly detection of traffic accidents needs to be competent in all traffic scenarios, these deep learning-based methods can just work on homogeneous scenes datasets, most of them perform poorly when faced with unknown road traffic scenes and complex traffic conditions.

To deal with the above challenges, we tackle the traffic anomaly detection problem based on vehicle detection and tracking. By analyzing various traffic accident videos,

*Jie Wu and Xue Bai contributed equally to this work.

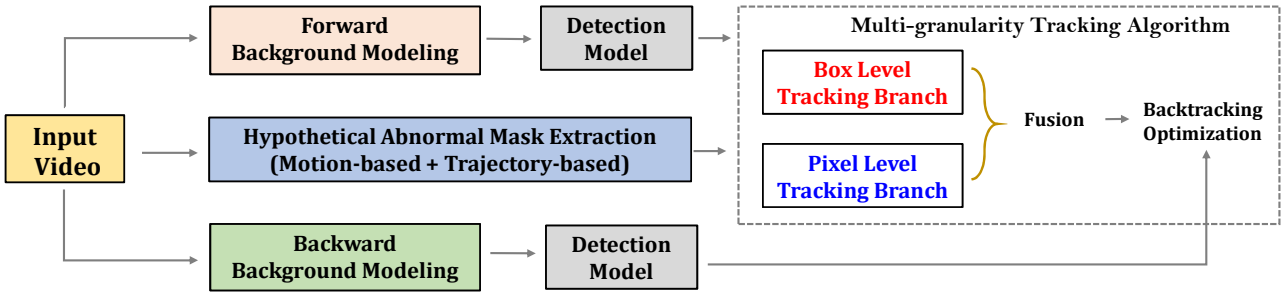

Figure 2. The illustration of multi-granularity tracking with modularlized components framework. This framework involves fusion from box-level tracking branch and pixel-level tracking branch. The backtracking optimization is performed to further improve the predictions.

we conclude that when a traffic accident occurs, the relevant vehicles will usually stop abruptly and last for a period of time. So we assume that if a stopped vehicle stays longer than the traffic light signal period, it can be regarded as an abnormal event. In this paper, we propose a multi-granularity tracking approach with modularized components to address the traffic anomaly detection task. Concretely, the novel multi-granularity tracking mechanism involves a box-level tracking branch and an improved pixel-level tracking branch inspired by [1]. Each branch manages to model abnormal concepts at different granularity levels and compensates for each other to make a robust prediction. The box-level branch links the detected boxes and constructs the tube to enclose the trajectory of the anomaly. The pixel-level branch introduces a similarity backtrack algorithm to accurately locate the start time of the anomaly. In modularized components, a vehicle detection model is exploited to detect vehicles in the video frames. Then we develop background modeling based on the Gaussian Mixture Model (GMM) to eliminate moving vehicles, so that stationary vehicles are easier to detect. To eliminate the interference outside the main road, such as parking lots and side roads that allow parking, we design a segmentation method based on frame changes and vehicle tracking results. Moreover, we introduce an anomaly fusion and backtracking optimization method to further boost the performance of anomaly predictions. The main contributions are summarized as follows:

- We present a multi-granularity tracking framework which contains a box-level tracking branch and a pixel-level tracking branch. Each branch contributes to capturing abnormal abstractions at different granularity levels for abnormal concepts modeling.
- We propose a novel mask extraction mechanism based on frame difference and vehicle tracking trajectory. It manages to effectively generate hypothetical anomaly regional masks at a lower false-positive rate.
- We propose an anomaly fusion and backtracking optimization method to refine the abnormal predictions, which can significantly improve the robustness and accuracy of the results.

Based on the above technical points, we evaluated our method on the Track 4 test set of the NVIDIA AI CITY 2020 CHALLENGE, We ranked first among the 8 participating teams, and we obtain the F1-score metric at 0.9855 and the RMSE metric at 4.8737. The source codes have been released at *https://github.com/WuJie1010/AICity2020-Anomaly-Detection*.

2. Related Work

As a most challenging task in the computer vision field, anomaly detection has been extensively studied for a long time [8, 17, 9, 6, 16, 2, 5, 27, 23, 32, 30, 33]. Most works employ normal videos to model abnormal concepts and treat the behaviors that deviate from the normal abstraction as anomalous. These researches have been conducted to leverage a series of statistic patterns, e.g., Hidden Markov Model [9, 6], Markov Random Field [8, 17], and sparse reconstruction [16, 2, 31, 15] to learn anomaly. With the development of deep learning technology, [5, 27] resort to the autoencoders with reconstruction loss to address the anomaly forecasting task. Sultani *et al.* [23] first propose weakly supervised anomaly detection that merely resorts to video-level labels (indicates whether the video is abnormal) to model abnormal concepts. They also attempt to optimize the detection model via both normal and abnormal videos.

Sultani *et al.* [23] collect the UCF-Crime [23] dataset, which is the largest anomaly detection datasets containing anomaly videos of diverse categories in complicated surveillance scenarios. Zhong *et al.* [32] formulate this weakly-supervised task as a supervised learning task under noise and employ a graph convolutional network to correct the noise labels. However, these works [23, 32, 30, 33] fail to take into account two core issues. First, they use the abnormal data with label information to model abnormal abstraction, which requires labor-intensive manual annotations. On the other hand, they fail to account for the more practical and meaningful anomaly such as vehicle anomaly detection.

Vehicle anomaly detection is more fine-grained anomaly detection, which is specially used to detect anomalies such as lane violations, wrong-direction driving, etc. In NVIDIA AI CITY CHALLENGE 2018[18] and NVIDIA AI CITY CHALLENGE 2019[19], unsupervised vehicle anomaly detection for road scenes have attracted considerable interests, which contributes to fine-grained anomaly detection in actual traffic accident scenarios and promoting the development of intelligent transportation. [28, 25] design the background modeling method to analyze the potential stationary vehicles. [20] proposes to use multiple adaptive vehicle detectors for abnormal proposals and adopt heuristics properties extracted from proposals to determine anomaly events. [1] presents a novel spatial-temporal information matrix, which transforms the analysis of a strip trajectory into an analysis of the spatial position. [1] ranked first among the 23 participating teams, and won the championship of anomaly detection track in NVIDIA AI CITY CHALLENGE 2019[19].

In this paper, we propose a multi-granularity tracking approach for unsupervised vehicle anomaly detection. We employ a box-level tracking branch and a pixel-level tracking branch to model abnormal concepts at different granularity levels, which jointly facilitate framework learning and improve the final performance. our proposed method achieves 0.9695 S4 score and ranks the first place among all the participant teams in the NVIDIA AI CITY CHALLENGE 2020.

3. Methodology

Figure 2 illustrates the proposed framework and its modular components. In the following sections, we first illustrate our detection model in section 3.1. Then we describe the background modeling and the extraction of the hypothetical abnormal masks in section 3.2 and 3.3, respectively. Section 3.4 introduces the proposed multi-granularity tracking approach, which employs a box-level tracking branch and a pixel-level tracking branch to model abnormal concepts at different granularity levels. Finally, how to obtain the fusion results and the backtracking optimization process are described in Section 3.4.3.

3.1. Detection Model

Recently, object detection becomes popular in both single-stage and two-stage detectors [14, 10]. In the single-stage pipelines, the locations of the target objects are generated directly from the feature map of the end of CNN. In two-stage pipelines, e.g. Fast R-CNN[4] and Faster-RCNN [22], the final predictions are obtained from features that generated in a specific region of interests, and the final predicted boxes are refined by CNN. Although single-stage detectors are efficient, current state-of-the-art object detectors usually adopt two-stage approaches for higher accuracy. In this task, we use a Faster R-CNN [22] to build our detection framework, which adopts SENet [7] with the depth of 152 as the backbone feature extractor. FPN [12] is worked on the backbone to increase semantic features information at each level in the extracted features. To fit into the track4 detection task, we clustering anchors on the track4 training dataset. Specifically, we used k-means clustering algorithm , and the distance metric is defined as:

Figure 3. Examples of detection results, which from video 13, 18, 23 and 28 in track4 test dataset. From the visualization, we observe that small targets can be predicted accurately.

$$D(box, centroid) = 1 - IoU(box, centroid). \quad (1)$$

The larger resolution, data flipping and data cropping are also exploited as data augmentation for facilitating training: 1) large resolution input is used to further boost the detection recall, especially for small targets. 2) Data flipping is used to ease the problem of false-positive caused by special scenarios. The data flipping method adopts a random mirror flip of the images, and the random probability is 0.5. For instance, the vehicles do not appear in specific areas of images in the training set. And the data flipping method can make up for this and provide more robust information for detection. 3)We observe that many vehicles only occupy pixel-level size in the image. and the vehicle size on the top area of the image is smaller than the bottom area of images due to the 3D perspective. Hence we adopt the random cropping method to learn multi-scale concepts. Concretely, randomly crop is employed to the whole images and then resize the cropped image to 1333×800.

The model is pre-trained on COCO [13], the detection training dataset is from AICity2020 track4 training videos [24]. The final model is trained on PaddlePaddle framework [1]. We extract one frame every four seconds from the training set video and assign bounding box level labels to the vehicles in images. Some visualizations of the detection predictions are shown in Figure 3.

[1] https://github.com/PaddlePaddle/PaddleDetection

Figure 4. Examples of background modeling. From the left to the right: original frame, background modeling by Moving Average, background modeling by MOG2 ordinally.

3.2. Background Modeling

As abnormal traffic events generally bring about stopped vehicles, detecting static vehicles is regarded as a robust and effective way in anomaly detection.

To obtain the stationary parts meanwhile fade the moving vehicles into the background, we try several algorithms for background modeling. As shown in Figure 4, MOG2[34, 35] is more stable than Moving Average [29]. Thus we adopt MOG2 in this paper, which selects the appropriate number of components of GMM for each pixel and provides better adaptability to varying scenes. In MOG2, T denotes the time period for updating GMM parameters and a larger T adapts better to gradual changes. In our work, the update interval is set as 120 frames at 30 fps, which corresponds to set T as 4s for test videos. As a result, all normal moving vehicles are removed from the frames and static vehicles remain in the background.

The background is extracted both in the forward and backward directions. The forward part is utilized to predict the candidate anomalies and the backward part is designed to refine the start time of abnormal traffic events precisely.

3.3. Extraction of Hypothetical Abnormal Mask

There is no unified definition of anomaly but basically it refers to anything we don't expect to happen normally [19]. Generally, anomalies happen on vehicles driving on the main road, and vehicles stop for a long time on parking lot are not anomaly. In order to avoid interference from stopping vehicles on the side roads and parking lots, we need to segment out hypothetical abnormal mask regions automatically. Due to the complexity of the road surface scene, and the anomalies sometimes deviate from the main road, it is hard to use the segmentation model to distinguish the hypothetical abnormal area. We propose a combined method to extract abnormal mask based on frame difference and vehicle tracking trajectory :

motion-based mask. We analyze differences between two frames to construct the abnormal mask. There are k interval frame between these two frames, if the differences exceed $diff$, we consider that the area has moving objects and retain this area. To cope with the camera shake and scene changes, we discard the result if the abnormal mask of a frame is greater than the set threshold M. Finally, we add up all the changes areas of a video to generate a motion mask.

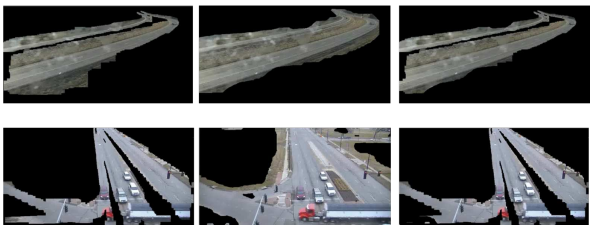

Figure 5. Examples of abnormal mask. From the left to the right: motion-based mask, trajectory-based mask, final fused mask. The top row shows motion-based mask can reduce false recall of detection, the bottom row shows trajectory-based mask can reduce the false recall of auxiliary roads.

trajectory-based mask. We use the multi-target tracking algorithm DeepSORT[26] to get the trajectory of the vehicle. For each trajectory, if the length of it is less than the threshold n or the travel distance of the trajectory is less than the threshold d, the trajectory is considered to be a false recall or a trajectory on the auxiliary road, and the vehicle detection results in this trajectory is not to be considered. For the qualified trajectories, in order to avoid the false recall of the parking lot that close to the main road, we process the detection results according to the size of the vehicle. Specifically, we narrow large vehicle detection boxes. Based on the above results, we sum up the detection result of each trajectory to the corresponding position, so as to obtain a trajectory-based mask. Additionallywe remove the connected region with small area to eliminate noises such as auxiliary road.

Finally, we take the intersection of the above two masks to get the final mask. These two mask can complement each other and Figure 5 shows some results of abnormal mask.

3.4. Multi-Granularity Tracking

In this paper, we design a multi-granularity tracking algorithm to analyze the candidate abnormal vehicles, which involves a box-level tracking branch and a pixel-level tracking branch. We illustrate the multi-granularity tracking algorithm in Figure 6.

3.4.1 Box-level Tracking Branch

To generate box-level tracking results, we first adopt the detection algorithm in section 3.1 to detect all bounding boxes, $\{B\}$ in the video frames after the forward background modeling process, with corresponding confidence scores $S(B)$. Subsequently, we link the detections across the single frame to produce a temporarily consistent spatio-temporal tube to track a particular vehicle. This box-level tracking process consists of four steps and the whole process is outlined in

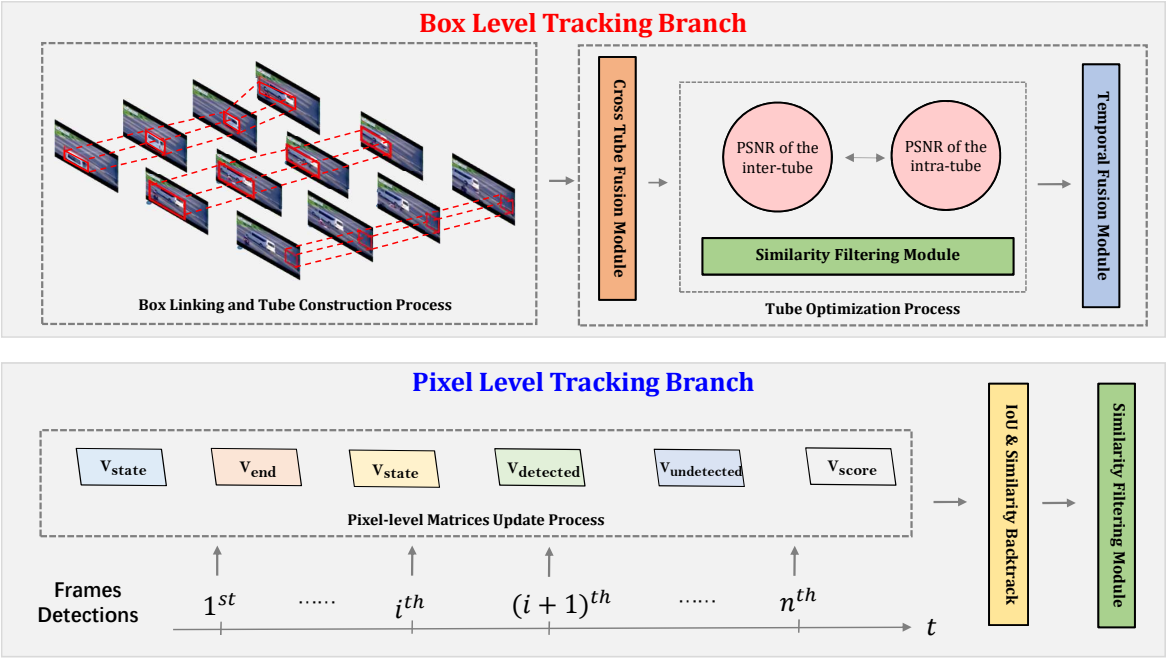

Figure 6. The illustration of multi-granularity tracking algorithm. This framework involves fusion from box-level tracking branch and pixel-level tracking branch.

detail in the Algorithm 1. The first step is the box linking and tube construction process, which can be seen as a hierarchical clustering problem. Detection results that reflect the same object in the video are grouped into one cluster. We first sort the detections according to the confidence score and pick the one with the max score B^p as the starting point of a cluster. Then the linking process is extended both forward and backward via the greedy search algorithm and the box with the max linking score in the consecutive time is added to the corresponding cluster. Specifically, the linking score $S_l(B_i, B_j)$ is defined as the intersection-over-union (IoU) of B_i and B_j. We continue the linking process until there is no box could obtain the IoU greater than λ_1. When a special cluster is constructed, we remove the linked boxes and collect a new cluster repeatedly until all boxes are grouped.

In complex traffic conditions, the detection results are crucial to the final performance of the task. To deal with possible missed and false detections, we designed two mechanisms to compensate for the detection performance. First, we compare the starting boxes of the extracted tubes one by one. When their IoU exceeds the threshold λ_1, we think that these two tubes are related to the same object and combine their tubes. Second, We observe that the abnormal vehicle has basically no pixel difference during parking, but the road background and the vehicle are generally different. Namely, the detection box area will have a big difference before the vehicle stops and after the vehicle drives away. However, if the candidate anomaly is caused by false detection of the background, the above phenomenon will disappear. Hence we introduce a similarity filtering module to filter out some false detections that are not actually vehicles. The similarity filtering module uses the Peak Signal to Noise Ratio (PSNR) as the measurement metric. Specifically, when the PSNR of inter-tube $\mathrm{PSNR}ter$ exceeds a certain value γ_a (the PSNR difference of inter-tube $\mathrm{PSNR}ter$ and intra-tube $\mathrm{PSNR}tra$ exceeds a certain value γ_b), we think they are actually background information and filter them out.

Then we merge the obtained tubes in the temporal dimension. When the end time of the current abnormal tube $t_e^{T_i}$ and the start time $t_s^{T_{i+1}}$ of the next one are within η, we think they belong to the same abnormal event and combine these tubes.

3.4.2 Pixel-level Tracking Branch

Inspired by [1], time-related pixel-level information is leveraged to predict anomaly in this paper. As shown in Figure 6, six spatial-temporal information matrices, i.e., $V_{undetected}$, $V_{detected}$, V_{score}, V_{state}, V_{start} and V_{end} are established to update each pixel information in the iterative manner.

A suspicious state is designed in [1] to record potential anomaly, which has a lower time restriction than the abnormal state. We follow this setting in this paper. When a suspicious anomaly is detected, the region of the anomaly

Algorithm 1 Box-level Tracking.

Input: Boxes set $\{B\}$; scores $S(B)$; length threshold ζ_1; linking IoU threshold λ_1; PSNR absolute threshold γ_a; PSNR relative threshold γ_r; Temporal fusion threshold η; initial tube list $L_1 = \varnothing$.
Step1: Box Linking and Tube construction process.
while $B \neq \varnothing$ **do**
  $B^p = \arg\max_{\{B\}} S(B)$;
  $B_{tp,m} = B^p$; T = $[B_{tp,m}]$;
  Forward Linking: $t = tp$
  while $S_l(B_{t,m}, B_{t+1,j}) > \lambda_1, j \in [1, N_{t+1}]$ **do**
    $B_{t+1,m} = \arg\max_{B_{t+1,j}} S_l(B_{t,m}, B_{t+1,j})$;
    T.add($B_{t+1,m}$), $t = t + 1$.
  end while
  Backward Linking: $t = tp$
  while $S_l(B_{t,m}, B_{t-1,j}) > \lambda_1, j \in [1, N_{t-1}]$ **do**
    $B_{t-1,m} = \arg\max_{B_{t-1,j}} S_l(B_{t,m}, B_{t-1,j})$;
    T.add($B_{t-1,m}$), $t = t - 1$.
  end while
  if len(T) $\geq \zeta_1$ **then**:
    L_1.add(T);
  end if
  $\{B\}$.delete(T).
end while
Step2: Cross Tube Fusion.
if $S_l(B_a^0, B_b^0) > \lambda_1$ **then**:
  Fuse T_a and T_b in L_1.
end if
Step3: Similarity Filtering.
for each T after step 2 **do**
  Compute PSNR_{tra} inside of T;
  Compute PSNR_{ter} between inside and outside of T;
  if $\mathrm{PSNR}_{ter} > \gamma_a$ or PSNR_{tra} - $\mathrm{PSNR}_{ter} < \gamma_r$ **then**:
    Discard T from L_1.
  end if
end for
Step4: Temporal Fusion.
for each T after step 3 **do**
  if $t_s^{T_{i+1}}$ - $t_e^{T_i}$ ¡ η **then**:
    Fuse T_i and T_{i+1} in L_1.
  end if
end for
Output: L_1.

and the bounding boxes in the previous frames are compared in the IoU and similarity backtracking algorithm. The backtracking strategy is described in detail in Algorithm 2. Specifically, we follow [1] to update the start time of the anomaly when the IoU is greater than 0.5 for the overlapped bounding boxes. However, sometimes the vehicle doesn't stop immediately in case of an abnormal event, which prevents us from getting an accurate start time. Therefore, we further design a similarity backtrack method, where PSNR and color histogram features are both extracted for the non-overlapped bounding boxes to measure the box similarity. Considering that the same vehicle has spatially and temporal coherence, we employ some restrictions to eliminate disturbances. The backtracking algorithm will continue until the vehicle is no longer detected in the proposed region. Furthermore, some relaxed constraints are used to expand the backtracking time to deal with discontinuous detection results.

Algorithm 2 Pixel-level Backtrack methods.

Input: A suspicious anomaly A records start time A_{start}, end time A_{end}, bounding box A_{bbox}; the j^{th} bounding box on frame i C_j^i (before A_{start}); width of a bounding box w, the shift of two boxes along width $shift_w$; similarity measurements $PSNR$ and $ColorHist$.
Params: Backtrack time threshold T_{time}, relaxed constraint satisfaction ratio T_{raio}^r; IOU threshold $T_{IOU}$and relaxed IOU threshold T_{IOU}^r; PSNR threshold $T_{PSNR}$and relaxed PSNR threshold T_{PSNR}^r; color histogram threshold T_{Color} and relaxed color histogram threshold T_{Color}^r.
$cnt = 0$; $cnt^r = 0$; $raio^r = 0$; $iou_{max} = 0$;
while $iou_{max} > T_{IOU}^r$ or $cnt < T_{time}$ or $raio^r > T_{raio}^r$ **do**
  $cnt = cnt + 1$
  $raio^r = cnt^r / cnt$;
  $s_{PSNR} = 0$; $s_{color} = 0$;
  for each j in C_j^i **do**
    if $abs(w(C_j^i) - w(A_{bbox}))/w(A_{bbox}) > 0.1$ or $shift_w(C_j^i, A_{bbox}) > 3 \times max(w(C_j^i), w(A_{bbox}))$ **then**:
      Discard C_j^i;
    end if
    $s_{PSNR} = max(s_{PSNR}, PSNR(A_{bbox}, C_j^i)))$
    $s_{color} = max(s_{color}, ColorHist(A_{bbox}, C_j^i))$
  end for
  if $iou_{max} > T_{IOU}^r$ or $s_{PSNR} > T_{PSNR}^r$ or $s_{color} > T_{color}^r$ **then**:
    $cnt^r = cnt^r + 1$
    if $iou_{max} > T_{IOU}$ or $s_{PSNR} > T_{PSNR}$ or $s_{color} > T_{color}$ **then**:
      $A_{start} = i/framerate$
    end if
  end if
  $i = i - 1$
end while
Output: A_{start}

Then we further refine to get final results for the preliminary abnormal candidate results. First, we use the similarity filtering module mentioned in section 3.4.1 to filter out some false positives that are not actually vehicles. Second, we use the similarity to backtrack the start time again, i.e., the time when the similarity has changed significantly is considered is to be a more accurate start time. Finally, we merge these results in the temporal dimension.

3.4.3 Fusion and Backtracking Optimization

As each branch helps to capture abnormal abstractions at different granularity levels, we can combine the predicted anomaly from each branch to achieve more robust results. Specifically, we take the union of the prediction results of the two branches. When both branches predict abnormal behaviors for the same video, we choose the time of the branch with earlier prediction starting time as the final results.

Considering that the results of background modeling in the forward direction may delay the appearance of vehicles, we additionally employ the results of background modeling in the backward direction to refine and trace the abnormal results. Specifically, we use the detections of the start time of the predicted anomaly to compare with the detections of the corresponding time in the backward modeling. When the number of traceback frames is less than the max traceback frame ζ_2 and the IoU between the detections is greater than a traceback IoU threshold λ_2, we update the starting time of this anomaly to the time of the current detection in backward modeling. The backtracking process is repeated until the threshold condition is not met.

4. Experiments

4.1. Experimental Setup

The track4 dataset in NVIDIA AI CITY CHALLENGE 2020 is divided into the training set and test set. Each set contains 100 videos with a length of approximately 15 minutes, a frame rate of 30 fps and a resolution of 800 $\times$ 410. The algorithm should identify all anomalies present in all 100 test set videos, and give the start time and confidence score. The anomalies can be due to car crashes or stalled vehicles. We first conduct the experiments in the training set to determine the model parameters through cross-validation. Then we directly adopt the parameters of each component obtained by cross-validation to obtain the final result in the test set.

4.2. Implementation Details

Detection Model. SENet-152 [7] is used as our detection backbone network. Specifically, Stochastic Gradient Descent (SGD) is adopted for the training process, and our model is trained with 50K iterations with the initial learning rate being 0.01 and a minibatch of 8. The learning rate is reduced by a factor of 10 at iteration 30K and 40K, respectively. Weight decay and momentum are set as 0.0001 and 0.9, respectively. We initialize our network with the weights pre-trained on COCO [13]. The shorter side of the input images is resized to 800 and the longer side is resized to less or equal to 1333. We have 5 layers feature map for FPN [12], from level 2 to level 6. We follow [21] to cluster ground truth boxes in the training dataset, and the selected anchors for each level are [16, 32, 64, 128, 256].

Extraction of Hypothetical Abnormal Mask. For the motion-based mask, the threshold of interval frame k is 5, and we extract five frames per second to calculate the changing area. The difference threshold $diff$ is 99 and M is set to 13,000. For the trajectory-based mask, the min trajectory length n is 5, and the min distance d of the trajectory is set to 50. The filtering area is 3,000 pixels for a small connected region.

Box-level Tracking. The length threshold ζ_1 is fixed to 50s and the linking IoU threshold λ_1 is 0.4. In the similarity filtering module, the PSNR absolute threshold γ_a is set to 22 and the PSNR relative threshold γ_r is 2.0; The temporal fusion threshold is set to 7000 frames.

Pixel-level Tracking. Thresholds for the normal-suspicious state transition and the suspicious/abnormal-normal state transition are fixed to 3 consecutive frames equally. Time thresholds for filtering suspicious candidates and coarse anomaly candidates are set to 20s and 40s respectively. The shortest traceback time T_{time} is 40s and relaxed constraint satisfaction ratio T^r_{raio} is 0.6. IOU threshold T_{IOU} and relaxed IOU threshold T^r_{IOU} are 0.3 and 0.5; PSNR threshold T_{PSNR} and relaxed PSNR threshold T^r_{PSNR} are 18 and 20; color histogram threshold T_{Color} and relaxed color histogram threshold T^r_{Color} are 0.88 and 0.9.

Backtracking Optimization. The max traceback frame ζ_2 is 160 frames and the traceback IoU threshold λ_2 is set to 0.6.

Table 1. Our results on Track4 test-set

F1	RMSE	S4 Score
0.9855	4.8737	0.9695

4.3. Evaluation Metric

A combined metric is adopted to evaluate the total performance of anomaly detection, which is determined in two aspects: F1-score and normalized root mean square error (NRMSE):

$$S4 = F1 \times (1 - \mathrm{NRMSE}). \tag{2}$$

The F1-score is the harmonic mean of precision and recall. Specifically, a true-positive (TP) detection is considered as the correct anomaly within (before or after) 10 sec-

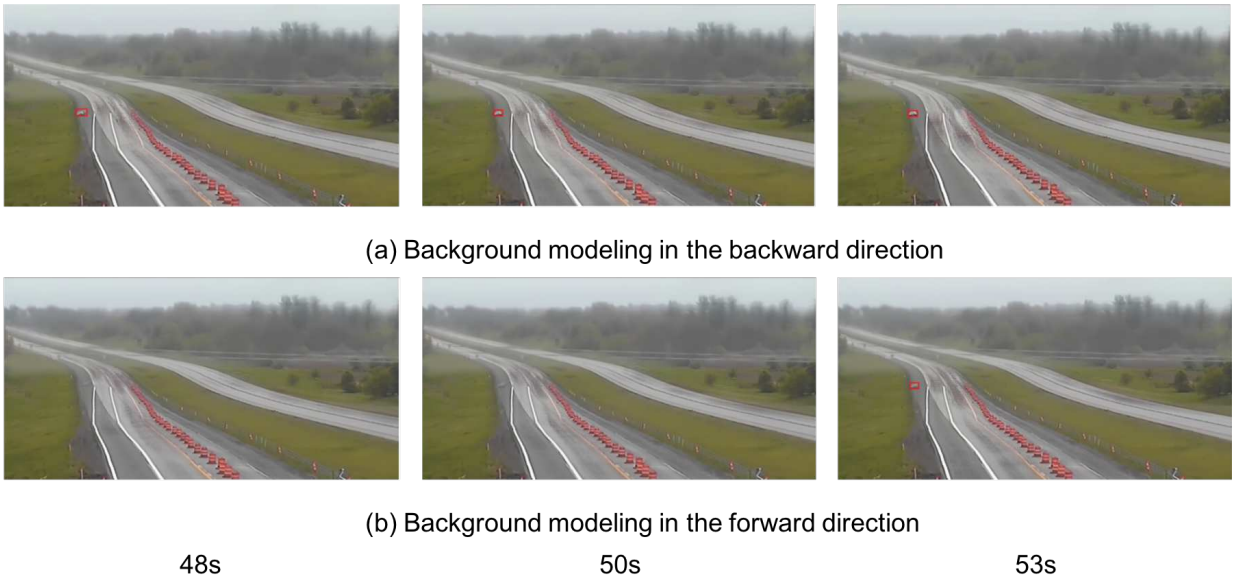

Figure 7. Example results, By background modeling in the forward direction, we can get a delayed start time of the anomaly, and then we get a more accurate time positioning by backtracking the detection result of images from background modeling in the backward direction.

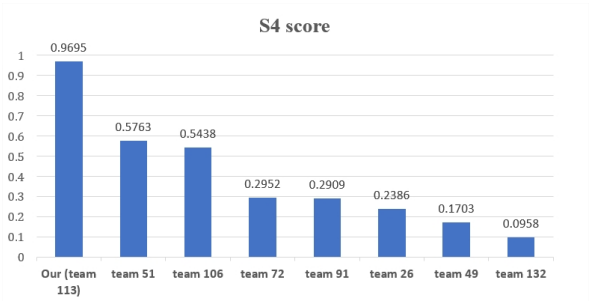

Figure 8. Compared results on the Track 4 test-set on the leaderboard.

onds of a real abnormal event. A false-negative (FN) is a real anomaly that our algorithm can not correctly predict. A false-postive (FP) denotes the predicted anomaly is not a real anomoly actually. The F1-score can be summarized as:

$$F1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}. \tag{3}$$

Normalized root mean square error (NRMSE) reveals the detection time error of the predicted time and ground truth anomaly time for all true-positive predictions. NRMSE employs a max-min normalization with a maximum value of 300 and a minimum value of 0. In short, NRMSE is defined as follow:

$$\mathrm{NRMSE} = \frac{\min(\sqrt{\frac{1}{\mathrm{TP}}\sum_{i=1}^{\mathrm{TP}}(t_i^p - t_i^{gt})^2}, 300)}{300}, \tag{4}$$

where t_i^{gt} denotes the ground truth starting time of the anomaly and t_i^p is the predicted starting time via our method.

4.4. Experimental results

We evaluate our method on the Track 4 testing data. As shown in Table 1, we achieve 0.9855 F1-score while the start time error is only 4.8737 seconds, which demonstrates the superiority and robustness of our proposed method. The final leaderboard results among all the teams are shown in Figure 8, we achieve 0.9695 S4 score and rank the first place among all the participant teams.

5. Conclusions

In this paper, we design a multi-granularity tracking approach with modularlized components, which contains the extraction of hypothetical anomaly regional mask, background modeling to eliminate dynamic traffic disturbance, the detection model to get all stopped vehicles, a multi-granularity tracking mechanism to analyze the candidate abnormal vehicles, and finally a fusion and backtracking optimization method to achieve more robust results. Results on NVIDIA AI CITY CHALLENGE 2020 show our proposed method shows promising performance, which gets a 0.9695 total score, 98.55% F1-score and 4.8737 RMSE.

References

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

[3] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1134–1142, 2015.

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

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

[6] Timothy Hospedales, Shaogang Gong, and Tao Xiang. A markov clustering topic model for mining behaviour in video. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1165–1172. IEEE, 2009.

[7] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[8] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2928. IEEE, 2009.

[9] Louis Kratz and Ko Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446–1453. IEEE, 2009.

[10] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[11] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):18–32, 2013.

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

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

[14] 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. Springe, 2016.

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

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

[17] Hajananth Nallaivarothayan, Clinton Fookes, Simon Denman, and Sridha Sridharan. An mrf based abnormal event detection approach using motion and appearance features. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 343–348. IEEE, 2014.

[18] 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.

[19] Milind Naphade, Zheng Tang, Ming-Ching Chang, David C Anastasiu, Anuj Sharma, Rama Chellappa, Shuo Wang, Pranamesh Chakraborty, Tingting Huang, Jenq-Neng Hwang, et al. The 2019 ai city challenge. In *CVPR Workshops*, 2019.

[20] Khac-Tuan Nguyen, Trung-Hieu Hoang, Minh-Triet Tran, Trung-Nghia Le, Ngoc-Minh Bui, Trong-Le Do, Viet-Khoa Vo-Ho, Quoc-An Luong, Mai-Khiem Tran, Thanh-An Nguyen, et al. Vehicle re-identification with learned representation and spatial verification and abnormality detection with multi-adaptive vehicle detectors for traffic video analysis. In *Proc. CVPR Workshops*, 2019.

[21] 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.

[22] 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.

[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] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019.

[25] 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.

[26] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[27] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015.

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

[29] 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 *IEEE/CVF Conference on Computer Vision Pattern Recognition Workshops*, 2018.

[30] Jiangong Zhang, Laiyun Qing, and Jun Miao. Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection. In *IEEE International Conference on Image Processing*, pages 4030–4034. IEEE, 2019.

[31] Bin Zhao, Li Fei-Fei, and Eric P Xing. Online detection of unusual events in videos via dynamic sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3313–3320. IEEE, 2011.

[32] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1237–1246, 2019.

[33] Yi Zhu and Shawn Newsam. Motion-aware feature for improved video anomaly detection. *arXiv preprint arXiv:1907.10211*, 2019.

[34] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 2004.

[35] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):p.773–780, 2006.