The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2022

Volume 19, Pages 15-20

IConTech 2022: International Conference on Technology

A Real-Time Video Surveillance for Rule Violation Detection on High Ways

Ibrahim DELIBASOGLU
Sakarya University

Abstract: Processing of images obtained with cameras is important for monitoring and/or securing highways. Early warning is important for highway safety in cases such as accident, traffic rule violation, automatic detection of people or animals entering the road. For this purpose, besides the detection of objects such as vehicles and people, other moving targets or objects falling on the road, such as boards, etc., must be detected. For this, besides object recognition, motion detection and object tracking are also needed. The purpose of motion detection is to classify pixels in an image as background and foreground. Within the scope of this study, deep learning-based machine learning method was used for the detection of vehicles and people entering the highway, and classical image processing methods were preferred for the detection of any moving object other than these. In this context, a software infrastructure has been developed in which different processes can work together and share data with each other. The developed software was tested on the Nvidia Jetson development card, and algorithm performance and speed tests were carried out on the selected GPU supported Nvidia Jetson card.

Keywords: Traffic surveillance, Highway security, Vehicle detection, Vehicle tracking

Introduction

Instant processing of images obtained with cameras is important for monitoring and/or securing highways. It is important for highway safety to create an early warning in cases such as accident, rule violation, automatic detection of a person or animal entering the road. For this purpose, besides the detection of objects such as vehicles and people, other moving creatures or objects such as boards falling on the road must be detected. For this, besides object recognition, object tracking and motion detection are also needed. The purpose of motion detection is to classify pixels in an image as background and foreground. The classification to be made is complicated according to the dynamic changes of the background such as the motion state of the camera, the ambient lighting, the movement of the trees in the image that are not defined as motion in the wind, and the ripple in the sea. The cameras from which the images are taken can be static or moving. In static cameras, the center of motion is fixed and the camera does not rotate. Therefore, it is easier to detect background fixed and moving objects. Moving cameras, on the other hand, are divided into cameras that move freely and rotate around a fixed center. Moving cameras with a fixed camera center can perform rotations called pan, tilt and zoom. The most well-known of these types of cameras are PTZ cameras. Drones/UAVs are the best examples of free-motion cameras. Considering that the cameras used in traffic monitoring will be PTZ cameras that can move at certain intervals in a way that changes angle, camera movement should be detected at least to prevent false detections when the camera moves.

When the object detection studies in the literature are examined, it is seen that the best results are obtained with Convolutional Neural Networks (CNN) based methods. In these artificial neural network (ANN) based methods, with the increase in the number of layers to be trained in the proposed architectures, deeper neural networks have been designed and the name "deep learning" has been used. In this context, different architectures have been proposed for the problem of detecting certain objects in an image, that is, finding the location of an object in the image and the class it belongs to. Unlike object classification, this problem also includes finding the re-

- Selection and peer-review under responsibility of the Organizing Committee of the Conference

gion containing the object. The proposed architectures can be grouped under two main categories. In regional-based CNN architectures proposed as the first type, a neural network first selects the regions that may belong to an object, and then another neural network performs the classification process. R-CNN (Girshick et al., 2014). Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren et al., 2015) are deep learning architectures for this category. YOLO (Redmond et al., 2016) and SSD (Liu et al., 2016) architectures can be given as examples of the second type. In this type of architecture, detection and classification processes are performed together, so it is more advantageous in terms of speed. Within the scope of this study, it is considered to use one of the current object detection architectures developed based on YOLOV5 (https://github.com/ultralytics/yolov5), considering speed and performance.

Studies on motion detection in the literature can basically be grouped under three categories; subtraction of consecutive frames, background modeling and optical flow methods. Although the first and simplest method, the subtraction of consecutive frames, can adapt to dynamic environments, its performance is extremely low, especially in moving cameras and moving backgrounds. In background modeling based methods, a model is constructed according to a certain number of previous frames, and pixels that differ from the background model are classified as moving pixels. For background modeling, classical image processing based methods (Gianni et al., 2016), statistical models (Sriram et al., 2013; Kwang et al., 2018) and neural networks (Massimo & Maurizio, 2017) have been used in studies in the literature. In optical flow-based methods, optical flow vectors obtained with deep learning architectures are used. There are studies on detecting moving pixels with the operations applied on the estimated flow vectors for each pixel and adaptive threshold value approaches (Junjie et al., 2018). FlowNet2 architecture is widely used in optical flow prediction (Ilg et al., 2017). Although good performances can be achieved in deep learning-based optical flow calculation, it is not suitable for use in real-time systems because it requires excessive processing power in high-resolution images. Therefore, considering the computational cost-performance balance, statistical methods based on background modeling can be considered as the most reasonable method in motion detection problem at present.

In this paper, we track the motion regions and detect object bounding boxes to determine some traffic rules violation. The details of the object detection, motion detection and tracking algorithm used in our method are given in Section 2.

Proposed Method

Object Detection

In this study, a training set was created from the images selected from the VisDrone (Zhu et al., 2021) dataset. The images taken from very vertical angle and including occluded objects were not selected in our dataset. Sample images from the data set are shown in Figure 2. The dataset, which is a subset of this Visdrone, was used to train a convolutional neural network model to detect vehicles and pedestrians on highways.

Figure 1.   Sample images from VisDrone dataset

While selecting the deep learning model, the model inference speed and accuracy criteria were taken as a basis, and I have decided to use the YOLOV5 architecture. Firstly, I have coded a conversion script to use the object labels of the VisDrone dataset in YOLOV5 model training. The loss function (loss) and mAP (mean Average Precision) metric values obtained during the training process are shown in Figure 2.

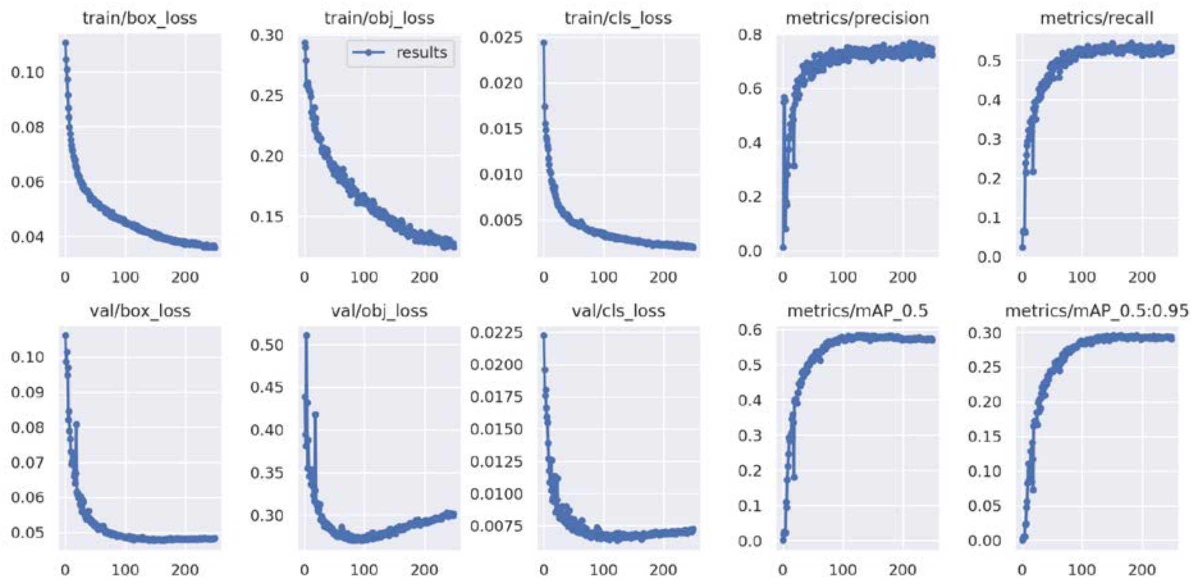

Figure 2. Training graphics

Motion Detection

Motion detection algorithm is implemented using C++ and CUDA. Background modeling-based method was widely used to solve motion pixels in the literature (Kim et al., 2013; Yu et al., 2019). In this work, we have developed motion detection algorithm based on BSDOF (Delibasoglu., 2021) method. In order to increase the speed of the algorithm for a better performance over Nvidia Jetson card, some minor changes were made to the BSDOF algorithm, and the background model was created on gray images instead of three band (RGB). While calculating the Homography matrix to detect camera movement, the CUDA implementation of the Lucas Kanade[4] algorithm, which is implemented in OpenCV, was used. The images used for testing and the sample motion mask images obtained are shown in Figure 3.

Figure 3. Sample motion mask and detected motion regions

Tracking and Producing Alarms

In the last step, motion detection and object detection results are merged and tracking is applied to produce the alarms. For this, firstly bounding boxes are obtained from the motion mask by extracting contours from binary masks and applying a region growing to merge so close bounding boxes. We compare the motion boxes (B_m) with the detection boxes (B_d), and eliminate the motion boxes intersected with the detection boxes. Thus, we only use the motion boxes, which could not be detected by an object detector, in the tracking stage. We store the tracked objects in a list, called $L_{tracker}$. In each frame, we compare the B_m *and* B_d *with* $L_{tracker}$ and update tracked object positions in the $L_{tracker}$. We control whether the intersection of union (IOU) is greater than T_{iou}=0.5 to match the boxes.

$$\frac{B_m \cap B_d}{B_m \cup B_d} > T_{iou} \qquad (1)$$

There is set a *hit counter* and *miss counter* for each tracked object. The targets are deleted if they are missed Tmiss times. Then, the movement of each object in the last *n* frames is calculated, and a distance threshold (T_{dist}) is used to decide whether a target is moving or not. T_{dist} is dynamically assigned to be the maximum of the width and height of the box. Thus, it is determined whether each target is moving or stationary. In the last step of the tracking, we check some specific situations to produce the following alarms:

- If a target classified as a vehicle is stopping for a long time, the system produces an alarm for "Stopping car".
- If a target classified as motion (B_m *detections*) is moving for a while, the system produces an alarm for "Moving target".
- If a target is moving in the opposite direction of the traffic flow (for example: going backwards or trying to turn), the system produces an alarm for "traffic rule violation".

The software developed in this study generates these three alarms. The overview framework of proposed method is shown in Figure 4. However, we would like to emphasize that the task of finding the traffic flow direction, which is extremely important for the final alarm, is not done within the algorithm.

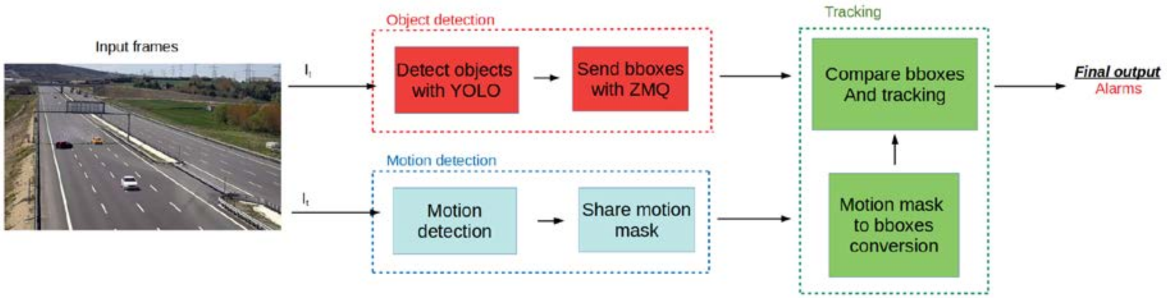

Figure 4. General framework of the proposed method.

Results and Discussion

In this study, a software has been developed to detect traffic rule violations. The developed software has been tested on a PC with an Nvidia graphic card RTX 2070, and an Nvidia Jetson NX development card. The object detection process is implemented with Python, while the motion detection and tracking parts are implemented in C++. The algorithm mostly uses the GPU to speed up the process for better performance, especially on the Jetson development board. The working speed of each step in the algorithm is given in Table 1.

Table 1. The speed of the modules used in the proposed method

Algorithm steps	Nvidia Jetson NX development board (FPS)	Nvidia RTX 2070 (FPS)
Motion detection	~14	~70
Object detection	~14	~80
Tracking	~50	~166

The "*motion detection*" and "*object detection*" algorithms/methods run in different processes in the implemented software architecture. Thus, the total speed of the proposed method is increased. The motion and object detection results are merged in the "*Tracking*" process, in the last step. Figure 5. shows the sample outputs for the three different alarms. The left columns show the produced alarms with the red bounding box, while the right column represents the object detection and motion detection outputs. The green bounding boxes represent the object detector result (with the object name) and the blue rectangles represent the boxes extracted from the motion mask. In the first row of the figure, two vehicles that are stationary for a while are detected. In the second row, the vehicle moving the opposite of the traffic flow is detected and the alarm is produced. While the camera is zooming in the second image, the blue boxes are also found as a motion region. These are eliminated with the help of the tracking algorithm, so that false alarms are not produced. In the last image, a vehicle trying to stop and turn back is detected.

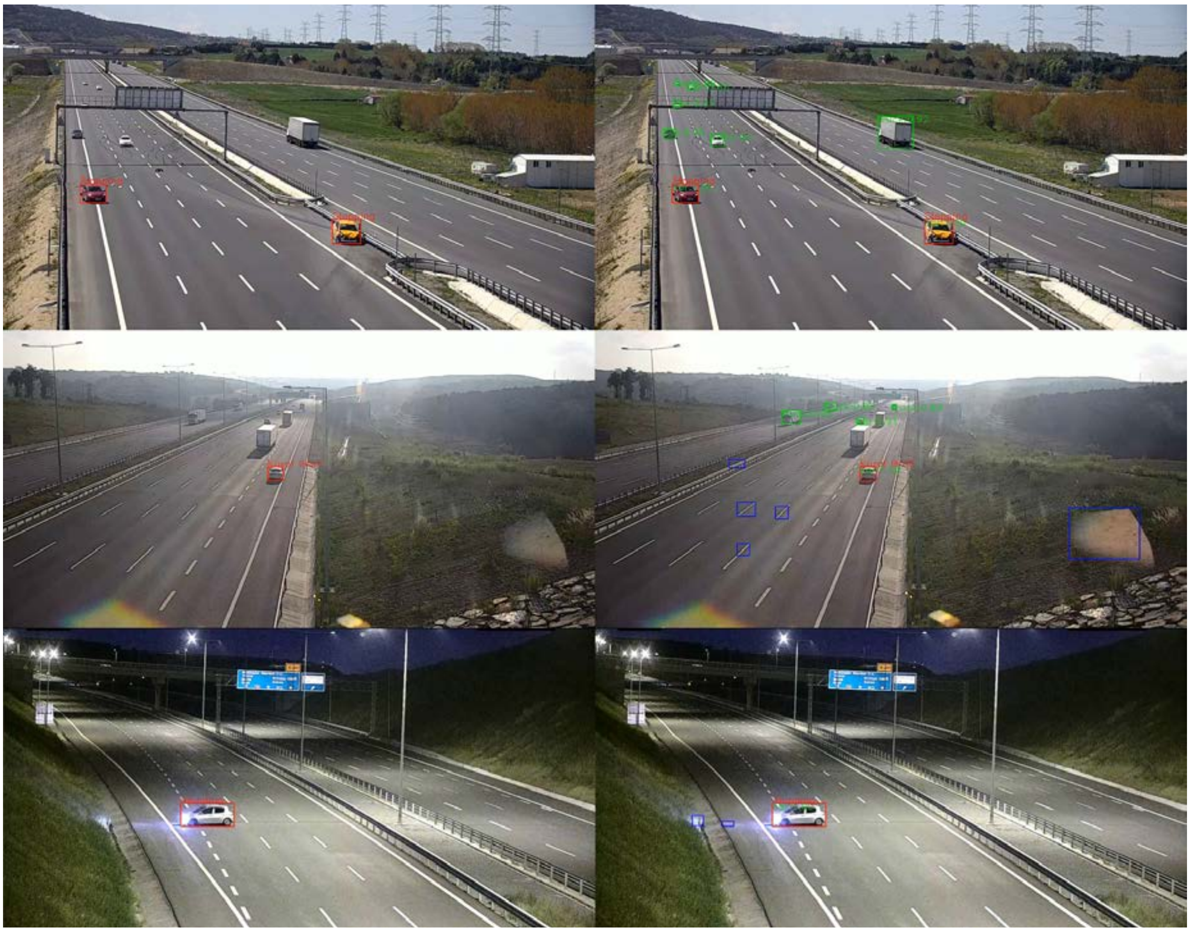

Figure 5. Sample detections and produced alarms

Conclusion

This article proposes a motion detection method that can work with a reasonable speed for high-dimensional videos in a Jetson NX edge device. The motion detection and object detection results are merged with a tracking approach to produce the alarms for traffic rule violation. Although the proposed tracking method is simple, it is sufficient to follow vehicles on not very busy roads. The motion detection algorithm, on the other hand, is very suitable for detecting any moving target (such as an animal) that cannot be detected by object recognition.

In future work, the tracking method could be improved to distinguish vehicles that are very close to each other to prevent the ID changes. Thus, the method could run in more heavy traffic. However, it is also important that deep learning-based ID assigning methods may run so slow for many targets and it results a lower FPS specially for edge device.

Scientific Ethics Declaration

The author declares that he is solely responsible for the scientific, ethical, and legal aspects of the paper published in EPSTEM.

Acknowledgements or Notes

* This article was presented as an oral presentation at the International Conference on Technology (www.icontechno.net) held in Antalya/Turkey on November 16-19, 2022.

*This research was supported by Sakarya University Scientific Research Projects Unit under the Project No: 2021-9-32-117 and Title: “Traffic safety and rule violation detection on highways”.

References

Allebosch,G., Deboeverie,F.,Veelaert,P., & Philips,W. (2015). EFIC: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints. *In International Conference on Advanced Concepts for Intelligent Vision Systems,* (pp. 130–141). Springer.

Delibasoglu, I. (2021). Real-time motion detection with candidate masks and region growing for moving cameras. *Journal of Electronic Imaging, 30*(6), 063027.

Girshick, R. (2015). Fast r-cnn. *Proceedings of the IEEE international conference on computer vision,* 1440–1448.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition,*580–587.

Github, *Yolov 5*. Retrived from: https://github.com/ultralytics/yolov5

Gregorio, M.D., & Giordano, M. (2017). Wisardrp for change detection in video sequences. *25 th Europian Sympousium on Artifical Neutral Networks, Computational Intelligence and Machine Leraning(ESANN)*,453-458.

Huang, J., Zou, W., Zhu, J., Z., & Zhu, Z. (2018). Optical flow based real-time moving object detection in unconstrained scenes. *arXiv:*1807.04890,1.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2462-2470.

Kim, S. W., Yun, K., Yi, K. M., Kim, S. J., & Choi, J. Y. (2012). Detection of moving objects with a moving camera using non-panoramic background model. *Machine Vision and Applications*, *24*(5), 1-14.

Kim, S. W., Yun, K., Yi, K. M., Kim, S. J., & Choi, J. Y. (2013). Detection of moving objects with non-stationary cameras in 5.8 ms: Bringing motion detection to your mobile device. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*s, 27–34.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. *In European conference on computer vision,* (pp:21–37). Cham: Springer.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition,* 779–788.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv:*1506.01497.

Varadarajan,S., Miller,P., & Zhou, H. (2013). Spatial mixture of gaussians for dynamic background modeling. *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 63–68. .

Yu, Y., Kurnianggoro,L., & Jo, K.-H. (2019). Moving object detection for a moving camera based on global motion compensation and adaptive background model. *International Journal of Control, Automation and Systems, 17*(7), 1866-1874.

Zhu P., Wen L., Du D., Bian, X., Fan, H., Hu, Q., Ling,H. (2021). Detection and Tracking Meet Drones Challenge. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *44*(11),7380-7399.

Author Information

Ibrahim Delibasoglu
Sakarya University
Sakarya /Turkey
Contact E-mail:*ibrahimdelibasoglu@sakarya.edu.tr*

To cite this article:

Delibasoglu, I. (2022). A real-time video surveillance for rule violation detection on highways. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 19,* 15-20.