Applied Soft Computing 113 (2021) 107950

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Road pollution estimation from vehicle tracking in surveillance videos by deep convolutional neural networks

Jorge García-González *, Miguel A. Molina-Cabello, Rafael M. Luque-Baena, Juan M. Ortiz-de-Lazcano-Lobato, Ezequiel López-Rubio

Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
The Biomedic Research Institute of Málaga (IBIMA), C/ Doctor Miguel Díaz Recio, 28, 29010, Málaga, Spain

ARTICLE INFO

Article history:
Received 17 February 2021
Received in revised form 24 August 2021
Accepted 24 September 2021
Available online 7 October 2021

Keywords:
Traffic air pollution
Object detection
Deep learning
Video surveillance

ABSTRACT

Air quality and reduction of emissions in the transport sector are determinant factors in achieving a sustainable global climate. The monitoring of emissions in traffic routes can help to improve route planning and to design strategies that may make the pollution levels to be reduced. In this work, a method which detects the pollution levels of transport vehicles from the images of IP cameras by means of computer vision techniques and neural networks is proposed. Specifically, for each sequence of images, a homography is calculated to correct the camera perspective and determine the real distance for each pixel. Subsequently, the trajectory of each vehicle is computed by applying convolutional neural networks for object detection and tracking algorithms. Finally, the speed in each frame and the pollution emitted by each vehicle are determined. Experimental results on several datasets available in the literature support the feasibility and scalability of the system as an emission control strategy.

1. Introduction

In recent decades the field of video surveillance has been the subject of intense research. The increase in the number of IP cameras, which are installed mainly for security purposes, has provided a massive amount of data which have made it possible to study not only the detection and tracking of vehicles on the road but also high-level characteristics related to their behavior. Thus, it is possible to detect anomalous patterns that differ from those of the normal behavior for a vehicle or to estimate parameters related to the environment such as the pollution of the area through which the vehicles circulate.

The issue of estimating air pollution caused by vehicle emissions and its effect on the air quality of an area has been approached from different points of view in some densely populated cities [1]. The use of emission monitoring sensors to measure harmful particles produced by traffic [2] has given good results, although it is not very suitable for large areas due to the cost of system installation. At road intersections, hybrid models that combine wavelength-based neural networks and genetic algorithms [3] have been used to determine area pollution.

In order to avoid the cost and difficulties involved in the use of air quality, environmental, and traffic density sensors on which the previous methods are based only static cameras present on the roads will be the source of information for our proposal. The analysis of the movement of vehicles on the roads will allow us to know the speed of each vehicle present in the video sequence. The contribution of each vehicle to the level of pollution in the area will depend on its estimated speed.

The proposed methodology starts with a detection phase of the vehicles appearing in the scene [4]. Traditional techniques applied to foreground object detection such as mixture Gaussian distributions [5] or statistical background modeling [6] have been replaced by deep neural networks, which have been incorporated into the field of video surveillance to address complex tasks such as object recognition [7] and provide much higher success rates in object identification and detection [8]. The recent deep neural network models Faster-RCNN [9] and YOLO [10] have been chosen in this work for that purpose.

Subsequently, a tracking phase is performed [11]. Each vehicle speed is estimated after obtaining the vehicle trajectory along the road. One issue to be taken into account is that the camera perspective may make it difficult the speed estimation. In order to obtain a distance value not distorted by the camera perspective, a

* Corresponding author at: Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain.
E-mail addresses: jorgegarcia@lcc.uma.es (J. García-González), miguelangel@lcc.uma.es (M.A. Molina-Cabello), rmluque@lcc.uma.es (R.M. Luque-Baena), jmortiz@lcc.uma.es (J.M. Ortiz-de-Lazcano-Lobato), ezeqlr@lcc.uma.es (E. López-Rubio).

https://doi.org/10.1016/j.asoc.2021.107950

previously defined homography allows points in the video frame to be projected onto the road plane.

Finally, the estimation of the pollution in the area is based on the number of vehicles that are detected and their corresponding speed.

The remaining of the paper is structured as follows: Section 2 presents related works, Section 3 shows and explains the overall proposal architecture, Section 4 explains the applied methodology, Section 5 outlines the experimental details such as homography generation (Section 5.1), pollution estimation (Section 5.3), resources (Section 5.4), evaluation (Section 5.5) and obtained results (Section 5.6), whereas Section 6 summarizes the conclusions. A final acknowledgments section is included in order to recognize the origin of the funding that allows this research.

2. Related works

The classification of vehicles which appear in typical traffic video sequences has been dealt with different techniques. A more traditional approach was used in [12], where a growing neural gas approach was proposed to classify the vehicles into several categories such as car, motorcycle, truck or van. First of all, the vehicles were detected by using a foreground object detection method. Then, the most significant features of the detected vehicles were obtained by a feature extraction process. And after that, the types of vehicles were determined by using a set of trained Growing Neural Gas (GNG) neural networks.

The same classification task was also addressed in [13,14] where a Convolutional Neural Network (CNN) architecture, namely Alexnet [15], is selected. In both works the deep neural system was particularly trained for vehicle classification into the above-mentioned categories: car, motorcycle, truck or van. Regarding the vehicle tracking system, the trajectories of the vehicles along the way need to be addressed [11]. For this purpose, both approaches were based on a previous work [16].

On the other hand, in the literature there are several proposals to detect or monitor the air pollution level, both from a general point of view and focused on the emissions of vehicles on the road. In [17] the proposed system analyzes the emission of gases from a vehicle, which are acquired through a physical system. The driver is notified in real time in case of high levels of pollution emitted by the vehicle. In [18] machine learning and IoT techniques are applied for the detection and monitoring of vehicle pollution. Each vehicle has a built-in sensor that measures its level of pollution, in addition to determining its location. With machine learning techniques and sensor information obtained from vehicles, it is possible to estimate the pollution generated in a location and warn the most polluting vehicles.

Recent proposals that analyze images in order to study environmental pollution generally include some deep learning technique. In [19] deep convolutional networks are applied for the detection of air pollution by means of images. For the training of the network, pollution measurements obtained in real time are taken through the Beijing Air Quality Observatory.

In [20] deep learning techniques are used on images captured by nearby vehicles or dedicated base stations, with the aim of analyzing vehicle emissions and detecting their pollutant level. Another similar proposal is applied in [21], where a convolutional network in two stages works with video traffic surveillance images. In this case the network is trained to detect the most polluting vehicles based on gases emitted by vehicles. It is important to note that these proposals do not determine the level of pollution of each vehicle in an area or road, but only if the vehicle is highly polluting or not.

A Faster-RCNN pre-trained network [22] was used to recognize the vehicles in a traffic scene [23]. 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 camera acquires the video sequence with a perspective that differs from that of the road along which the vehicles circulate, which entails a lack of homogeneity in the distance measurements in each part of the video frame. A Self-Organized Map (SOM), which models the distribution of the vehicles and their size, is applied to correct the perspective and compute the speed more precisely.

A proposal to estimate the pollution with a different schema was addressed in [24]. The authors propose a neural network model to infer pollution levels from existing data sources in a specific place by using traffic and meteorological data as inputs. Vehicle stops and delays, traffic flows and congestions, as well as wind speed, wind direction, rain, radiation and air pressure are taken into account for the estimation.

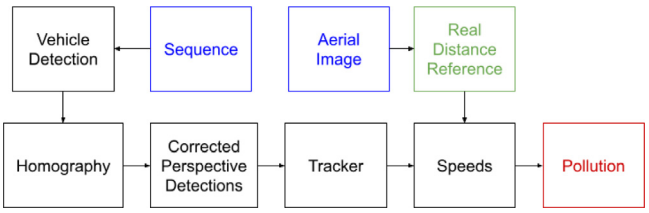


Fig. 1. Overall system scheme. A blue box indicates input data, a green one indicates manual selection, and a red one indicates output. Vehicles in the sequence are detected, their positions are corrected using an homography, and then they are tracked. Speed is obtained using each vehicle trace and the equivalence between pixels and meters from the aerial image. Pollution emitted by the vehicle is based on the estimated speed.

3. Proposal architecture

A general overview of the proposal scheme is displayed in Fig. 1. The information acquired by the IP camera is supplied to the model frame by frame. Vehicles from these incoming images are detected by an object detection method. Their positions are then corrected using an homography **H** in order to resemble an aerial perspective so that the distances are not distorted by camera perspective. A tracker records vehicle positions to be able to calculate the speed (pixels/time). Then, a real space reference (distance/pixel) from an aerial image is used to obtain real speed (distance/time). Once each vehicle speed is computed, that vehicle contribution to pollution is estimated for each frame.

Fig. 2 illustrates how the tracking is applied. Due to the high frames per second rates usual surveillance cameras present, two consecutive positions from the same vehicle should be very similar so a simple tracker based on matching last known vehicle centroids with incoming centroids from the current frame is enough to obtain a reliable trace for each vehicle.

4. Methodology

In this section, the methodology of our proposal is detailed. It is assumed that the vehicles move on a surface that can be regarded, to a first approximation, as a plane. In addition to this, it is also assumed that the video camera is static. This way, it is possible to project the acquired video frames on the plane where the vehicles move. In other words, we propose to estimate a homography **H** which projects the plane of the camera $\mathcal{C}$ on the plane where the vehicles move $\mathcal{V}$:

$$\mathbf{y} = \mathbf{H}\mathbf{x} \tag{1}$$

where:

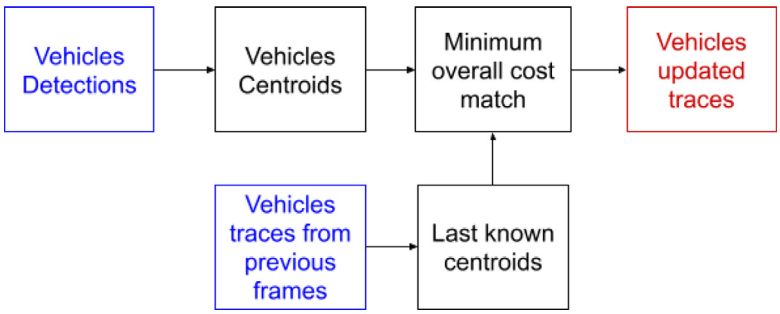


Fig. 2. Tracker scheme. Blue boxes indicate inputs, the red box indicates output. Centroids are obtained from vehicle detections, and matched with last known vehicle centroids in order to get the minimal overall cost.

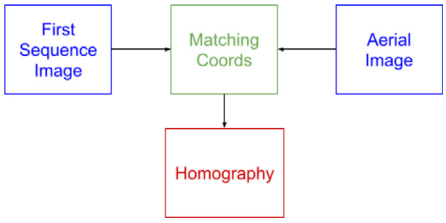


Fig. 3. Homography obtaining scheme. A blue, green and red box indicates input, manual selection and output respectively. The first image from the sequence and an aerial image of the same place are chosen. After manually selecting the same positions on both images an homography matrix **H** to correct the perspective is generated.

- $\mathbf{x} = (x_1, x_2, 1)^T$ are the homogeneous coordinates of a point in the video frame. The coordinates x_1 and x_2 are measured in pixels.
- $\mathbf{y} = (y_1, y_2, 1)^T$ are the homogeneous coordinates of the point in the plane where the vehicles move $\mathcal{V}$. The coordinates are rectified, i.e., y_1 and y_2 correspond to actual distances in the real world.
- $\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$ is a 3×3 homogeneous matrix with real values. Since **H** is homogeneous, it only has eight degrees of freedom because it is defined up to a scale.

The estimation of **H** is done offline, i.e., prior to the processing of the surveillance videos. Such estimation is carried out by mapping four or more calibration points from the $\mathcal{C}$ to $\mathcal{V}$. This way, a reference set of calibration points $\mathcal{R}$ is employed to estimate **H**:

$$\mathcal{R} = \{(\mathbf{x}_k, \mathbf{y}_k) \mid k \in \{1, \ldots, K\}\} \tag{2}$$

Please note that the larger K, the more accurate the estimation. Fig. 3 illustrates the homography obtaining process. In order to get the homography **H**, an image from the sequence and an aerial image from the same place are used. A number of common identifiable positions are selected on both images so they represent the same location with different perspective. The homography matrix is used to correct the perspective of the scene in order to accurately obtain the speed of each vehicle.

As a surveillance video is acquired, an object detection deep convolutional network is employed to process the incoming frames to detect vehicles. Each detected vehicle is defined by a set of points $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3, \tilde{\mathbf{x}}_4\}$ on the incoming video frame which correspond the four corners of a rectangle that encloses the vehicle. Then the points are projected on $\mathcal{V}$:

$$\tilde{\mathbf{y}}_i = \mathbf{H}\tilde{\mathbf{x}}_i \tag{3}$$

where $i \in \{1, 2, 3, 4\}$ is the index of the corner point. After that, the center $\hat{\mathbf{y}}$ of the vehicle on $\mathcal{V}$ is estimated by taking the mean of the corner points:

$$\hat{\mathbf{y}} = \frac{1}{4}\sum_{i=1}^{4} \tilde{\mathbf{y}}_i \tag{4}$$

It must be highlighted that this estimation of the center of the vehicle is invariant to the homography transformation, i.e., it does not matter whether the center is computed on $\mathcal{C}$ or on $\mathcal{V}$:

$$\hat{\mathbf{y}} = \frac{1}{4}\sum_{i=1}^{4} \tilde{\mathbf{y}}_i = \frac{1}{4}\sum_{i=1}^{4} \mathbf{H}\tilde{\mathbf{x}}_i = \mathbf{H}\left(\frac{1}{4}\sum_{i=1}^{4} \tilde{\mathbf{x}}_i\right) \tag{5}$$

so that the estimation of the center computed on $\mathcal{C}$ is given by:

$$\hat{\mathbf{x}} = \frac{1}{4}\sum_{i=1}^{4} \tilde{\mathbf{x}}_i \tag{6}$$

The estimated centers $\hat{\mathbf{y}}$ of the detected vehicles on $\mathcal{V}$ are supplied to an object tracker, that associates detected centers $\hat{\mathbf{y}}(t)$ at time instant t with detected centers $\hat{\mathbf{y}}(t+\tau)$ at time instant $t+\tau$, so that the actual velocity vector in the real world of a moving vehicle at time $t+\tau$ can be estimated as:

$$\mathbf{v}(t+\tau) = \frac{1}{\tau}\left(\hat{\mathbf{y}}(t+\tau) - \hat{\mathbf{y}}(t)\right) \tag{7}$$

Finally, the instantaneous pollution $z(t)$ generated by a vehicle at time t can be estimated as:

$$z(t) = F(v(t)) \tag{8}$$

$$v(t) = \|\mathbf{v}(t)\| \tag{9}$$

where $\|\cdot\|$ stands for the Euclidean norm of a vector, and F is a suitable estimation function that translates speed into pollution. Since the instantaneous pollution z is measured in emitted mass of pollutant per distance, the overall pollution Z associated to a vehicle during its appearance in the scene is estimated as follows:

$$Z = \sum_{t=1}^{T} v(t) F(v(t)) \tag{10}$$

where T is the number of time units t that the vehicle appears in the scene. Please note that you must multiply $z(t)$ by the traveled distance $v(t)$ during a unit of time t in order to obtain the emitted mass of pollutant during that unit of time, which is obtained as $v(t)z(t)$. After that, you sum the emitted masses of pollutant for all units of time t to obtain Z, as given by Eq. (10).

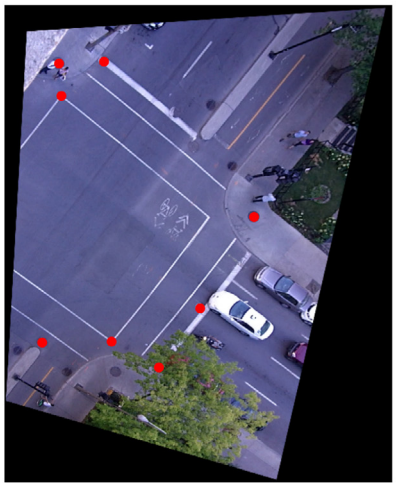
Please note that Eq. (10) can be regarded as a discrete time approximation of the following:

$$Z \approx \int_{1}^{T} v(t) F(v(t))\, dt \tag{11}$$

Eq. (11) is the limit of Eq. (10) when the length of the time intervals dt which are considered tends to zero.

(a) Original image from sequence *St-Marc*

(b) Real satellite reference image from crossing between *Rue Saint-Marc* and *Boulevard de Maisonneuve O*.



(c) Image 4a after applying Homography **H** to resemble image 4b perspective and proportions.

Fig. 4. Perspective correction. Red dots represent the points selected to perform the homography. In order to get 4(c), 4(a) is turned according to 4(b) perspective.

5. Experiments

5.1. Homography

In order to get the matrix **H**, eight identifiable and equivalent points from I_{video} and $I_{reference}$ are manually selected where I_{video} is a frame from the video sequence and $I_{reference}$ is the corresponding satellite image from the same place obtained from Google Maps.[1] Fig. 4 shows an example of the process. The real distance reference value is also obtained from Google Map along with $I_{reference}$. Since points are manually selected, the match is not entirely exact. Besides, these correspondences influence the calculation of the homography matrix and its accuracy. However, these slight variations do not result in a significant error in the computation of the real speed.

5.2. Speed estimation

In order to obtain pollution estimation, first each vehicle speed must be estimated for each frame. Speed is only measured between the coordinates used to create homography in order to get the better real space perspective. For the jth tracked vehicle, let us note the vehicle track $\mathbf{T}_{j,f} = (\hat{\mathbf{y}}_0, \hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_m)$ where f is the frame where the vehicle is first detected, $f + m$ is the frame where the vehicle is last detected, and $\hat{\mathbf{y}}_i$ is the jth vehicle center on corrected frame $f + i$ with $0 \leq i \leq m$. Let us also note p the real space reference given by the aerial image (measured in meters/pixel), and q the sequence number of frames per second (measured in frames/second). Three speed approximation procedures are proposed:

- Linear Approximation.
  Constant speed and straight direction are assumed for the entire vehicle track. We compute the speed from the first and the last detected positions of the vehicle. This leads to a constant estimation of $v_{i,j}$ in km/h for all $i, 0 \leq i \leq m$ as follows:
  $$v_{i,j} = 3.6 \frac{\left\| \hat{\mathbf{y}}_0 - \hat{\mathbf{y}}_m \right\|}{m} pq \tag{12}$$
  with $\|\cdot\|$ stands for the Euclidean norm of a vector.
- Piecewise Linear Approximation.
  Non constant speed and non straight direction are assumed. The speed is estimated for frame $f + i$ by comparing the position at frame $f+i$ and position at frame $f+i-u$ with u as a configurable parameter. This leads to a variable estimation for $v_{i,j}$ in km/h for $i, 0 \leq i \leq m$ as follows:
  $$v_{i,j} = 3.6 \frac{\left\| \hat{\mathbf{y}}_{i-u} - \hat{\mathbf{y}}_i \right\|}{m} pq \tag{13}$$
  where u is a tunable parameter and $\|\cdot\|$ stands for the Euclidean norm of a vector.
- Optical Flow.
  Here the speed is estimated by the average of the optical flow of the pixels that belong to the bounding box of the tracked vehicle. Let us note $\mathcal{S}_{j,i}$ the set of the pixels in $\mathcal{V}$ within the bounding box of the jth tracked vehicle at frame

[1] https://www.google.es/maps/.

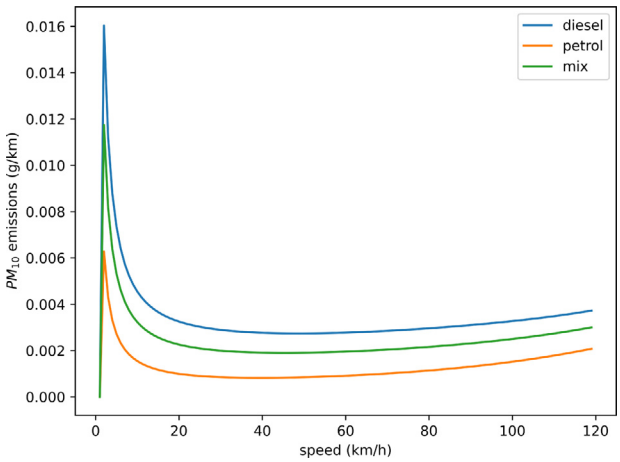


Fig. 5. Emission relation between speed (km/h) and emissions (g/km) for diesel and petrol cars according to Eq. (15) and Table 1. 'Mix' indicates the weighted curve following Eq. (16).

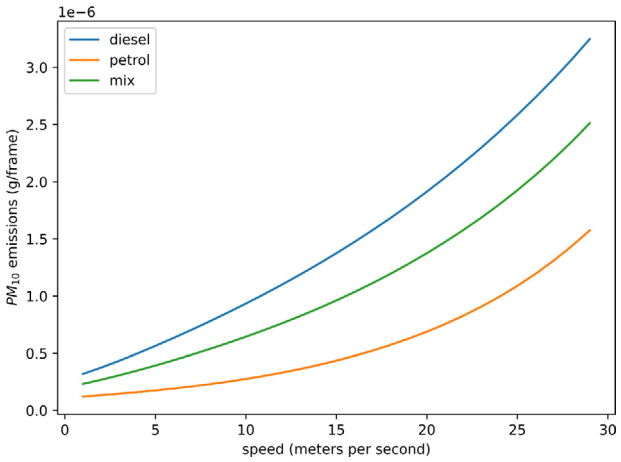


Fig. 6. Emissions relation between speed (m/s) and emissions (g/frame) for diesel and petrol cars following Eq. (18) and Table 1. Mix indicates the weighted curve following Eq. (16).

$f+i$. Then the optical flow $\varphi_{\mathbf{y}}$ (in pixels/frame) is computed for all the pixels $\mathbf{y}$ which belong to $\mathcal{S}_{j,i}$. This leads to a variable estimation for $v_{i,j}$ in km/h for $i, 0 \leq i \leq m$ as follows:

$$v_{i,j} = \frac{3.6pq}{\left|\mathcal{S}_{j,i}\right|} \sum_{\mathbf{y}\in\mathcal{S}_{j,i}} \varphi_{\mathbf{y}} \tag{14}$$

where $|\cdot|$ stands for the cardinal of a set. Gunnar-Farneback algorithm has been applied to obtain $\varphi_{\mathbf{y}}$.

5.3. Pollution estimation

In the same way that in [23], pollution estimation is based on the Emission Factor (F), which is measured in units of litre/100 km for fuel consumption and in g/km for PM_{10} particles. Our estimation is based on the emission curves published in Production of Updated Emission Curves for Use in the National Transport Model (PUEC from now on) from United Kingdom Department for Transport. The document is available online.[2]

$$F(v) = \frac{a + bv + cv^2 + dv^3 + ev^4 + fv^5 + gv^6}{v} \tag{15}$$

Eq. (15) models a general car emission curve, and Table 1 shows the adequate coefficient values according to PUEC provided data for year 2020.

Since our system cannot detect the fuel type for each car, Eq. (16) shows how the factor emission F is weighted.

$$F_{mix}(v) = S_{petrol} * F_{petrol}(v) + S_{diesel} * F_{diesel}(v) \tag{16}$$

with S_{petrol} = 0.44 and S_{diesel} = 0.56. S_{petrol} and S_{diesel} values are in line with the ratio of cars fueled by petrol and diesel respectively, according to 2020 data published in PUEC. Even though our system is able to identify several kinds of vehicles (car, truck, bus or motorcycle) it is assumed that the emission factor curve is the same for all vehicles.

Fig. 5 shows the relation between speed, which is measured in kilometers per hour (kph), and factor emission (g/km). The emissions are set to 0 if the speed is lesser than 1, with the aim of evading Eq. (15) infinite divergence.

Since Eqs. (15) and (16) outputs are expressed in mass of pollutants per kilometer (g/km) and it is desired to measure pollution by frame (g/frame), given the speed v_{ij} of jth vehicle from ith frame measured in km/h, and an interval of time $\triangle t$ measured in seconds, the following calculation is carried out in order to obtain the increment of pollution $\triangle Z_{ij}$ (g/frame):

$$\triangle r_{ij} = \frac{v_{ij}}{3600} \triangle t \tag{17}$$

$$\triangle Z_{ij} = F(v_{ij}) \triangle r_{ij} \tag{18}$$

where $\triangle r_{ij}$ is the distance traveled by vehicle j during frame i (km/frame), with increment of time $\triangle t = \frac{1}{q}$ and q the number of frames per second (see Fig. 6).

Estimated pollution for the jth vehicle Z_j and estimated pollution for the ith frame Z_i can be obtained as follows:

$$Z_j = \sum_i \triangle Z_{ij} \tag{19}$$

$$Z_i = \sum_j \triangle Z_{ij} \tag{20}$$

with $\triangle Z_{ij}$ being the estimated ith vehicle increment of pollution at frame j.

5.4. Resources

The video sequences which are chosen for the experiments are *Sherbrooke* (30 frames per second and 4000 frames) and *St-Marc* (30 frames per second and 2000 frames) from dataset [25].[3] These sequences were selected because the dataset provides the record position, hence map image can be retrieved (crossing between *Rue Sherbrooke* and *Avenue du parc la Fontaine* and crossing between *Rue Saint-Marc* and *Boulevard de Maisonneuve O*, both in Montreal, Canada). Both of them provide only partial annotations, thus the vehicles have been manually annotated using CVAT.[4]

As object detection methods, *ultralytics*[5] pretrained Yolo V5 [10, 26] with structure l based on backbone *ResNet 101* (*yolov5* from now on) and Tensorflow[6] pretrained Faster R-CNN based on backbone *ResNet V2* [9] (*fasterv2* from now on) have been used.

2 www.gov.uk/government/uploads/system/uploads/attachment_data/file/662795/updated-emission-curves-ntm.pdf.

3 www.jpjodoin.com/urbantracker/dataset.html.

4 github.com/openvinotoolkit/cvat.

5 github.com/ultralytics/yolov5/.

6 https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md.

Table 1
Coefficients to be applied to Eq. (15) following PUEC 2020 data.

	a	b	c	d	e	f	g
Petrol	0.01185628	0.00034047	1.2576E−6	1.0462E−7	−7.216E−10	6.0976E−12	0
Diesel	0.02918783	0.0013909	2.8984E−5	−6.175E−7	9.9971E−9	−7.31	2.1786E−13

Table 2
The table shows main statistics for system using manual annotations as well as *yolov5* and *fasterv2* object detection methods and Piecewise Linear Approximation as speed estimation method. The number of tracked vehicles is shown as well as the average generated pollution (g) per vehicle with its standard deviation, the average moving speed detected (km/h) with its standard deviation, the total pollution estimated (g), and the error percentage.

	Manual	*yolov5*	*fasterv2*
Sherbrooke			
Number of tracked vehicles	48	69	154
Average pollution by tracked vehicle	$3.974E-5 \pm 1.373E-5$	$1.513E-5 \pm 1.987E-5$	$5.351E-6 \pm 1.216E-5$
Average speed	37.712 ± 12.368	334.97 ± 14.616	29.126 ± 15.714
Total pollution	$1.908E-3$	$2.057E-3$	$2.751E-3$
% Pollution Error	–	7.84%	44.19%
St-Marc			
Number of tracked vehicles	10	17	51
Average pollution by tracked vehicle	$2.041E-5 \pm 8.188E-6$	$4.667E-6 \pm 7.982E-6$	$2.552E-6 \pm 6.004E-6$
Average speed	27.073 ± 14.953	20.069 ± 14.639	20.08 ± 13.842
Total pollution	$2.041E-4$	$2.1936E-4$	$5.0529E-4$
% Pollution Error	–	7.45%	147.53%

fasterv2 is a detection model based on two networks: first, a network uses selective search to generate region proposals where perhaps an object could be found, then a second network uses these proposals to detect the objects. *yolov5* instead of using two networks in order to propose and detect objects, uses only one network by dividing the image into a grid where detections are made for each grid cell. Due to *fasterv2* and *yolov5* different detection strategies, their advantages are different. While *fasterv2* should be the most robust model, it is also slower than *yolov5*. *yolov5* has proven to be a detection model with a great speed–performance ratio but it should fail to detect many small objects grouped together.

The system has been implemented using Python 3,[7] Tensorflow,[8] [27] Pytorch[9] [28] and Numpy [29].

5.5. Evaluation

Due to the critical role which the object recognition method plays within the system, two different object detection methods have been tested. The system performance is compared with the performance obtained by means of the manual annotations (ground truth). The tth frame accumulated pollution (AZ_t) is defined as follows:

$$AZ_t = \sum_{i=1}^{t} Z_i \tag{21}$$

With Z_i as ith frame pollution as defined by Eq. (20).

Accumulated Pollution Error (AZE) at frame t is therefore defined as:

$$AZE_t = |AZ_{manual,t} - AZ_{r,t}| \tag{22}$$

where $AZ_{r,t}$ is AZ_t when the object detection system r is used, and $r \in \{yololv5, fasterv2\}$.

[7] www.python.org/.
[8] www.tensorflow.org/.
[9] pytorch.org/.

5.6. Results

Fig. 7 is composed of images which are taken at different stages of the system process. They show the perspective correction made by the homography.

Fig. 8 shows the pollution estimated as well as the number of tracked vehicles for each frame for both sequences and different object detection methods using Piecewise Linear Approximation as speed estimation method. It can be observed the strong relationship between the number of moving vehicles and the estimated level of pollution. According to these results, the object detection method performance seems to be essential for the system to work properly and accurately. *fasterv5* shows greater number of tracked vehicles than both manual annotations and *yolov5*, this vehicles fake traces were generated due to duplicate detections.

Figs. 9 and 10 shows how the estimated accumulated pollution evolves for each object detection method using Piecewise Linear Approximation as speed estimation method and how accumulated pollution estimation error evolves compared to use manual annotations. The results reinforce the idea that the object detection method performance is key for the system performance. In both videos, the pollution obtained using *fasterv2* is further away from that obtained using manual annotations than when*yolov5* was used. This difference always occurs by increasing the pollution and is probably due to *fasterv2*'s duplicate detection propensity, which creates fake traces recording non-existent pollution.

Considering Table 2 and relating its content to the results shown by Fig. 10, it can be observed that the number of different detected traces is not critical to make a better estimation of pollution as long as no vehicle is lost for too long and no vehicles are detected double. Although the system may temporarily lose a vehicle due to failure of the detection method, once the system detects the moving object again, their contribution to pollution is taken into consideration, even though that object is understood as a different vehicle. Since our system does not aim to count vehicles but to estimate their pollution, this is not relevant and it is preferable to lose a vehicle and find it later even supposing that it is a different vehicle than losing it and do not find it again due to object detection method failures.

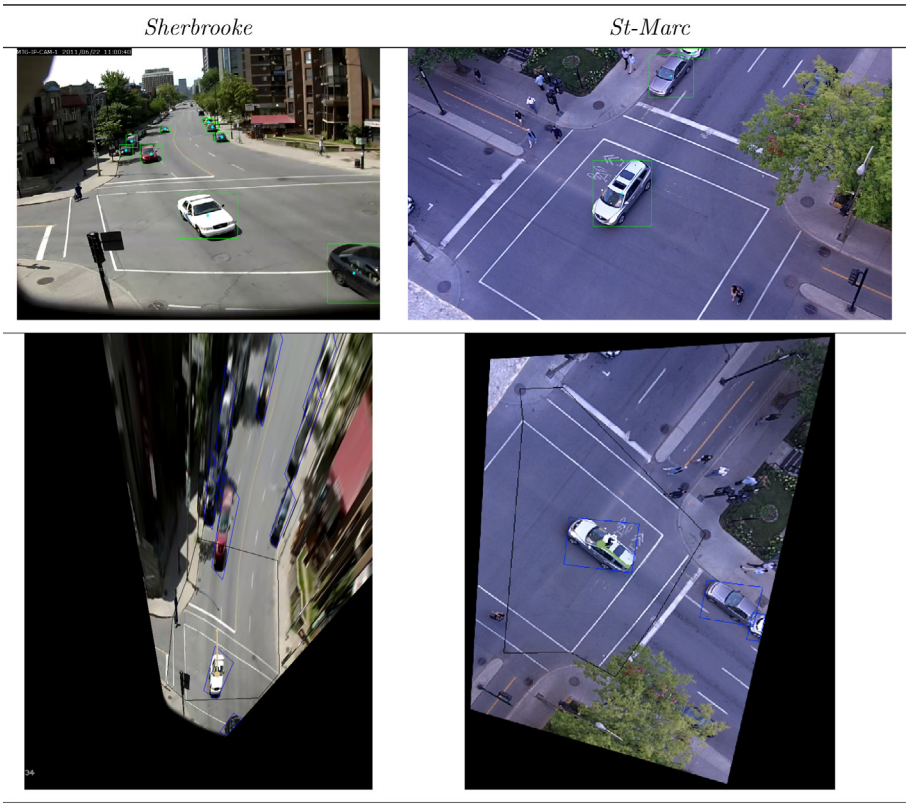


Fig. 7. Qualitative results. First row shows original images (frame 34 from *Sherbrooke* and frame 894 from *St-Marc*) with vehicles detected by *fasterv2*. The second row shows the same images after the perspective is corrected by means of the homography. The images shows the coordinates used to create homography (polygon marked as black lines), and the tracked vehicles shows their last trajectory.

Fig. 11 shows how accumulated pollution error evolves for each video and speed approximation method using manual annotations instead of an object detection method. Linear Approximation results in a greater accumulated pollution in both videos while Optical Flow seems to obtain the lowest. Applied to *St-Marc* sequence, Both Piecewise Linear Approximation and Optical Flow obtain a similar Behavior. The Linear Approximation as the outlier method is the expected behavior due to the naive assumptions it requires. Both Optical Flow and Piecewise Linear Approximation are expected to approximate better the speed and the great difference between them applied to *Sherbrooke* video is probably due to the relationship between the performance of those methods and the video conditions, such as distance, angle and type of movements.

6. Conclusions

In this work, a methodology to estimate pollution from vehicles in traffic lanes using the IP cameras already installed throughout the cities has been proposed. This framework takes advantage of the latest deep learning-based object detection models for the detection and tracking of vehicles on the road. The pollution calculation is based mainly on the speed of the vehicles that circulates through the region of the analyzed scene. For that purpose, it has been necessary to obtain a correspondence between distance measures in meters and the pixels of the image by means of homography transformations which permit real distances to be obtained from the image of the location of the scene on Google Maps. Two of the main object detection techniques, FasterRCNN and YoloV5, have been studied and they show very different results, allowing to conclude the importance of an appropriate object detection method and recommending the use of YoloV5 for this purpose. Three speed estimation methods are also tested (Linear Approximation, Piecewise Linear Approximation and Optical Flow) with similar outcome using Piecewise Linear Approximation and Optical Flow in one of the sequences and Linear Approximation estimations as outliers due to their naive assumptions.

The main limitation of our proposal is that it is not possible to obtain a data set of emissions per vehicle together with the sequence of images where these vehicles are driving. Thus, although our proposal is theoretically based and the results on real traffic videos are consistent, they cannot be agreed with actual pollution data. This disadvantage could be seen as one of the main challenges to move forward. In addition, it would be interesting to automate the correspondence between the common identifiable positions of the analyzed sequence with the key points of the map.

Nevertheless, these results provide a reliable and valid starting point to implement a pollution monitoring strategy throughout a city and using only the cameras already installed as a resource.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by the Ministry of Science, Innovation and Universities of Spain under grant RTI2018-094645-B-I00, project name “Automated detection with low-cost hardware of unusual activities in video sequences”. It is also partially supported by the Autonomous Government of Andalusia (Spain) under project UMA18-FEDERJA-084, project name “Detection of

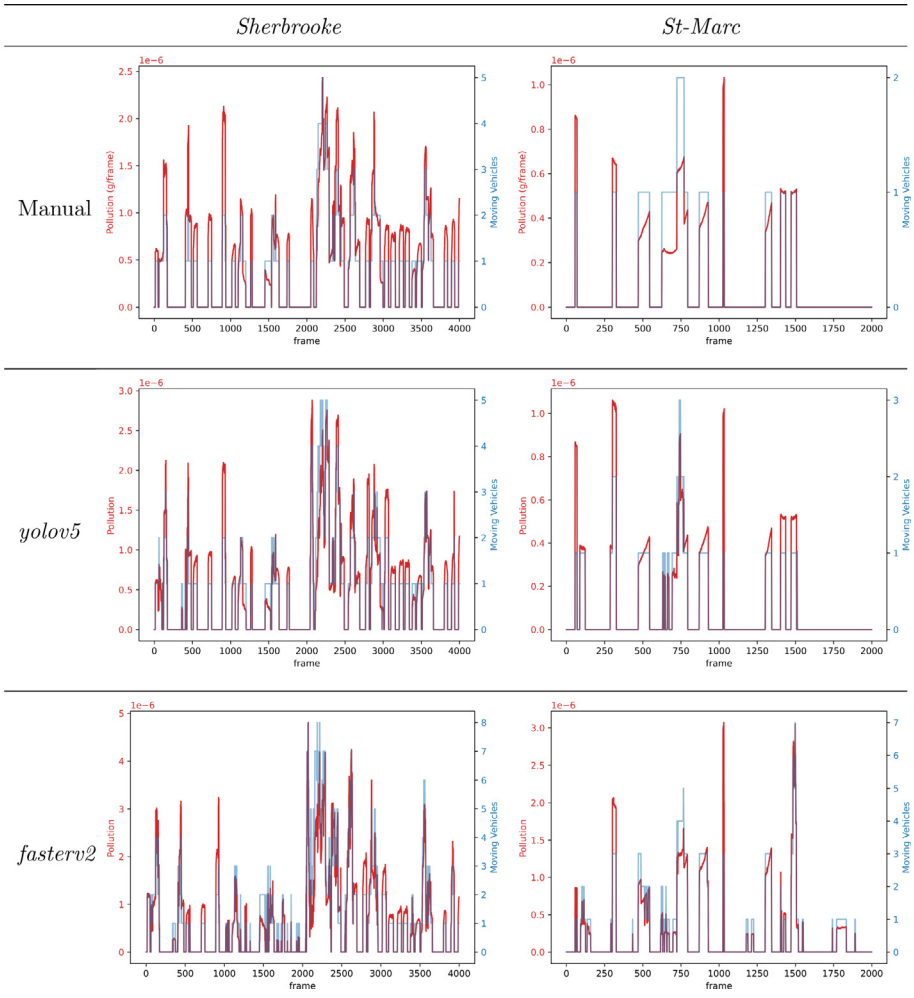


Fig. 8. Color red and left vertical axis indicate mass of air pollutants (PM_{10} g) by frame as defined by Eq. (18). Color blue and right vertical axis indicate moving vehicles detected at each frame. It is important to note that moving Vehicles scales are different to adapt to each video and method. Each column corresponds to a video sequence (*Sherbrooke* (first column) or *St-Marc* (second column)), whereas each row is devoted to a different object recognition method all using Piecewise Linear Approximation as speed estimation method.

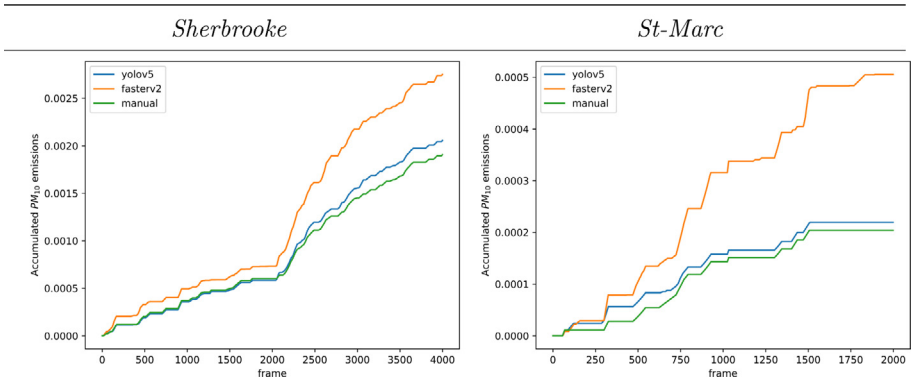


Fig. 9. Accumulated pollution (AZ_t) for each sequence and object recognition method using Piecewise Linear Approximation as speed estimation method (the nearest to *manual* (green), the better).

anomalous behavior agents by deep learning in low-cost video surveillance intelligent systems". All of them include funds from the European Regional Development Fund (ERDF). It is also partially supported by the University of Malaga (Spain) under grants B1-2019_01, project name "Anomaly detection on roads by moving cameras", and B1-2019_02, project name "Self-Organizing Neural Systems for Non-Stationary Environments". The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They also gratefully acknowledge the support of NVIDIA Corporation with the donation of two Titan X GPUs. Finally, the authors thankfully acknowledge the grant of the Universidad de Málaga and the Instituto de Investigación Biomédica de Málaga - IBIMA. Funding for Open Access charge: University of Málaga/CBUA.

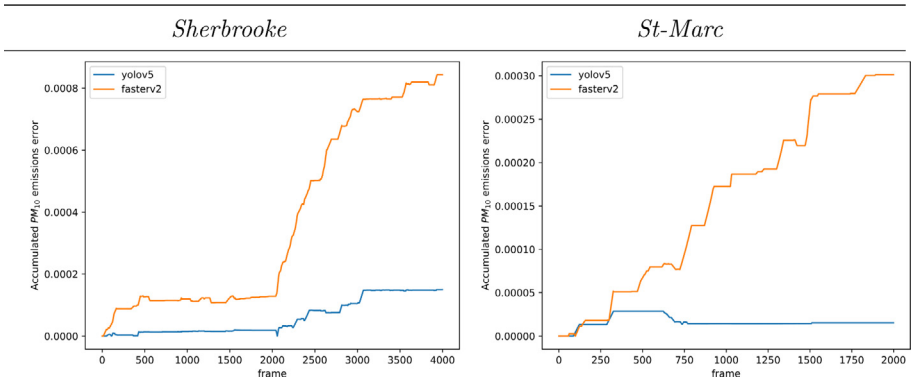


Fig. 10. Accumulated pollution Error (AZE_t) for each sequence and tested object recognition method using Piecewise Linear Approximation as speed estimation method (the lower, the better).

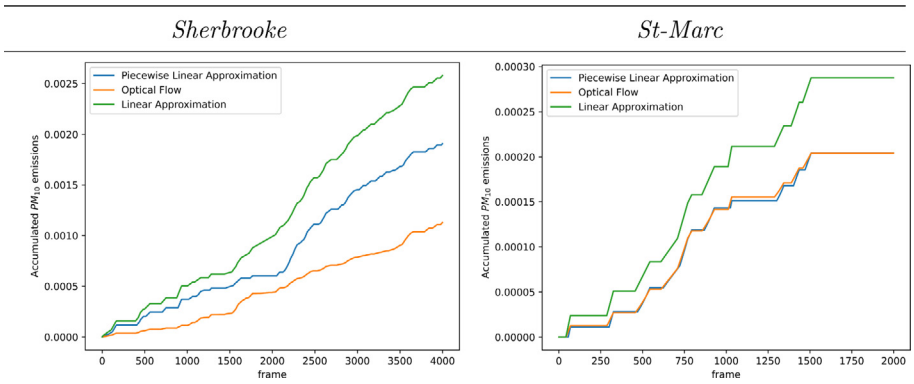


Fig. 11. Accumulated pollution (AZ_t) for each sequence and tested speed approximation method using manual annotations instead of an object detection method.

References

[1] O.V. Lozhkina, V.N. Lozhkin, Estimation of road transport related air pollution in Saint Petersburg using European and Russian calculation models, Transp. Res. 36 (2015) 178–189.

[2] S. Yang, Y.-J. Wu, J. Woolschlager, Integrated modeling framework for highway traffic pollution estimation and dispersion, Am. J. Environ. Sci. 12 (3) (2016) 140–151, http://dx.doi.org/10.3844/ajessp.2016.140.151.

[3] Z. Wang, F. Lu, Q.-C. Lu, D. Wang, Z.-R. Peng, et al., Fine-scale estimation of carbon monoxide and fine particulate matter concentrations in proximity to a road intersection by using wavelet neural network with genetic algorithm, Atmos. Environ. 104 (2015) 264–272.

[4] T. Bouwmans, Traditional and recent approaches in background modeling for foreground detection: An overview, Comp. Sci. Rev. 11–12 (2014) 31–66, http://dx.doi.org/10.1016/j.cosrev.2014.04.001.

[5] Z. Zivkovic, F. Van Der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, Pattern Recognit. Lett. 27 (7) (2006) 773–780.

[6] R. Luque, E. Domí nguez, E. Palomo, J. Muñoz, An ART-type network approach for video object detection, in: Proceedings of the 18th European Symposium on Artificial NeuralNetworks - Computational Intelligence and Machine Learning, ESANN 2010, 2010, pp. 423–428.

[7] H. Xue, Y. Liu, D. Cai, X. He, Tracking people in RGBD videos using deep learning and motion clues, Neurocomputing 204 (2016) 70–76, http://dx.doi.org/10.1016/j.neucom.2015.06.112.

[8] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, http://dx.doi.org/10.1038/nature14539.

[9] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, 2016, arXiv:1506.01497.

[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, 2016, arXiv:1506.02640.

[11] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Comput. Surv. 38 (4) (2006) http://dx.doi.org/10.1145/1177352.1177355.

[12] M.A. Molina-Cabello, R.M. Luque-Baena, E. López-Rubio, J.M. Ortiz-de Lazcano-Lobato, E. Domínguez, 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.

[13] M.A. Molina-Cabello, R.M. Luque-Baena, E. López-Rubio, 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.

[14] M.A. Molina-Cabello, R.M. Luque-Baena, E. Lopez-Rubio, K. Thurnhofer-Hemsi, Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images, Integr. Comput.-Aided Eng. 25 (4) (2018) 321–333.

[15] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.

[16] R.M. Luque-Baena, E. López-Rubio, E. Domínguez, E.J. Palomo, J.M. Jerez, A self-organizing map to improve vehicle detection in flow monitoring systems, Soft Comput. 19 (9) (2015) 2499–2509.

[17] A. Argüelles Cruz, M. De Luis, P. Moreno Aguilera, C. Yáñez Márquez, Mobile system surveillance for vehicular pollutants emission, based on Wi-Fi ad-hoc network, Lecture Notes in Comput. Sci. 8276 (2013) 294–302, http://dx.doi.org/10.1007/978-3-319-03176-7_38.

[18] C. Shetty, B. Sowmya, S. Seema, K. Srinivasa, Air pollution control model using machine learning and IoT techniques, Adv. Comput. 117 (1) (2020) 187–218, http://dx.doi.org/10.1016/bs.adcom.2019.10.006.

[19] C. Zhang, J. Yan, C. Li, X. Rui, L. Liu, R. Bie, On estimating air pollution from photos using convolutional neural network, in: Proceedings of the 24th ACM International Conference on Multimedia, 2016, pp. 297–301, http://dx.doi.org/10.1145/2964284.2967230.

[20] S. Kundu, U. Maulik, Vehicle pollution detection from images using deep learning, in: Intelligence Enabled Research: DoSIER 2019, Springer Singapore, 2020, pp. 1–5, http://dx.doi.org/10.1007/978-981-15-2021-1_1.

[21] X. Wang, Y. Kang, Y. Cao, Sdv-net: A two-stage convolutional neural network for smoky diesel vehicle detection, in: Chinese Control Conference, CCC, Vol. 2019-July, 2019, pp. 8611–8616, http://dx.doi.org/10.23919/ChiCC.2019.8865919.

[22] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.

[23] M.A. Molina-Cabello, R.M. Luque-Baena, E. López-Rubio, L. Deka, 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.

[24] M.A. Molina-Cabello, B.N. Passow, E. Dominguez, D. Elizondo, J. Obszynska, Infering air quality from traffic data using transferable neural network models, in: International Work-Conference on Artificial Neural Networks, Springer, 2019, pp. 832–843.

[25] J. Jodoin, G. Bilodeau, N. Saunier, Urban tracker: Multiple object tracking in urban mixed traffic, in: IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 885–892, http://dx.doi.org/10.1109/WACV.2014.6836010.

[26] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, A. Chaurasia, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, F. Ingham, Frederik, Guilhen, A. Colmagro, H. Ye, Jacobsolawetz, J. Poznanski, J. Fang, J. Kim, K. Doan, L. Yu, Ultralytics/yolov5: v4.0 - nn.SiLU activations, Weights & Biases logging, PyTorch Hub Integration, Zenodo, 2021, http://dx.doi.org/10.5281/zenodo.4418161.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, (32) 2019, pp. 8024–8035.

[29] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, http://dx.doi.org/10.1038/s41586-020-2649-2.