Fast Unsupervised Anomaly Detection in Traffic Videos

AKSHAYA SURENDRAN, Technical University Dortmund, Germany

Road traffic control has been a topic of interest for a long time to ensure the safety of vehicles and pedestrians. However, events such as accidents or natural disasters cannot be avoided. Therefore, it is crucial to be prepared and to be able to take counteractions on time to prevent human losses. Also, due to the advancement in intelligent transportation systems, anomaly detection in traffic videos has recently gained attention. This report summarizes an unsupervised anomaly detection method in traffic videos, which comprises one deep learning-based object detection module and two statistical decision-making modules.

ACM Reference Format:
Akshaya Surendran. 2022. Fast Unsupervised Anomaly Detection in Traffic Videos. 1, 1 (February 2022), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Anomaly detection (also outlier detection) is the identification of rare events, or observations that raise suspicions by deviating significantly from the majority of the data. It has many applications in a variety of domains like system health monitoring, fraud detection, etc. This report focuses on its application in identifying outliers in traffic videos such as roads accidents, fires, etc. Further, there are three broad categories of anomaly detection techniques, namely - supervised, unsupervised and semi-supervised. The method summarized here used an unsupervised approach, which detects anomalies in an unlabeled test dataset. It works on the principle that most data points in the unlabeled data set are "normal" and looks for data points that differ from the "normal" data points [4][10].

This report summarizes the method proposed by the authors Keval Doshi and Yasin Yilmaz [2], i.e., fast unsupervised anomaly detection system in traffic videos, a novel framework composed of the nearest neighbor and K-means clustering algorithms to detect anomalies. The method focuses on outliers related to the stationary vehicles on the highways. It comprises three modules, namely: preprocessing, candidate selection and backtracking anomaly detection. The proposed method ranked 2^{nd} in the NVIDIA AI CITY challenge 2020.

The proposed method in the paper is summarized in section 2. Section 3 presents the dataset and evaluation criteria of the experiment performed. The scope of further improvements is discussed in section 4. And finally, the summary of the report is provided in section 5.

2 PROPOSED METHOD

This section introduces the proposed model. The paper focuses only on the stationary objects in the video, specifically cars, and trucks. The idea behind it is in most cases when an accident happens, or some anomalous activity occurs, the vehicles tend to come to a halt. The following subsections detail the three modules of the model, that are:

Author's address: Akshaya Surendran, akshaya.surendran@tu-dortmund.de, Technical University Dortmund, Germany.

- Preprocessing Module: This module focuses on detecting the stationary objects in the video. It comprises three stages - background modelling, road segmentation, and object detection.
- Candidate Selection Module: This module aims at removing the misclassified objects such as traffic signals etc and localizing the potential anomaly regions, using nearest neighbor and K-means clustering respectively.
- Backtracking Anomaly Detection Module: This module outputs the true onset time of the anomaly.

2.1 Preprocessing

Preprocessing is the first stage of the model. This module outputs the stationary objects detected in the videos. In the following subsections, the sub-modules of this stage are discussed.

2.1.1 ***Background Modelling.*** Background modelling is an efficient way to obtain or filter out foreground objects in a video. It is one of the major tasks in the field of computer vision that aims at detecting changes in image sequences. As this method, focuses on anomalies related to stationary objects, background modelling is done using the moving average technique to suppress the moving vehicles in the video and to emphasize the stationary objects.

The idea behind the averaging technique is to detect active objects from the difference obtained from the background model and the current frame. In this method, the running average over the current frame and the previous frames is computed in a sequence of frames in a video. This gives the background model and any new object introduced becomes the part of the foreground. Then, the current frame holds the new object with the background. Then the absolute difference is computed between the the current frame (which is newly introduced object) and background model (which is a function of time) [12]. For a given video V with N frames $F^1, ..., F^N$, the weighted sum or running average is obtained using the below-mentioned equation:

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

where F^t_{avg} is the averaged image at time $t = 100, ..., N$ and α is the update rate which decides the speed of updating and m is a fixed interval. If the threshold alpha is set to a higher value, the average image tries to catch fast and shortchanges in the data. A lower value would not consider fast changes in the input images as part of the background. In this work, the α value is set to 0.1 and m to 30. And to further reduce the complexity for averaging, a sampling period of 100 is used, i.e., only 1 frame in every 100 frames is considered. For eg: if a video has 30 frames in 1 sec, and the video is 5 minutes long then there would be 9000 frames and in this method, only 80 would be considered.

2.1.2 ***Road Segmentation.*** Since the primary objective in this work is detecting stationary vehicles on the highways, ignoring any other stationary vehicle detected in the nearby parking lots or surrounding areas in the background. Also, assuming any anomalous activity would occur only on the highways. Therefore, in order to just focus on the highways, segmentation maps of the roads are extracted by using an unsupervised approach.

Once a moving vehicle is detected in a video, the frequency map for the image is continuously updated. Frequency in this context means the rate of change of intensity per pixel. Then the image is normalized and binarization is done, i.e., assigning 1 to the area of interest and 0 to the other regions or vice versa, to extract the segmentation map (S) as shown in Figure 1 [1].

2.1.3 ***Object Detection.*** The proposed method uses YOLO(version 3) [9] for detecting objects in the video. YOLO is an abbreviation for the term 'You Only Look Once'. This is a single-stage object detection and classification algorithm

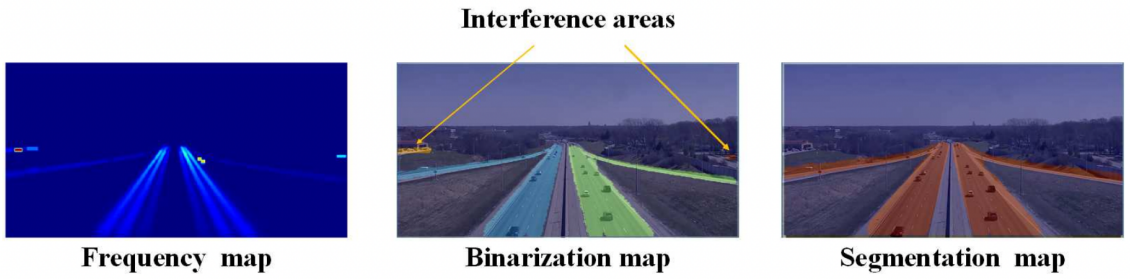

Fig. 1. From left to right, it respectively represents the effect of binarization and filtering small connected areas. As shown in the middle figure, the small connected regions are often interference areas such as parking lots or houses [1].

that uses neural network. Here, object detection is considered as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities[8].

Object detection combines the two tasks of localization an object and classifying it. Object localization refers to identifying the location of one or more objects in an image and drawing a bounding box around them. A bounding box is an outline that highlights an object in an image. In figure 2 (last image - final detections), there are three bounding boxes around the three objects - dog, cycle, and car. Each bounding box is defined by some attributes. They are the probability of a bounding box containing an object, the center of the object with respect to the image, the height and width of the bounding box. If there are more than two objects in an image, the algorithm divides the image into S × S grid as shown in the figure 2. The grid where the center of the object is located is responsible for detecting that object[8].

Each grid predicts the bounding box and calculates the confidence scores for those boxes. These scores reflect the confidence and the accuracy of the model in predicting an object. The confidence score is calculated using the intersection over union (IOU) and the probability of whether a bounding box contains an object. If no object exists in that cell, the confidence scores should be zero. Otherwise, the confidence score ideally should be equal to the IOU between the predicted box and the ground truth. IOU is a concept that is used when the model predicts multiple bounding boxes for the same object with different probabilities. The overlapping area between the different bounding boxes predicted for the same class in an image is seen, and if it is above a certain threshold then the bounding boxes are considered as one. The IOU threshold set in this paper is 0.3.

The grid also predicts the class of the bounding box. This works as a classifier and gives probability distribution over all the possible classes. Now, the confidence score for the bounding box and the class prediction are combined into one final score, representing the probability that a particular bounding box contains a specific type of object. These scores represent both the probability of that object class appearing in the box and the performance of the model. Now, although each grid predicts multiple bounding boxes, most of them would have a low confidence score, therefore a threshold is set and the boxes with a final score below the threshold are not considered. The threshold set in this proposed method for the final score is 0.1. It can be seen in the last image in figure 2, the algorithm has successfully detected the three objects in the given image [8].

In this paper, the authors have leveraged transfer learning to detect objects using YOLO, pretrained on the MS-COCO dataset to localize the regions of potential anomalies. MS-COCO is large-scale object detection, segmentation, and captioning dataset from Microsoft. It consists of 80 object categories [7]. Now, for each object detected in the F^t_{avg}

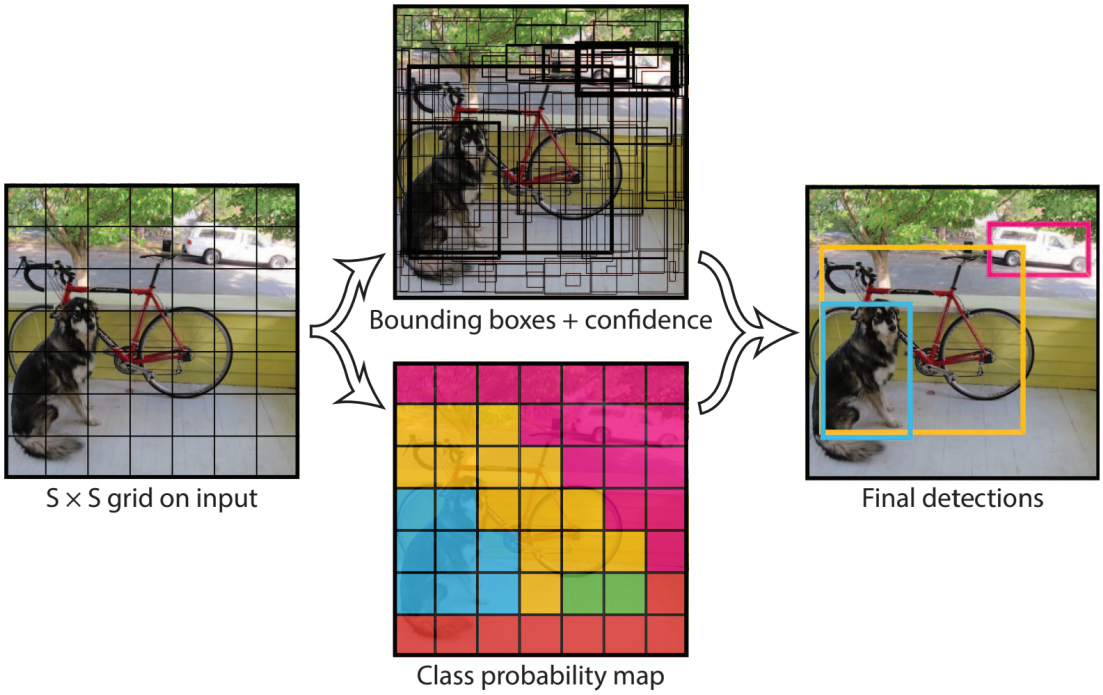

Fig. 2. The YOLO Model. It divides the image into an S × S grid and for each grid cell predicts bounding boxes, confidence for those boxes, and C class probabilities [8].

frame, from the background modelling phase, bounding box (location) along with the class probabilities (appearance) is obtained. Then, for each video V, using the output from this stage, a set C_{XY} consisting of the center (c^t_{xi}, c^t_{yi}) of each object i detected at time instance t and a set L_{XY} consisting of the corresponding width and height (w^t_i, h^t_i) are built. In this paper, only a few classes corresponding to vehicles such as cars, trucks, etc., are considered and the rest of the bounding boxes are not considered.

Figure 3, represents the entire preprocessing pipeline. The input of this stage are the video frames, and the output are the centers (set C_{XY}) and height and width (set L_{XY}) of the bounding boxes of the detected objects.

2.2 Candidate Selection

In this section, the second pipeline is discussed. It consists of two stages, outlier detection, where using a nearest neighbor approach, the misclassified vehicles are removed, and then hotspot detection, where K-means clustering is used to locate the regions where a potential anomaly might have occurred.

2.2.1 ***Outlier Detection***. There are chances that some slow-moving vehicles are not removed by the background model and are detected by the detection algorithm. Since a low confidence threshold of 0.1 is set for the YOLO algorithm, there are chances that the algorithm misclassifies objects in the background like traffic signals and road signs. And, it is not possible to remove such misclassifications and slow-moving objects using a single frame. Therefore, the bounding boxes of all objects detected across the various frames in a video are combined. The nearest neighbor approach is

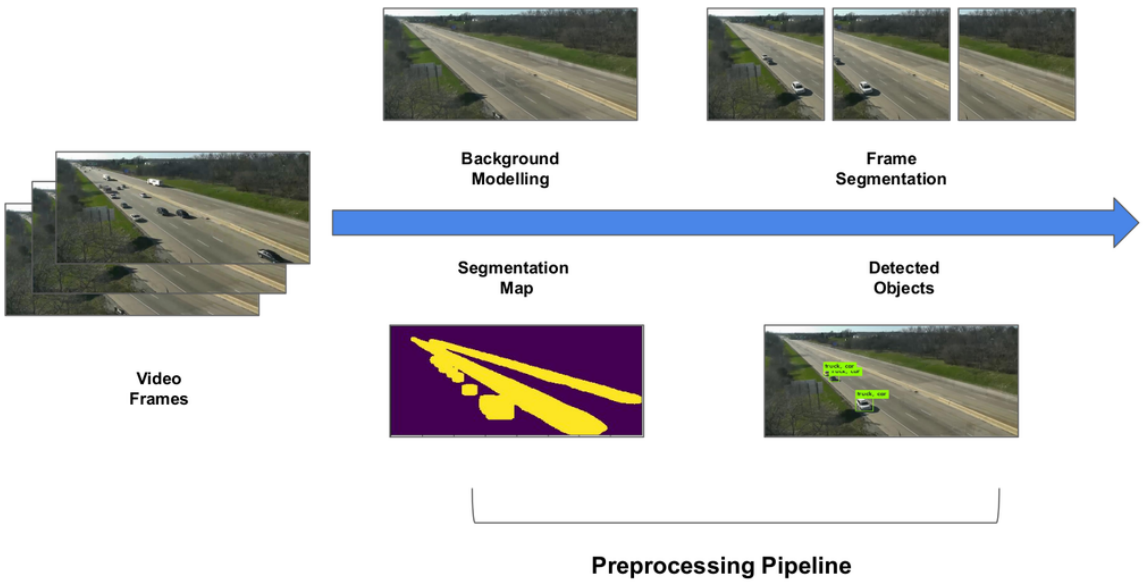

Fig. 3. The preprocessing stage of the proposed method [2].

employed for this purpose. K nearest neighbor is a simple algorithm that classifies any new data point based on how it's neighboring points are classified [5]. The centers (c^t_{xi}, c^t_{yi}) of the bounding box for an object i detected at each time instance t, obtained from the previous pipeline, are mapped to a 2-dimensional plane. The logic behind it being the objects in the background that are misclassified tend to occur frequently at the same location throughout the video and form a cluster. On the other hand, slow-moving objects do not occur frequently at the same location. Then for each point (c^t_{xi}, c^t_{yi}), the k nearest neighbor(kNN) distance $d^t_{xi,yi}(k)$ to its neighboring points is computed and thresholds $l1$ and $l2$ are set. Specifically, a point (c^t_{xi}, c^t_{yi}) is considered as misclassified if

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

and as a slow moving vehicle if

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

2.2.2 ***Hotspot Detection***. In the hotspot detection stage, K-means clustering algorithm is employed to localize potential "hot spots", i.e., locate regions where the stationary objects were detected. This step provides K centroids $(m_1, n_1), ..., (m_K, n_K)$ or potential spatial locations in the video where an anomaly might have occurred. K- means clustering is an unsupervised learning method that aims to partition n observations into K clusters in which each observation belongs to the cluster with the nearest mean or cluster centroid [5].

The value of K is selected using the elbow method. The method consists of plotting the explained variation against the number of clusters and picking the elbow of the curve as the number of clusters to use [5]. Here, in this case, the average within-cluster sum of squares is used as the metric. And then once, the centroids or potential locations are obtained, the segmentation map from subsection 2.1 is used to verify whether the centroids lie in the region of interest and if not then that centroid is not considered. Finally, the centers of the detected object i at each time instance t are used, to iteratively, look for the first-time instance $t_{K\alpha}$ where an object is detected at each of the K locations (centroids) or potential regions of interest. Since the objects are detected every 100 frames and due to the delay caused by a small α as mentioned in subsection 2.1, a backtracking algorithm is used, which computes and monitors a similarity score to find the true onset time of anomaly, which is discussed in the subsection 2.3.

Figure 4, summarizes the candidate selection stage of the proposed method. First, using the KNN approach, the slow-moving objects and the misclassified objects are identified. Next, the K-means clustering and segmentation map are used to find the potential anomalous regions. And three regions of interest or potential anomalous regions are identified. In this module, the input is the set C_{XY} consisting of the centers of the objects detected in the F^t_{avg} frames, and the segmentation map (S) and the output are the K centroids representing the location of the potential anomaly.

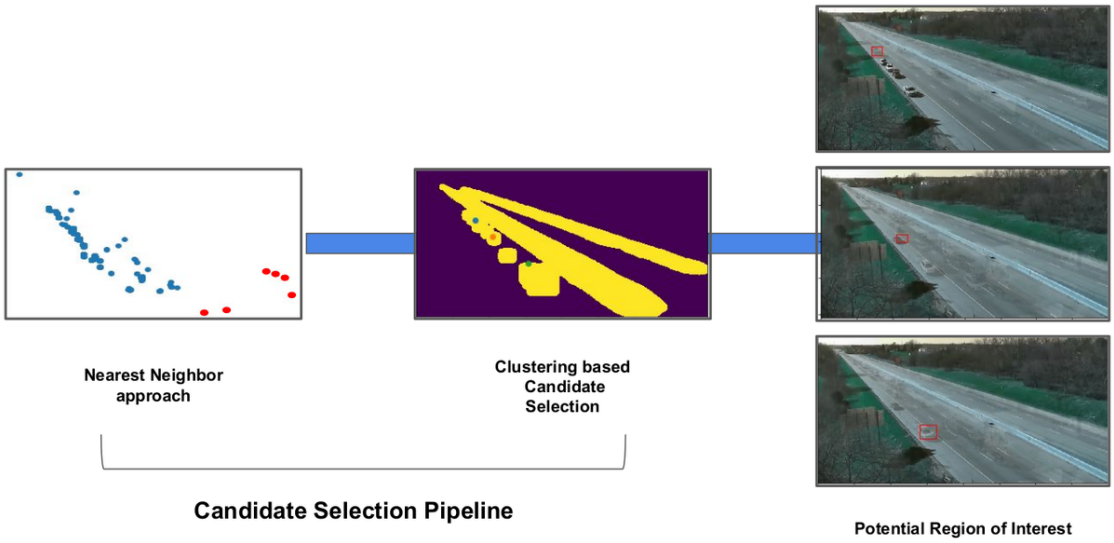

Fig. 4. The candidate selection stage of the proposed method. The blue dots represent objects of interest whereas the red dots represent misclassified objects [2].

2.3 Backtracking Anomaly Detection

This section talks about the last stage of the proposed method i.e., backtracking anomaly detection. Given the potential anomaly onset time $t_{K\alpha}$ for K centroids from the last subsection 2.2 and region of interest (wti , hti) (ie., the width and height of the bounding box, obtained from the object detection stage) extracted from the set L_{XY}, the structural similarity (SSim) index is computed, 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 $t_{K\alpha}$. SSim index is used for measuring the similarity between two images. SSim is computed based on three factors - luminance, contrast, and structure [11]. Ideally, if there is no stalled vehicle at the location, the

structural similarity value should be very low and almost close to zero. And the value dramatically increases as soon as a stalled vehicle appears in the frame. A threshold for the SSim index value is set to detect this change. Specifically, the focus is on whether the increase is persistent over several frames or occurs only over a couple of frames. Finally, the onset time of the anomaly is declared when the index exceeds the specified threshold.

Figure 5, summarizes the backtracking anomaly detection module of the method. In this stage, the input is the K centroids and the set L_{XY} consisting of the width and height of the region of interest. The output is the true anomaly time onset.

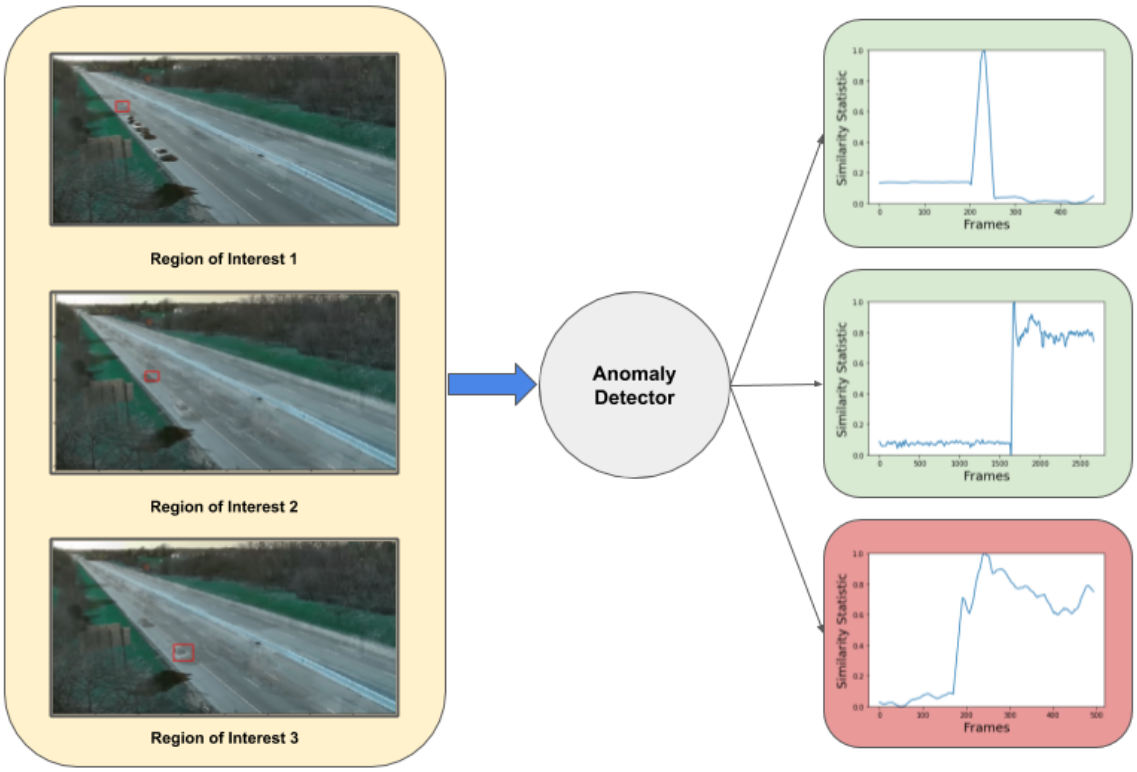

Fig. 5. Backtracking anomaly detection pipeline for the proposed framework. The structural similarity is monitored for each region of interest. The first case is disregarded as the increase in statistic is not persistent. In the second and third region of interest, an anomaly occurs but since it is earlier in the third case, it is considered as anomalous [2].

3 EXPERIMENT

This section talks about the dataset used and the evaluation criteria. The Track 4 training and testing set of the NVIDIA AI CITY 2020 challenge consisted 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 constituted of stalled vehicles and crashes.

Table 1. Performance of the Proposed Model

F1 Score	RMSE	S_4 Score
0.5926	8.2386	0.5763

The evaluation for the competition had two major criteria, namely detection delay measured by the root mean square error (RMSE) and the detection performance measured by the $F1$ score. The final statistic was termed as S_4 and was computed using the formula mentioned below:

$$S_4 = F1 * (1 - NRMSE) \tag{4}$$

The range of scores was from 0 to 1, with 1 signifying the best performance that could be achieved. Table 1 summarizes the figures obtained by the model.

4 IMPROVEMENTS

Even though the proposed method in the paper used no external data and the framework had zero training computational overhead, it could be improved further. First, though YOLO is very fast and processes images in real-time at 45 frames per second. It still lags behind some state of art detection systems in accuracy. It struggles in detecting small objects. Also, the loss function of the model treats the error in a small and large bounding box equally. A small error in a large box can be ignored but a small error in a small box has a much greater impact on IOU. So instead of YOLO, Faster-RCNN or a combination of Faster-RCNN and YOLO could be used. Faster-RCNN is two-stage object detection algorithm that uses a region proposal network first to generate regions of interest and then do object classification and bounding box regression. This method is typically slower and take considerably longer but are much better at detecting small objects. As per the experiment performed in [8] the mean average precision (mAP), when Faster-RCNN and YOLO are combined, is more than YOLO and Faster-RCNN used individually.

The MOG2 (Mixture-of-Gaussians) technique for background modelling has proven to be more stable than the moving average technique as per [6]. The idea behind MOG2 is that it assumes that different distributions represent each different background and foreground colors. The most probable background colors would remain longer and more static [14]. Also, in this paper, the background modelling is performed only in one direction, but the background modelling could also be done from both directions and one could be utilized to predict the candidate anomalies and the other one could be designed to refine the start time of abnormal traffic events precisely. The later procedure would provide a more precise anomaly onset time [6][13].

Since in the preprocessing module in subsection 2.1 segmentation map is used to focus only on the highways where cars could be found and all surrounding regions such parking lots or farms next to the highways are ignored and not considered as part of the area of interest. So, in case after an accident car goes sideways and drives off the road, and is no more part of the area of interest in the segmentation map, the model might fail to consider this scenario an anomaly. So, maybe a trajectory-based feature can also be included in the model, to detect such anomalies as in [3].

5 SUMMARY

This report briefs the proposed model in Doshi et al.[2], which is an unsupervised anomaly detection system for traffic videos. The model ranked second in the NVIDIA AI CITY 2020 challenge, and significantly reduces the testing computational overhead and completely removes the training overhead. The model focuses on the anomalies related to stationary objects and comprises three modules namely - preprocessing, candidate selection, and backtracking anomaly detection.

The preprocessing module focuses on detecting the stationary objects in the video. In the first stage i.e., background modelling, the moving average technique is used to filter out the moving objects in the video and obtain averaged frames containing the stationary objects i.e, vehicles such as cars and trucks. In this stage, only 1 out of 100 frames are considered. Next is the road segmentation stage where a segmentation map is built to focus on the area of interest which is the highway in this case assuming that most of the anomalies would occur on the roads. The final stage is object detection, where the algorithm YOLO is used to detect stationary objects. In the second module, candidate selection, the algorithm KNN and K-means clustering algorithms are used to remove the misclassified objects detected by the detection algorithm and to obtain the location of the stationary object or the potential anomaly respectively. The backtracking anomaly detection pipeline computes the structural similarity index to obtain the true onset time of the anomaly.

The model achieved an $F1$ score of 0.5926 along with 8.2386 root mean square error($RMSE$). Although the model is very fast and significantly reduces the computation overhead, there are a few aspects in which it could be improved like, combining Faster-RCNN and YOLO for object detection instead of just YOLO and employing the MOG2 for background modelling rather than the moving average technique.

REFERENCES

[1] Shuai Bai, Zhiqun He, Yu Lei, Wei Wu, Chengkai Zhu, Ming Sun, and Junjie Yan. 2019. Traffic Anomaly Detection via Perspective Map based on Spatial-temporal Information Matrix. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[2] Keval Doshi and Yasin Yilmaz. 2020. Fast Unsupervised Anomaly Detection in Traffic Videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2658–2664. https://doi.org/10.1109/CVPRW50498.2020.00320

[3] Zhouyu Fu, Weiming Hu, and Tieniu Tan. 2005. Similarity based vehicle trajectory clustering and anomaly detection. In *IEEE International Conference on Image Processing 2005*, Vol. 2. II–602. https://doi.org/10.1109/ICIP.2005.1530127

[4] D. M. Hawkins. 1980. *Identification of outliers*. Chapman and Hall, London [u.a.].

[5] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2014. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.

[6] Yingying Li, Jie Wu, Xue Bai, Xipeng Yang, Xiao Tan, Guanbin Li, Shilei Wen, Hongwu Zhang, and Errui Ding. 2020. Multi-Granularity Tracking with Modularlized Components for Unsupervised Vehicles Anomaly Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2501–2510. https://doi.org/10.1109/CVPRW50498.2020.00301

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs.CV]

[8] Joseph Redmon, Santosh Divvala, Ross B. Girshick, and Ali Farhadi. 2015. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788. https://doi.org/10.1109/CVPR.2016.91

[9] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. https://doi.org/arXiv:1804.02767

[10] Vipin Kumar Varun Chandola, Arindam Banerjee. 2009. Anomaly detection: A survey. (2009). https://doi.org/10.1145/1541880.1541882

[11] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

[12] Zheng Yi and Fan Liangzhong. 2010. Moving object detection based on running average background and temporal difference. In *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*. 270–272. https://doi.org/10.1109/ISKE.2010.5680866

[13] Yuxiang Zhao, Wenhao Wu, Yue He, Yingying Li, Xiao Tan, and Shifeng Chen. 2021. Good Practices and A Strong Baseline for Traffic Anomaly Detection. *CoRR* abs/2105.03827 (2021). arXiv:2105.03827 https://arxiv.org/abs/2105.03827

[14] Z. Zivkovic. 2004. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 2. 28–31 Vol.2. https://doi.org/10.1109/ICPR.2004.1333992