Anomalous trajectory detection for automated traffic video surveillance

Jose D. Fernández[0000−0003−3702−2230], Jorge García-González[0000−0001−8610−3462], Rafaela Benítez-Rochel[0000−0001−8500−0488], Miguel A. Molina-Cabello[0000−0002−8929−6017], and Ezequiel López-Rubio[0000−0001−8231−5687]

[1] Department of Computer Languages and Computer Science
University of Málaga, Málaga, Spain
[2] Instituto de Investigación Biomédica de Málaga – IBIMA, Málaga, Spain
josedavid@uma.es ; {jorgegarcia,benitez,miguelangel,ezeqlr}@lcc.uma.es

Abstract. Vehicle trajectories extracted from traffic video sequences can be helpful for many purposes. In particular, the analysis of detected anomalous trajectories may enhance drivers' safety. This work proposes a methodology to detect anomalous vehicle trajectories by using a vehicle detection, a vehicle tracking and a processing of the tracking information steps. Once trajectories are detected, their velocity vectors are estimated and an anomaly value is computed for each trajectory by comparing its vector with those from its nearest neighbours. The management of these anomaly values allows considering which trajectories are suitable to be potentially anomalous considered. Real and synthetic videos have been included in the experiments to perform the goodness of the proposal.

Keywords: Anomaly detection · Object tracking · Object detection · Video surveillance · Deep learning.

1 Introduction

Anomaly detection is an important part of understanding the activities of video surveillance systems [1]. The use of video in computer vision is an important and current research topic that includes autonomous surveillance systems. These systems follow three basic steps: detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to detect unexpected situations.

For moving object detection, the latter existing state-of-the-art methods are based on Deep Learning (DL) [2]. This kind of methods have become a powerful technique for object detection due to it is significantly more robust to occlusion, complex scenes, and challenging illumination and, on the other hand, because of the availability of large video data sets and DL frameworks. In fact, the progress of object detection is usually separated into two historical periods (before and after the introduction of DL).

Regarding the object tracking, a single method cannot provide good accuracy for different kinds of videos with different situations. The tracking methods can be divided into three categories based on the use of object representations, namely, methods establishing point correspondence, methods using primitive geometric models, and methods using contour evolution. In this work, we have used a method within the first category, that is because we have found that the selected method is very appropriate to the characteristics of the analyzed traffic camera videos and it makes the process of identifying anomalous trajectories easier.

Therefore, the aim of this work is to detect anomalies in vehicle traffic. A deep neural network is used to detect vehicles, then a specific method is described in order to track the vehicles, and finally, a technique to detect anomalous trajectories of vehicles is introduced.

Detection of vehicles that appear in traffic video sequences has been addressed in the literature in order to carry out different tasks. Moreover, the trajectories of the vehicles are determined along the way, so it may be necessary to deal with the tracking of vehicles [3].

Vehicle classification on-road traffic is an important task due to its great potential in traffic management. Different techniques have been used to classify vehicles; an example of a traditional proposal can be found in [4], where it uses a foreground object detection method and a feature extractor to obtain the most significant features of the detected vehicles into several categories such as car, motorcycle, truck or van. However, more recent works [5, 6] can do that task by using a neural system based on a Convolutional Neural Network (CNN) architecture, which is a DL network.

Another interesting task related to vehicle detection consists of deriving traffic information. In [7] YOLOv4 algorithm for vehicle detection is used and then the lane-by-lane vehicle trajectories using the detected locations of vehicles are estimated. Based on the estimated vehicle trajectories, the traffic volumes of each lane-by-lane traveling directions and queue lengths of each lane are estimated by matching vehicle locations with HD map.

Latest advances in research related to the field of traffic video surveillance have made possible the study of other relevant problems whose solution arises from the analysis and detection of vehicles along the road. This is the case of the detection of pollution levels of transport vehicles problem which has been addressed in [8, 9]. In this works the object detection and classification process is based on a pre-trained Faster-RCNN model [10]. With that recognition and vehicle tracking, the system predicts the pollution of the selected area in real-time. The model which estimates the pollution is based on the frequency of vehicles and their speed.

The rest of the paper is organized as follows. Section 2 describes in detail the methodology used to solve our problem. Section 3 shows several experimental results over several well-known public traffic surveillance sequences. Finally, Section 4 outlines the conclusions of the paper.

2 Methodology

To detect anomalous trajectories in video feeds from traffic cameras, we propose a system architecture with the following stages:

1. Vehicle detection: a deep neural network is used to detect vehicles in each image frame of the video.
2. Vehicle tracking: we then keep track of vehicles across multiple frames to get their frame-by-frame positions.
3. Processing of tracking information: for each vehicle, its tracked frame-by-frame positions are processed to highlight possible anomalous trajectories.
4. Thresholding: each possible anomalous trajectory is flagged as anomalous if it surpasses any of several possible thresholds.

Next, each stage is described in detail. The first stage of our proposed system employs an object detection deep convolutional network to obtain tentative detections of vehicles. The output of the object detection network for an input image $\mathbf{X}$ is a set of detections S where each detection consists of an axis-aligned bounding box, an object class label, and a confidence level:

$$S = \mathcal{F}(\mathbf{X}) \tag{1}$$

$$S = \{(a_i, b_i, c_i, d_i, q_i, r_i) \mid i \in \{1, ..., N\}\} \tag{2}$$

where N is the number of detections, $(a_i, b_i) \in \mathbb{R}^2$ are the coordinates of the upper left corner of the i-th detection within the image $\mathbf{X}$, $(c_i, d_i) \in \mathbb{R}^2$ are the coordinates of the lower right corner of the i-th detection within $\mathbf{X}$, q_i is the class label of the detection, and $r_i \in \mathbb{R}$ is the confidence level of the detection.

We threshold the detections at a minimum confidence level of r_{min}, and disregard the object class after we filter out the non vehicle classes. Therefore a filtered set of detections is obtained as follows:

$$S' = \{(a_i, b_i, c_i, d_i, q_i, r_i) \in S \mid r_i \geq r_{min}, q_i \in V\} \tag{3}$$

where V is the set of vehicle classes.

After that, the filtered set S' is passed on the non-maximal-suppression algorithm in order to obtain a further filtered set of detections S'' for the current video frame which is fed to the next stage.

The second stage is object tracking. We pose the tracking problem as a linear sum assignment between the detections at frames $t+1$ and t contained in the detection sets S''_{t+1} and S''_t, where the cost $C_{ij}$of matching two detections $s_i, s_j \in S''$ is the Euclidean distance between the centers $\boldsymbol{\mu}_i$, $\boldsymbol{\mu}_j$ of their associated bounding boxes:

$$\boldsymbol{\mu}_i = \left(\frac{a_i + c_i}{2}, \frac{b_i + d_i}{2}\right) \qquad \boldsymbol{\mu}_j = \left(\frac{a_j + c_j}{2}, \frac{b_j + d_j}{2}\right) \qquad C_{ij} = \left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\| \tag{4}$$

Our algorithm works well only in ideal conditions: when object movements between frames are relatively small, and objects are consistently detected across all frames where they appear. The traffic camera videos we analyze mostly fulfill the former requirement, but even object detectors with very high mAP scores fail the latter requirement for vehicles with small apparent sizes. To make this algorithm more robust under these conditions, we forbid assignments (i.e. recognizing bounding boxes from frames t and $t+1$ as the same tracked object across both frames) that break one of these heuristics:

- The difference in size between both bounding boxes is larger than a given threshold ratio in any of the two dimensions of the axis-aligned bounding boxes. This avoids false tracking instances between objects of very different sizes, and incidentally makes the system more robust in noisy conditions, when the bounding box estimates fluctuate wildly across frames.
- The displacement between the centers of both bounding boxes is larger than a given threshold. Rather than using an absolute threshold, it is relative to the size of the minimum dimension of the bounding box at $t+1$. This avoids false tracking instances where an object is detected at frame t but not at $t+1$, and a nearby object detected at $t+1$ is erroneously assigned to it. This is most useful for distant objects with small apparent sizes in the image, because object detection networks are more prone to fail to detect small objects. While this heuristic is also useful for objects in the foreground with large apparent sizes, it has to be fine-tuned, however, because fast objects on the foreground can show large displacements that are unlikely for distant objects. To address this issue, we apply two different threshold values, depending on the size of the bounding box relative to the size of the image.

When a vehicle detected in frame t is not assigned a detection in frame $t+1$, we do not keep track of it in case we detect it later.

After this stage, we have vehicle trajectories across the camera's field of view. We posit that we can detect vehicles performing anomalous maneuvers by comparing their trajectories with nearby trajectories. To accomplish this in a simple but effective way, we can measure how different is the velocity of a car with respect to the velocity of its nearest neighbors.

To describe this in more detail, we will introduce some formal notation. Let $\boldsymbol{\mu}_i(t) \in \mathbb{R}^2$ be the position (the center of the bounding box) of vehicle i at frame t. If the tracking stage has determined that the same vehicle was in position $\boldsymbol{\mu}_i(t-1)$ for the previous frame, we define its velocity vector $\mathbf{v}_i(t) \in \mathbb{R}^2$ as follows:

$$\mathbf{v}_i\left(t\right) = \boldsymbol{\mu}_i\left(t\right) - \boldsymbol{\mu}_i\left(t-1\right) \tag{5}$$

Here we subsume the frame rate of the camera as a scale factor. If vehicle i is not tracked both at t and $t+1$, its velocity is undefined. For each vehicle i in the current time step with defined velocity $\mathbf{v}_i(t)$, we consider the set of all detections of other vehicles with defined velocities in the last F frames:

$$D_i\left(t\right) = \left\{j \,:\, \exists \mathbf{v}_j\left(t'\right), i \neq j, t - t' < F\right\} \tag{6}$$

Then we select $D_i^N(t) \subseteq D_i(t)$, the subset of the N nearest detections for vehicle i at time t. Proximity is measured between detections of vehicle i at time t and each vehicle j at time t' as the Euclidean distance between their respective positions $\boldsymbol{\mu}_i(t)$ and $\boldsymbol{\mu}_j(t')$.

After selecting the nearest neighbors $D_i^N(t)$ for each vehicle i, we have to measure how different its velocity vector $\mathbf{v}_i(t)$ is with respect to the velocity vectors of its nearest neighbors $\mathbf{v}_j(t') \in D_i^N(t)$. This can be accomplished by defining the *anomaly value* $A_i(t)$ of vehicle i at frame t as the mean of the moduli of differences among vector velocities:

$$A_i(t) = \text{mean}\left(\|\mathbf{v}_i(t) - \mathbf{v}_j(t')\| \;:\; j \in D_i^N(t)\right) \tag{7}$$

As a measure of anomaly, $A_i(t)$ has some issues. It is somewhat noisy, with large spikes whenever one-off tracking errors happen. We minimize the impact of these errors using a median filter:

$$A_i'(t) = \text{median}\left(A_i(t), A_i(t-1), A_i(t-2)\right) \tag{8}$$

such that $A_i'(t)$ is defined only if the three values $A_i(t)$, $A_i(t-1)$ and $A_i(t-2)$ are defined.

Another difficulty is that $A_i'(t)$ is not dimensionless, i.e., it depends on frame rate and distance to the camera, among other considerations. In order to address this inconvenient, we consider $A_i'(t)$ as *potentially anomalous* if its value is equal or higher than the P_k percentile of all anomaly values measured in the last F frames.

Finally, we declare that a potentially anomalous $A_i'(t)$ value is actually anomalous if either of these conditions are met:

- Its value relative the P_k percentile is larger than a specific ratio s: $A_i'(t) \geq s \cdot P_k$.
- The vehicle keeps a consistent record of potentially anomalous values for a large number of consecutive frames. This is somewhat dependent on the characteristic time and size scales of road vehicles.

3 Experimental Results

In order to evaluate the goodness of the proposed methodology, several experiments have been carried out. The following subsections depict the methods and the datasets we have employed, and the obtained results.

3.1 Methods

OpenCV is used to process video snippets, while yolov5 is the object detection network employed for the first stage of our system. Yolov5 is designed to detect objects in still images, so it is used to detect vehicles in each frame. We configure the network to return only the object detections corresponding to vehicles.

More specifically, the following COCO classes are included in the vehicle classes set V: *car*, *motorcycle*, *bus*, and *truck*. Also, we disregard object class during non-maximal-suppression since this network occasionally detects truck cabins as standalone cars, even if they are within the bounding box of the whole truck. All other configuration parameters are left unmodified, particularly $r_{min} = 0.25$.

SciPy's implementation is used to solve the linear sum assignment for the tracking stage. Everything else is implemented in Python. Regarding the parameters to detect anomalies, we have found that the size of the set $D_i^N(t)$ (the number of neighbors to check if a trajectory is anomalous) should be small to avoid a long warm-up time, during which comparisons will include lots of vehicles in wildly different trajectories. We use a number of nearest neighbors $N = 5$. Regarding the memory size F, i.e., the number of past frames for which we keep track of past vehicle detections, we have found results to be better for very large values. This amounts to not forgetting any vehicle detection for the duration of each video. With respect to the parameters for trajectory processing, we flag as potentially anomalous all detections at or above the P_k percentile.

3.2 Dataset

Several videos taken from different datasets have been considered in the experiments. These videos allow us to analyze the performance of the system under different anomaly conditions, such as vehicle in the opposite direction or risky fast vehicle. Videos without any anomalous vehicle trajectory are also selected. We used videos from three datasets:

- Two videos from a project [11] which deals with the anomalous trajectory detection in traffic videos offers several real and synthetic sequences. The selected sequences are a real video[3] (noted as *Video1*) that shows a vehicle backing onto a busy road and a synthesized video with CARLA [12] (noted as *Video2*) that depicts a car doing counterflow driving.
- Two videos from the Ko-PER Intersection dataset [13]: the sequences *seq. 1a - SK_4* and *seq. 2 - SK_4*, both cases depicting an intersection with traffic lights. The first video presents no anomalies, while the second video shows a vehicle that waits to turn. In order to test the proposed methodology, we have considered as a potential anomaly that vehicle.
- Two videos from the 2014 CDNET dataset [14]. The sequences *highway* and *streetLight* with no anomalies exhibit a road in a video taken from a camera looking at incoming traffic and from the side, respectively.

3.3 Results

From a quantitative point of view, we have selected several well-known measures in order to test the performance of the proposal. In this work, the spatial accuracy (S) has been considered. This measure provides values in the interval $[0, 1]$,

[3] Clip from 02:10 to 02:31 in this Youtube video: https://youtu.be/BF3WuB-7iPo

Table 1. Performance of the system for sequences *Video1* and *Video2* regarding different values of parameters s scaling factor and P_k percentile. Best results are highlighted in **bold**.

Video1	P_{99}	P_{98}	P_{95}	P_{90}	P_{85}
$s = 3$	0.000	0.000	**1.000**	0.500	0.167
$s = 4$	0.000	0.000	**1.000**	0.500	0.250
$s = 5$	0.000	0.000	**1.000**	0.500	0.250
$s = 6$	0.000	0.000	**1.000**	0.500	0.250

Video2	P_{99}	P_{98}	P_{95}	P_{90}	P_{85}
$s = 3$	**1.000**	**1.000**	0.334	0.077	0.052
$s = 4$	**1.000**	**1.000**	**1.000**	0.200	0.071
$s = 5$	**1.000**	**1.000**	**1.000**	0.500	0.143
$s = 6$	0.000	**1.000**	**1.000**	**1.000**	0.250

Table 2. Performance of the system for tested videos. First row shows the ground truth (GT) which represents the number of anomalous trajectories for each video, while remaining rows depict the performance of the system for the indicated measure.

Measure	*Video1*	*Video2*	*seq. 1a - SK_4*	*seq. 2 - SK_4*	*highway*	*streetLight*
GT	1	1	0	1	0	0
TP	1	1	0	1	0	0
FP	0	0	1	1	1	2
FN	0	0	0	0	0	0

where higher is better, and represents the percentage of hits of the system. True positives or number of hits (TP), false negatives or misses (FN) and false positives or false alarms (FP) are also considered in this work. The spatial accuracy is defined as follows: $S = TP/(TP + FN + FP)$.

First of all, in order to establish the value of parameters P_k percentile of all anomaly values and the ratio s used as thresholds, we have tuned different configurations. With these possible configurations, the system may be more demanding or tolerant with the potential anomalies.

The set of values we have tested are $k = \{85, 90, 95, 98, 99\}$ and $s = \{3, 4, 5, 6\}$ with the sequences *Video1* and *Video2*. The obtained results are shown in Table 1. Due to both videos present only one anomaly, the configuration with fixed values $s = 4$ (or $s = 5$ or $s = 6$) and P_{95} achieves the best performance. This way, we have found that a good strategy to flag a vehicle as actually anomalous is to use either of these heuristics: either its $A_i(t)$ value persists as potentially anomalous for more than a certain number of frames (we have used 60 which corresponding with approximately 2-3 seconds for the videos we are using), or it is $s = 4$ times larger than the P_{95} percentile.

Table 2 summarizes the performance of the system. As can be observed, our proposal perfectly detects the anomalous trajectories which are in the videos. Additionally, it must be highlighted that there is no false negatives. Nevertheless, several false positives are detected.

Going deeper, Figure 1 shows the anomaly values $A_i(t)$ for detected vehicles of a selected video plotted by time and a specific frame from that video. It is interesting to observe how real anomalies are well detected in videos *Video1*, *Video2* and *seq. 2, cam. SK_4*. For example, the anomalous trajectory detected in *Video1* is predicted as that because the vehicle maintains $A_i(t)$ values above the

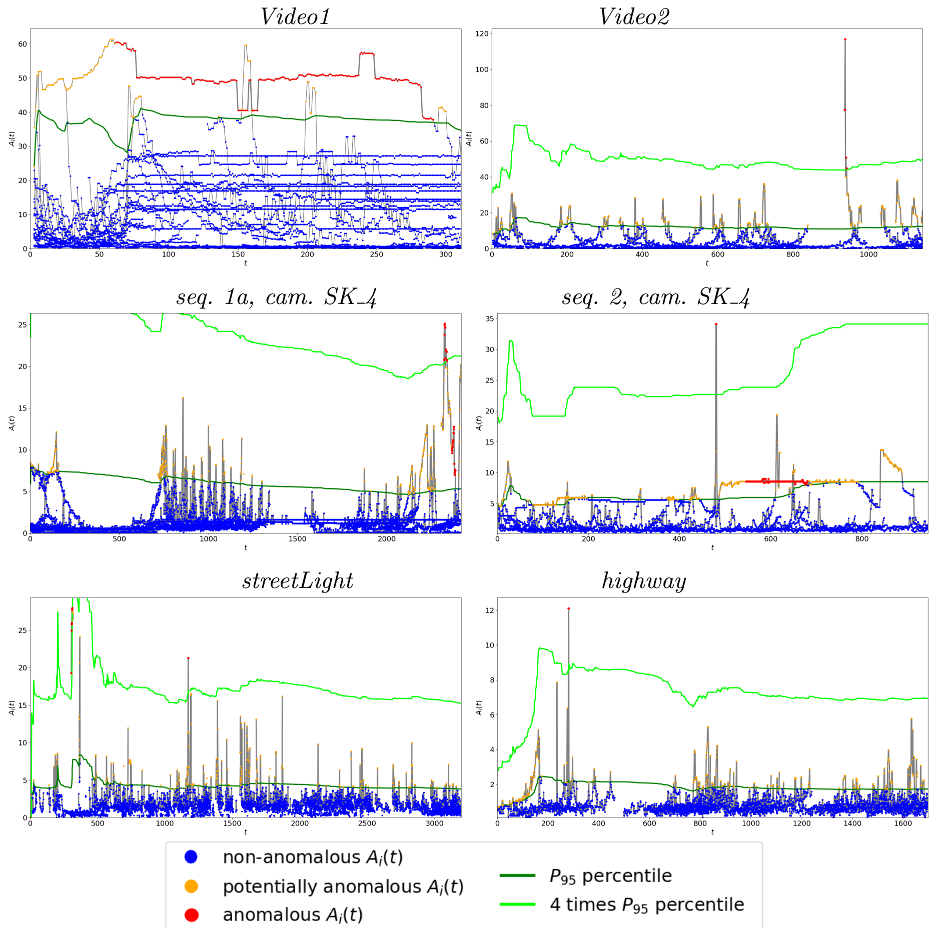

Fig. 1. *Anomaly values* $A_i(t)$ for detected vehicles plotted by time (frame number) in each tested video.

P_{95} percentile for a long time. Regarding false positives anomalous trajectories, they exhibit a low number of frames with an anomalous *anomaly value* $A_i(t)$ and they are related to tracking errors which can be tackled with better detection and tracking subsystems. From the analysis of the frames where false positives appear, we have deduce that they can be categorize in two possible classes: the first vehicle(s) in a given trajectory is liable to be flagged as anomalous; and, when an area (usually an intersection) accumulates lots of different overlapping trajectories, some vehicles will be flagged as anomalous because their trajectories happen to overlap with very different ones.

The detail of an specific frame showing some vehicles flagged by the system is reported in Figure 2. Left image shows a vehicle is backing onto a busy road, center image exhibits a vehicle is driving in opposite direction, and right image

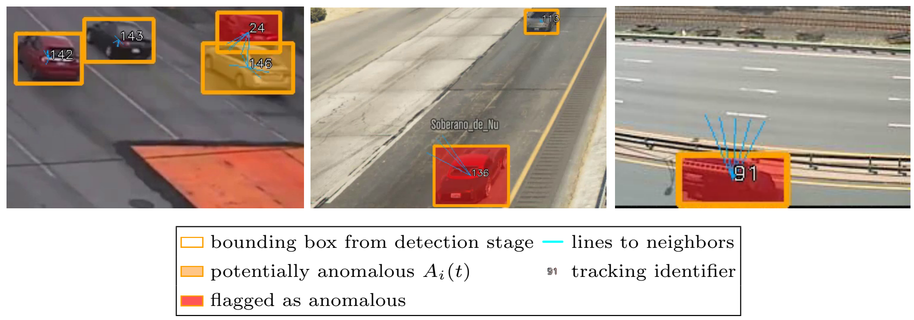

Fig. 2. Predictions made by the proposal for different videos in an specific instant. From left to right: *Video1* (frame 76), *Video2* (frame 938) and *streetLight* (frame 309).

reports a vehicle which is not actually anomalous, but it is the first vehicle to go through its lane, and it gets compared to vehicles from another lane going in the opposite direction. The $N = 5$ nearest neighbors (detections from the current or past frames) used to compute the anomaly value $A_i(t)$ for that tracked vehicle at time t can be observed for each vehicle.

4 Conclusions

The detection of anomalous vehicle trajectories from traffic video sequences has been addressed in this work. From an input traffic video, the proposed methodology detects the vehicles, tracks them to obtain their trajectory, and then estimates which trajectories can be considered anomalous. Given the fact that we are measuring how different the velocity vector of each vehicle is with respect to the velocity vector of the nearest vehicles, we are considering that a vehicle is anomalous if this difference is very high for a given frame or continuously high for a given length of time. We find that this strategy may detect anomalous trajectories consisting of vehicles going the wrong way in a traffic lane or backing next to a busy lane and works especially well when the vehicle is in the foreground or near the foreground. Experiments with real and synthetic traffic videos demonstrate that the approach assesses appropriately different anomalous vehicle behaviours in different scenarios.

Acknowledgments

This work is partially supported by the following Spanish grants RTI2018-094645-B-I00 and UMA18-FEDERJA-084. All of them include funds from the European Regional Development Fund (ERDF). The authors acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. The authors acknowledge the funding from the Instituto de Investigación Biomédica de Málaga – IBIMA and the Universidad de Málaga.

References

1. B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1103–1108.
2. Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
3. A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.
4. M. A. Molina-Cabello, R. M. Luque-Baena, E. López-Rubio, J. M. Ortiz-de Lazcano-Lobato, E. Domínguez, and J. M. Pérez, "Vehicle classification in traffic environments using the growing neural gas," in *International Work-Conference on Artificial Neural Networks*. Springer, 2017, pp. 225–234.
5. M. A. Molina-Cabello, R. M. Luque-Baena, E. López-Rubio, and K. Thurnhofer-Hemsi, "Vehicle type detection by convolutional neural networks," in *International Work-Conference on the Interplay Between Natural and Artificial Computation*. Springer, 2017, pp. 268–278.
6. M. A. Molina-Cabello, R. M. Luque-Baena, E. Lopez-Rubio, and K. Thurnhofer-Hemsi, "Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images," *Integrated Computer-Aided Engineering*, vol. 25, no. 4, pp. 321–333, 2018.
7. J. S. Sehyun Tak, Jong-Deok Lee and S. Kim, "Development of ai-based vehicle detection and tracking system for c-its application," *Journal of Advanced Transportation*, 2021.
8. M. A. Molina-Cabello, R. M. Luque-Baena, E. López-Rubio, L. Deka, and K. Thurnhofer-Hemsi, "Road pollution estimation using static cameras and neural networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.
9. J. García-González, M. A. Molina-Cabello, R. M. Luque-Baena, J. M. Ortiz-de Lazcano-Lobato, and E. López-Rubio, "Road pollution estimation from vehicle tracking in surveillance videos by deep convolutional neural networks," *Applied Soft Computing*, vol. 113, p. 107950, 2021.
10. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.
11. G. C. Leiva, "Videovigilancia de trayectorias anómalas de vehículos en vídeos de tráfico," March 2021.
12. A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.
13. E. Strigel, D. Meissner, F. Seeliger, B. Wilking, and K. Dietmayer, "The ko-per intersection laserscanner and video dataset," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1900–1901.
14. Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 387–394.