A Comparative Study of Faster R-CNN Models for Anomaly Detection in 2019 AI City Challenge

Linu Shine, Anitha Edison, Jiji C. V.
College of Engineering, Trivandrum
Kerala, India
linushine@cet.ac.in, anithaedison@cet.ac.in,jijicv@cet.ac.in

Abstract

Traffic anomaly detection forms an integral part of intelligent traffic monitoring and management system. Timely detection of anomalies is crucial in providing necessary assistance to accident victims. Track 3 of 2019 AI city challenge addresses traffic anomaly detection problem. We propose an unsupervised method to tackle this problem. Proposed system consists of three stages. The first stage is a background extraction stage which isolates the stalled vehicles from moving vehicles. An anomaly detection is the second stage that identifies the stalled vehicles in the background and finally anomaly confirmation module confirms anomaly and determines the start time. We have used faster RCNN (FRCNN) with Inception v2 and ResNet 101 to detect stalled vehicles and confirm possible anomalies. A comparative study shows that FRCNN with Inception v2 gives superior performance.

1. Introduction

Automated detection of traffic anomalies has become an integral part of road safety, now more than ever. Traffic cameras are deployed in every nook and corner, generating Terabytes of data every millisecond. Combing through huge volume of data to retrieve relevant information is a herculean task making human monitoring time consuming and inefficient. To properly utilize the generated data, automatic means to monitor video is required. A main area that requires automated monitoring is abnormal event detection from traffic videos. Existing abnormal event detection methods focus mainly on pedestrian anomalies[7, 14]. Detecting traffic anomalies is gaining importance as the future is inclined towards an intelligent transportation culture. Challenges like 2019 AI City Challenge helps in focusing the attention of researchers to the most relevant social problems and coming up with most promising solutions. Prime deterrent in the path of fully automated traffic anomaly detection is the lack of annotated traffic videos; this is where transfer learning becomes relevant. We propose an unsupervised transfer learned system for abnormal event detection in traffic videos.

Anomaly detection is not a binary classification problem of differentiating between normal and abnormal frames, mainly because the definition of normal and abnormal events differ from scene to scene. It cannot be approached as a typical classification problem, since it is not feasible to collect all abnormal events in a scenario. Common approach is to learn the normal patterns in a scene and identify any deviation as anomaly. This method is applicable to surveillance scenarios captured by UCSD, Avenue and Shanghai abnormal event detection datasets, which includes monitoring of a single or homogeneous scenes. But this cannot be applied to traffic analysis and monitoring, since data obtained is from multiple camera feeds.

This paper presents an unsupervised method for traffic anomaly detection. Our method uses background extraction followed by object detection to identify stalled vehicles. Anomaly is confirmed if the detection is consistent for a specific time window in the back ground. Our training phase does not involve explicit labelling of anomalies, instead a rule based decision module which does not require any training is used to detect anomalies. Rest of the paper is organised as follows. Section 2 reviews some related work in anomaly detection. Our proposed method for detecting traffic anomalies is detailed in Section 2. Experiments and discussions are included in Section 4 and Section 5 concludes the paper.

2. Related Work

A number of attempts have been made to automatically detect abnormal events in videos. Research going on in the area of anomaly detection in videos can be categorized into a number of classes depending on the training or learning framework used.

Some of the earliest methods used statistical models to

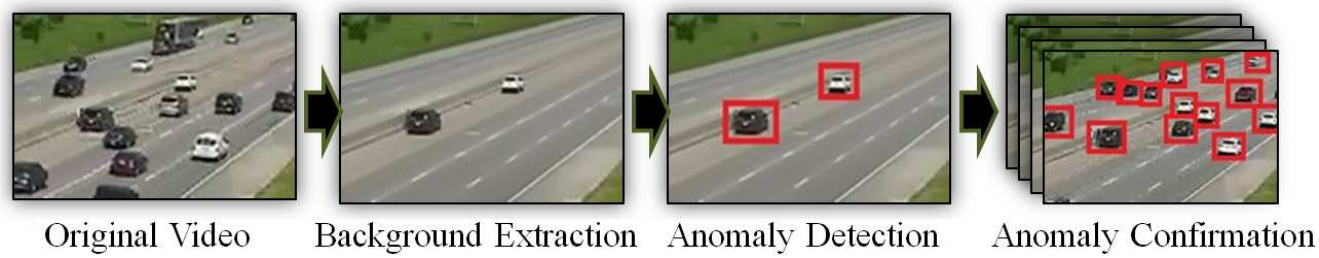


Figure 1: Flow diagram of the proposed traffic anomaly detection method. Proposed system consists of three stages. Background extraction stage isolates the stalled vehicles from moving vehicles. Anomaly detection stage identifies the stalled vehicles in the background and anomaly confirmation confirms anomaly and determines the start time of anomaly

capture the relevant information about normal activities. Hidden Markov Models (HMM) are the most popular statistical model used to detect irregular events in a scene [11]. HMM was used to model motion patterns in both space and time to detect anomalies in crowded scenes. Gaussian mixture models (GMM) and Markov Random Fields (MRF) [18] are used as well. Ryan *et al* used textures of optical flow to train a normality model using GMM [20].

A major class of algorithms use binary classifiers like support vector machines (SVM). In these methods, descriptors extracted from videos were used to train a classifier. Wang *et al.* suggested the use of a motion descriptor, histogram of optical flow orientation extracted from normal videos to train a one class SVM [23] for recognizing unusual actions. Cui *et al.* explored use of interaction energy potentials and velocity of actions to train an SVM classifier to detect abnormal interactions [4].

Nearest neighbour clustering of features extracted from optical flow were also used to differentiate normal and abnormal events[2]. Vehicle trajectories were clustered using fuzzy K-means clustering to detect abnormal vehicle patterns[6].

Another popular set of works perform abnormal event detection in a sparse reconstruction framework. Here, a sparse dictionary learned from normal training features is used to represent the test videos. A sparse reconstruction cost (reconstruction error) is estimated and thresholded to detect anomaly. An important work in this category learns a dictionary from multi-scale histogram of optical flow (MHOF) features extracted from normal training videos [3]. MHOF consist of histogram of optical flow quantised to 16 bins which include two scales of 8 bins each. Lower 8 bins correspond to flow magnitude less than a threshold while the upper 8 corresponds to that greater than the threshold. Since the bins depend on magnitude values of optical flow, the authors claim that their feature descriptor keep tab of the motion energy information in addition to motion direction. Li *et al*. used histogram of maximal optical flow projections (HMOFP) in the same sparse reconstruction framework for abnormal event detection in crowded scene [12]. HMOFP as the name suggest is obtained by projecting the optical flow vectors in each bin to angle bisector of that bin and choosing the magnitude of maximal projection as descriptor corresponding to each bin. Zhao *et al.* used Histogram of oriented gradients (HOG) and histogram of optical flow (HOF) descriptors computed in a space-time cuboid around spatio-temporal interest points for sparse dictionary learning [27]. The dictionary was updated online as more and more samples were observed and they used fast sparse coding solvers for enhanced speed. A very fast approach for abnormal event detection proposed by Lu *et al* uses spatio-temporal gradient for sparse learning[14]. They used a sparse combination learning instead of sparse dictionary learning and this improved the detection speed considerably. To obtain the spatio-temporal gradient, the video is divided in space-time volumes of specific size called cuboids. Motion pattern is represented by collecting the spatial and temporal gradients of each pixel in the cuboid. An acceleration descriptor was used in sparse reconstruction framework to detect anomalies [5].

Deep learning techniques, especially, convolutional neural networks (CNN) have provided efficient solution for various computer vision tasks such as object detection, image classification, action recognition, etc. Despite the success of CNNs in these fields, the unavailability of supervised data for training limits the use of CNNs for anomaly detection. Hasan *et al*. used convolutional autoencoders to learn motion features and normal patterns from training video sequences [7]. Learned network can reconstruct normal sequence with small error while irregular sequences result in large reconstruction error. To improve capturing the spatio-temporal information, long short term memory (LSTM), a powerful network for capturing sequential information was used together with convolutional autoencoders to form convolutional LSTM autoencoder [1, 16]. Temporally coherent sparse coding was introduced to preserve similarity between neighboring frames by Luo *et al*. [15]. They have weighted sparse coefficients of adjacent frames

using a similarity measure and implemented it as a special case of recurrent neural network (RNN) which was called stacked RNN (sRNN). Detection and recounting of abnormal events using anomaly detector trained on faster RCNN output was proposed by Hinami *et al*. [9]. A two-stage cascaded deep convolutional network which provides a faster localization of anomalies is proposed in [21]. Deep learning methods like contractive and variational autoencoders were also used for detection of abnormal events in videos [10]. A recurrent variational autoencoder that uses motion as well as appearance information was proposed for event detection[26]. State-of-the-art anomaly detection results on all standard datasets were given by generative adversarial network trained on normal videos [13]. The method used future frame prediction with an additional constraint on motion along with appearance based constraints.

3. Methodology

The objective of the 2019 AI city challenge is identifying traffic anomalies and the time they occur, with a tolerance of 10 seconds. Traffic anomalies in this challenge are mainly vehicles stalled on road or in the hazard lane due to collisions, breakdown etc.

As shown in figure 1 proposed method consists of three stages, background extraction, anomaly detection and anomaly confirmation. Once a vehicle halts, it becomes part of the background. By extracting the static background of traffic videos, the stalled vehicles can be isolated. The anomaly confirmation module checks if the detected vehicle is result of a traffic anomaly and if yes; it corrects the time of stopping by performing a second stage detection in corresponding original video.

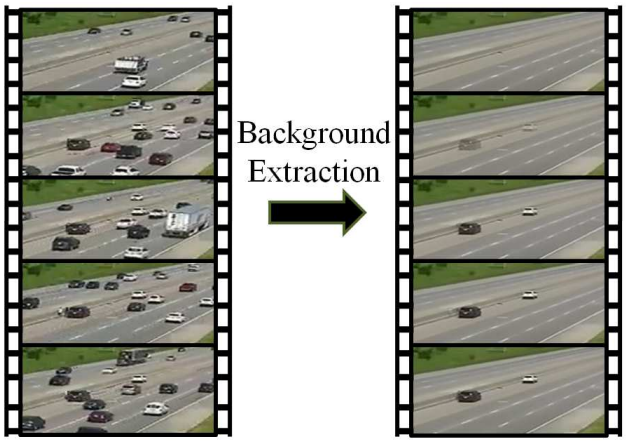


Figure 2: Example of background extraction using AGMM method. Stalled vehicles appear in the background after a few frames

3.1. Background extraction

Core module of the proposed system is a background extractor which models the static background in traffic videos.

For background extraction we use improved adaptive Gaussian mixture model proposed by Zivkovic [28]. The basic idea is to represent the static parts of an image with a statistical model. In this method, a mixture of Gaussian distributions are used to represent each pixel in the model. When a new frame arrives, the parts of the image that do not fit to this statistical Gaussian model, is grouped as foreground object. The value of a pixel at any time t is denoted by Φ_t. Now a Bayesian decision Υ (Equation 1) decides whether pixel belongs to back ground(BG) or foreground(FG).

$$\Upsilon = \frac{P(BG/\Phi_t)}{P(FG/\Phi_t)} = \frac{P(\Phi_t/BG)P(BG)}{P(\Phi_t/FG)P(FG)} \quad (1)$$

This method is impervious to variations in lighting, slow moving objects, noise like image jitter and addition or removal of new objects to the scene.

Figure 2 shows an example of the background extracted using this method. All moving vehicles constitute the foreground elements and all stationary elements in the video form the background. Whenever a moving vehicle comes to a stop and remains stationary within the field of view of the camera for some time, it will be considered as part of the background. As shown in figure, the stalled vehicles appear in the background after a few frames. Once the stalled vehicle starts moving, this vehicle gets removed automatically from the background by the background updating process of the module. The extracted background is passed to the anomaly detection module to check for the presence of any stalled vehicles.

3.2. Anomaly detection

We assume that the anomalies are caused due to vehicles that are stationary for considerable period of time. After some time the stalled vehicles become a part of the extracted background. In order to detect the anomaly, the next major step is to identify vehicles in the background.

The main aim of the module is to correctly identify all stalled vehicles as soon as it appears in the background. We have used faster region based convolutional neural network (FRCNN)[19] to detect vehicles in the background. In FRCNN framework, image features are extracted using a CNN network. We have experimented using two different networks; ResNet 101 [8] and Inception v2 [22] (FRCNN-ResNet 101 and FRCNN-Inception v2). Inception uses concept of wider network while ResNet goes for deeper; each has its own advantage. ResNet uses skip connections to avoid vanishing or exploding gradients, thereby facilitating the use of deeper layers to extract finer details from the

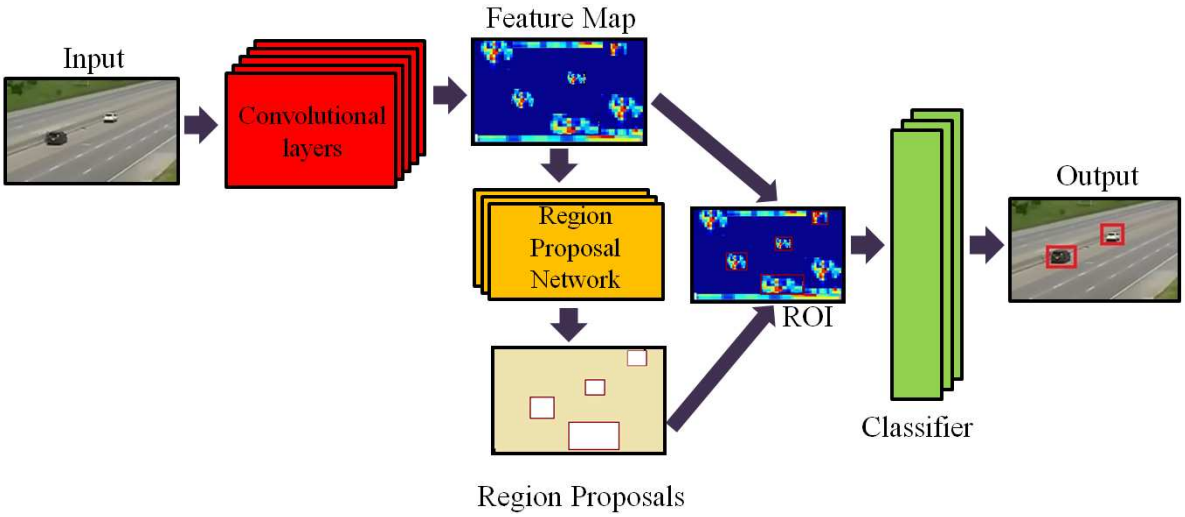


Figure 3: Flow diagram of faster RCNN used for vehicle detection. We have experimented using ResNet 101 and Inception v2, in this framework.

image. Inception performs convolution operation with different sized kernels to accommodate the scale variations of objects in the image. To address vanishing gradient problem, auxiliary classifiers are used to calculate losses during training. In Inception v2, $m \times m$ convolution layers are implemented using stacked $1 \times m$ and $m \times 1$ layers which are found be computationally cheaper and faster. The basic modules of ResNet and Inception network are shown in figure 4.

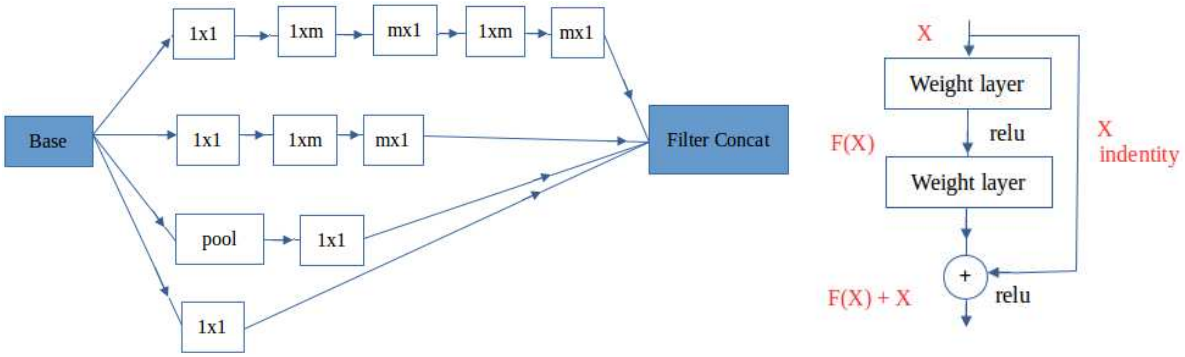
From the feature map extracted using CNN, a region proposal network predicts locations that may contain anomalies. It provides objectness score and bounding box coordinates of the object. The objectness score is the probability that an object is present in the proposed region. Each proposed region undergoes ROI pooling to convert the proposals into fixed size feature maps for classification. This is further passed to the CNN and then branches to fully connected layers for object classification and bounding box regression.

Figure 5 (a) shows the flow diagram of our anomaly detection stage. In this stage FRCNN model identifies vehicles in the extracted background, for each new detection the centroid of bounding box, the start frame of the first detection and a new anomaly counter is initialized. In subsequent frames if a vehicle is detected in the same position as the saved detection the counter is incremented; if not the new detection is saved. This process is repeated till the last frame of the video. If any of the anomaly counters exceed the threshold value, confirmation module is triggered. Suitable selection of threshold value helps in eliminating false positives like vehicle stoppage at traffic signals.

3.3. Anomaly confirmation

Whenever stalled vehicles become part of background, they are detected by the anomaly detection module. Next step is to validate if the detection is a real anomaly and if so, extract start time of anomaly. In the anomaly detection module, if frequency of any detected vehicle from the background is more than a particular threshold, then it is considered as a potential anomaly. The centroid of corresponding detection is passed on to the anomaly confirmation module. As a verification step and to get the anomaly start time, the object detector is run on the corresponding original video. If at any time the centroid of a detected vehicle in a video matches with that of anomalous vehicle from the background extracted video, then its frame number is noted as start frame of anomaly. A confirmation counter is incremented every time the vehicle spotted at the same position continuously for successive frames. The detections are monitored continuously for 1 minute. If the confirmation counter value is greater than a threshold, the anomaly is confirmed and start time of the detection is computed from anomaly start frame. The corresponding video number and anomaly time are written into a text file. Flow chart of anomaly confirmation module is shown in figure 5 (b).

In our framework, false anomaly detections were mainly caused by vehicles parked on private property. Since these vehicles are stationary they become part of the extracted background, but they are not anomalies. In object detection module, parked vehicles are detected, but they are not detected continuously in most of the original videos. So if the counter is set to a proper threshold value, false anomalies due to parked vehicles can be removed. Similarly, other false positives are also removed by the two stage detection,



(a) (b)

Figure 4: Basic module of (a) Inception v2 (b) ResNet. Inception captures multi-scale features using wider networks while ResNet allows extraction of finer details using deeper networks.

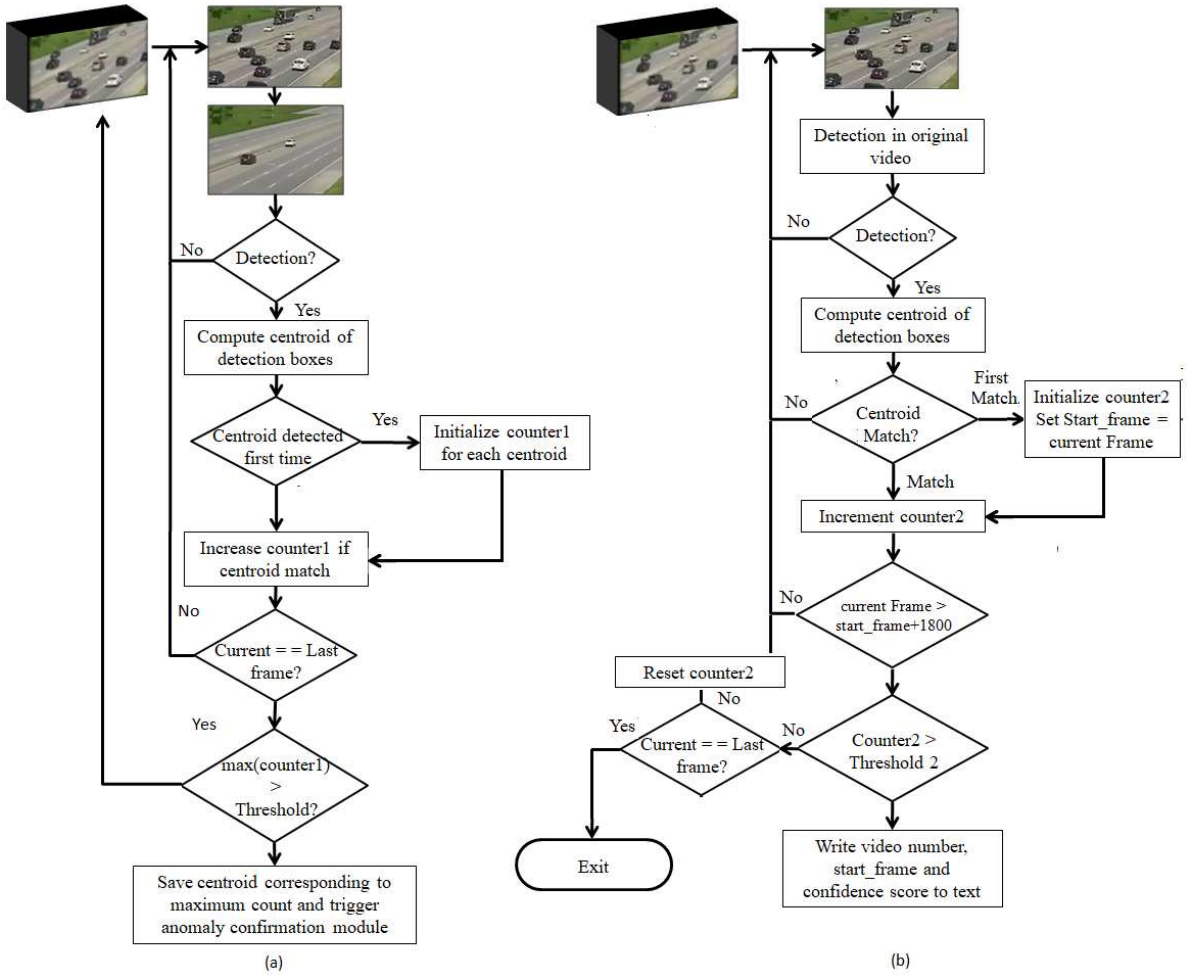


Figure 5: Flow chart of our algorithm (a) Anomaly Detection Module (b) Anomaly Confirmation Module. Anomaly detection module detects potential anomalies and pass them on to anomaly confirmation stage for verification and to obtain exact start time of anomaly

since false detections do not occur in the same position in background and original video. Further to reduce false positives, the detection threshold is suitably adjusted.

4. Experiments and discussions

In this section we present the experimental validation of our method used for traffic anomaly detection. The proposed method is tested on track 3 test data of 2019 AI City challenge. Datasets used for training FRCNN models, evaluation measures for track 3 AI City challenge and experimental results with experimental settings are detailed in following subsections.

4.1. Datasets

In addition to the train data of AI city challenge, we have used parts of UA-DETRAC dataset [25] and UAV 123 dataset [17] for training our anomaly detection module.

4.1.1 AI city Track 3 dataset

This dataset contains 100 train and 100 test videos of resolution 800×410. Each video is recorded at 30 fps and is approximately 15 minutes long. Both test and train videos contain anomalies due to car crashes or stalled vehicles. Our method is unsupervised, implying that no information regarding whether a video contains an anomaly or not, is provided during training. Our anomaly detection module to detect stalled vehicles, is a FRCNN object detector and this was trained using transfer learning. Random 460 original and background frames are picked from train video and the vehicles are manually annotated for training.

4.1.2 UA-DETRAC dataset

UA-DETRAC is a widely used multi-vehicle detection and tracking dataset. It consists of videos recorded at 25 fps, with resolution of 960×540 pixels. There are more than 140,000 frames and 8250 vehicles that are manually annotated. We have used a subset of UA-DETRAC dataset provided by [24]. Only 72,402 images out of 140,000 images from DETRAC dataset are used to train our detector.This includes 5,31,402 labelled bounding boxes. Multi-object annotations are converted to single object annotation (car).

4.1.3 UAV123 Dataset

UAV123 dataset contains a total of 123 video sequences with more than 110K frames. It consists of videos recorded at 30 fps. 3000 manually annotated images from UAV dataset are used for training the detection network. This dataset contains only one or two vehicles in each frame, so the detector will get fine-tuned to differentiate the background. Training Faster RCNN on UAV dataset reduced the false positives

4.2. Evaluation Measures

The evaluation metrics of the challenge are F_1-score, root mean square error(RMSE) of detection time and S_3-score. For evaluating F_1 score, a detection is considered true positive (TP) only if the anomaly is detected within ± 10 seconds from the onset of anomaly. F_1-score is the harmonic mean of precision and recall. A normalized RMSE (NRMSE) is obtained by min-max normalization from 0 to 300. RMSE score above 300 is normalized to 1. S_3-score is evaluated as

$$S_3 = F_1 \times (1 - NRMSE) \tag{2}$$

4.3. Experimental Settings

As mentioned in section 3.2, we have experimented the use of FRCNN-ResNet 101 and FRCNN-Inception v2 for vehicle detection. FRCNN-ResNet 101 pre-trained on Kitti dataset and FRCNN-Inception v2 pre-trained on COCO dataset is further trained with DETRAC and additional images from AI City Track 3 train dataset. For eliminating additional false positives, this networks are further trained on UAV123 dataset. For ease the aforementioned networks will be referred to as ResNet 101-DETRAC, Inception v2-DETRAC, ResNet 101-UAV and Inception v2-UAV respectively henceforth in the paper.

We used two different machines for training and testing the detectors. ResNet 101-DETRAC and ResNet 101-UAV are trained on 3.7 GHZ Intel Xeon W-1245 CPU with 32 GB RAM and 8 GB Nvidia quadro P4000 GPU, while Inception v2-DETRAC and Inception v2-UAV are trained on 3.0 GHZ Intel Xeon E-51660 CPU with 16 GB RAM and 8 GB Nvidia quadro M4000 GPU. We trained the detectors for 2,00,000 epochs. It took approximately 48 hours for training ResNet 101-DETRAC and 50 hours for Inception v2-DETRAC. Testing on 100 videos of 15 minutes each, took nearly 96 hours for ResNet 101-DETRAC and 75 hours for Inception v2-DETRAC. Training Inception v2-UAV and ResNet 101-UAV took approximately 28 hours. The huge time requirement for training and testing the videos, deterred experimentation on other state of the art architectures like RetinaNet.

4.4. Experimental Results

The background extraction module provided a clear background of traffic videos along with stalled vehicles, if any. The crux of the challenge was to correctly detect all the stalled vehicles and identify the exact start time of anomaly.

Initially the confirmation module was entirely based on extracted background. On retrospection, we found there is a delay for stalled vehicle to appear in the extracted background as shown in figure 7. Figure is an example of difference in detection time in extracted background and original video. A stalled vehicle is detected in the frame 5559 in

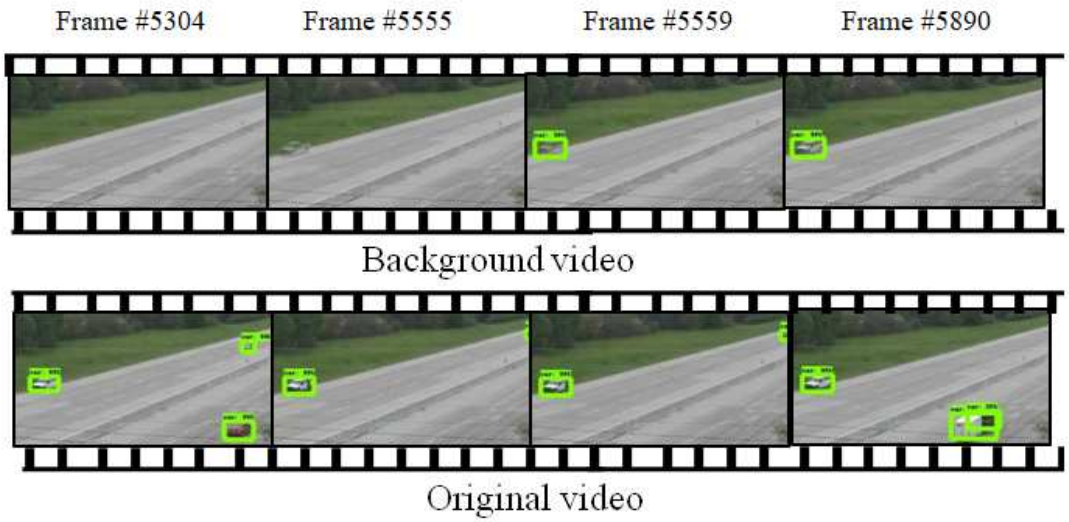


Figure 6: Frame difference of anomaly detection in background and original videos

Figure 7: Sample detection outputs of FRCNN using Inception v2-DETRAC, ResNet 101-DETRAC, Inception v2-UAV and ResNet 101-UAV

background image and its takes few more frames to get clear image of vehicle in the background. While in the original video vehicle appears and is detected at the same anomalous position in frame 5304. It takes 8.5 seconds to get a

detection of anomalous vehicle in background. So when detection models are run on the background video, a constant correction factor in time is applied to obtain start time of anomaly. The results are shown in Table 1. RMSE score is too high. It is found that the correction factor depends on quality of the video and efficiency of detector module. So correction factor varies with videos and hence a fixed correction in time cannot be applied, hence we used two stage detection.

Detection Network	F1	RMSE	S3
Inception v2-DETRAC	0.2813	274.4512	0.0240
Resnet 101-DETRAC	0.3333	265.9480	0.0378

Table 1: Comparison of anomaly detection before time correction

In two stage detection, once anomaly is detected in background image, the start time of anomaly is collected using detection on original video. The reference position of anomaly from the background, is used to mark the time of anomaly in second stage. Table 2 shows the results when anomaly start time is marked using detection on original video. RMSE score of both detectors has improved with the change in detection model.

Detection Network	F1	RMSE	S3
Inception v2-DETRAC	0.5965	249.7808	0.0999
Resnet 101-DETRAC	0.4483	200.3549	0.1489
Inception v2-UAV	0.6462	131.2005	0.3636

Table 2: Comparison of anomaly detection after time correction

Vehicles waiting in the traffic signal and parked vehicles appear in the extracted background and will be detected. These detections are not anomalies and can be eliminated by selecting the detection count suitably. These false positives are further reduced by training detectors using UAV123 dataset. Faster RCNN trained on UAV123 dataset takes considerable time for detecting a vehicle in background, so the number of detections in the same position is less. In most of the cases the parked vehicle are not detected by FRCNN with inception, trained on UAV dataset. As shown in table 2, this considerably reduces the false positives thereby increasing the F1 score.

Figure 6 shows a comparison of detection outputs. In figure 6(c) the false positives detected by Inception v2-DETRAC is eliminated when trained using UAV123 dataset. Further it is noted that training using UAV123 also improved the detection rate as shown in figure 6(d).

Our model could not detect vehicles which are too far from the camera. It is also unable to identify the similarity in detected anomalous vehicles when the camera is zoomed. As a solution we plan to check the feasibility of using a re-identification module for anomaly confirmation. Further training can be used to detect very small vehicles. Another future work will be to eliminate false positive detections by using a vehicle classification stage. We also plan on using lane detection module to eliminate false anomalies due to parked vehicles.

5. Conclusion

In this paper we propose a three stage system for detecting anomalies in traffic videos beginning with a background extraction stage followed by two level detection of stalled vehicles in original and extracted background videos. First detection stage in background videos filters all possibles anomalies, which are later confirmed using a second stage detection in original video. We have compared two FRCNN models for anomaly detection and found that FRCNN-inception v2 network trained on UAV123 dataset performs better than FRCNN-ResNet 101 network. Our future direction will be to eliminate false positive detections by using an additional classifier and lane detection module. Another future direction is to check the feasibility of using a re-identification module for anomaly confirmation.

References

[1] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *Proc. of International Symposium on Neural Networks*, pages 189–196, 2017.

[2] Rensso Victor Hugo Mora Colque, Carlos Caetano, Matheus Toledo Lustosa de Andrade, and William Robson Schwartz. Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos. *IEEE Trans. on Circuits and Systems for Video Technology*, 27(3):673–682, 2017.

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

[4] Xinyi Cui, Qingshan Liu, Mingchen Gao, and Dimitris N Metaxas. Abnormal detection using interaction energy potentials. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3161–3167, 2011.

[5] Anitha Edison and CV Jiji. Optical acceleration for motion description in videos. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1642–1650, 2017.

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

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

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Ryota Hinami, Tao Mei, and Shin'ichi Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In *Proc. of IEEE International Conference on Computer Vision*, pages 3619–3627, 2017.

[10] B Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *arXiv preprint arXiv:1801.03149*, 2018.

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

[12] Ang Li, Zhenjiang Miao, Yigang Cen, and Qinghua Liang. Abnormal event detection based on sparse reconstruction in crowded scenes. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1786–1790, 2016.

[13] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6536–6545, 2018.

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

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

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

[17] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *European conference on computer vision*, pages 445–461. Springer, 2016.

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

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

[20] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. Textures of optical flow for real-time anomaly detection in crowds. In *Proceedings of the IEEE international conference on advanced video and signal based surveillance*, pages 230–235, 2011.

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

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[23] Tian Wang and Hichem Snoussi. Detection of abnormal visual events via global optical flow orientation histogram. *IEEE Trans. on Information Forensics and Security*, 9(6):988–998, 2014.

[24] 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.

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

[26] Shiyang Yan, Jeremy S Smith, Wenjin Lu, and Bailing Zhang. Abnormal event detection from videos using a two-stream recurrent variational autoencoder. *IEEE Trans. on Cognitive and Developmental Systems*, 2018.

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

[28] 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):773–780, 2006.