Road pollution estimation using static cameras and neural networks

Miguel A. Molina-Cabello*, Rafael Marcos Luque-Baena*, Ezequiel López-Rubio*,
Lipika Deka† and Karl Thurnhofer-Hemsi*
**Department of Computer Languages and Computer Science*
University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
†*Department of Computer Technology*
De Montfort University, The Gateway, Leicester LE1 9BH, United Kingdom
Emails: {miguelangel,rmluque,ezeqlr,karlkhader}@lcc.uma.es, lipika.deka@dmu.ac.uk*†

Abstract—**This paper presents a methodology for estimating pollution on roads by analyzing traffic video sequences. The objective is to take advantage of the huge network of static cameras which is possible to find in the road system of any state or country to estimate the pollution on each area. This proposal uses deep learning neural networks for the object detection, and a pollution estimation model based on the frequency of vehicles and their speed. The experiments show promising results which suggest that the system can be used alone or combined with existing systems for measuring pollution on roads.**

1. Introduction

Currently, the latest advances in research related to the field of traffic video surveillance, have allowed to study other very interesting factors from the analysis and detection of vehicles along a road. In addition, the rise in the implementation and use of IP cameras, mainly for security reasons, is generating such a large amount of information that we could use to analyze the normal behavior of vehicles, detect anomalous patterns (for example, driving in the opposite direction) or estimate the air pollution in the traffic environment.

The assessment of air pollution caused by vehicle emissions and forecasting of air quality have been managed from different points of view [1]. One of approaches consist to measure the air concentration produced by traffic with monitoring sensors, although it is not highly suitable for monitoring large areas due to the cost of sensor installation and application. In [2] the authors manage to determine traffic pollution at road intersections using hybrid models that combine wavelet neural network and genetic algorithms. Additionally, in [3] a comparison between two different emission and dispersion models is analyzed.

Unlike other models that estimate traffic pollution based on air quality sensors, environmental sensors and sensors that determine traffic density, only the static cameras present on the roads are used in this work. Our proposal tries to optimistically estimate the level of traffic pollution from the vehicle motion analysis on the highways. To do this, on each sequence frame, the number of vehicles circulating and their speed is detected. With this information, it is possible to estimate the pollution level produced in each instant of time.

The propose methodology begins with a phase of detection of the vehicles that appear in the scene [4]. Due to the improvement in the power of the hardware devices, the recent development of deep learning techniques (which allow to tackle complex tasks such as the recognition of objects) is being progressively incorporated in the field of traffic video surveillance [5]. In fact, many traditional techniques for the detection of foreground objects (Gaussian mixture [6], statistical background modeling [7], etc.) are being replaced by deep neural networks, which provide much higher success rates in the identification and detection of objects [8]. In this work we will use the Faster-RCNN network to recognize the vehicles in the scene [9]. Subsequently, a tracking phase is considered in order to obtain the partial trajectories along the road [10].

The perspective of the camera makes difficult the computation of the speed of each vehicle. A self-organized neuronal network which model the distribution of the vehicles and their size, is applied to correct this perspective. Using the calculated speed and the number of vehicles, it is possible to estimate the pollution in each frame, and consequently, in the whole scene.

The remaining of the paper is structured as follows. Section 2 presents the architecture of our approach, where each subsection describes each part in detail. Section 3 outlines the experimental results carried out, whereas Section 4 summarizes the conclusions.

2. System architecture

The developed proposal can be described as it is exhibited in Figure 1. A frame of a video sequence is provided to the system, which is composed by three modules. The first one is a vehicle detection and tracking process, when we select the desired objects (vehicles in this case) from the remaining objects (people, plants or others). The second one is a speed estimation module in order to calculate the

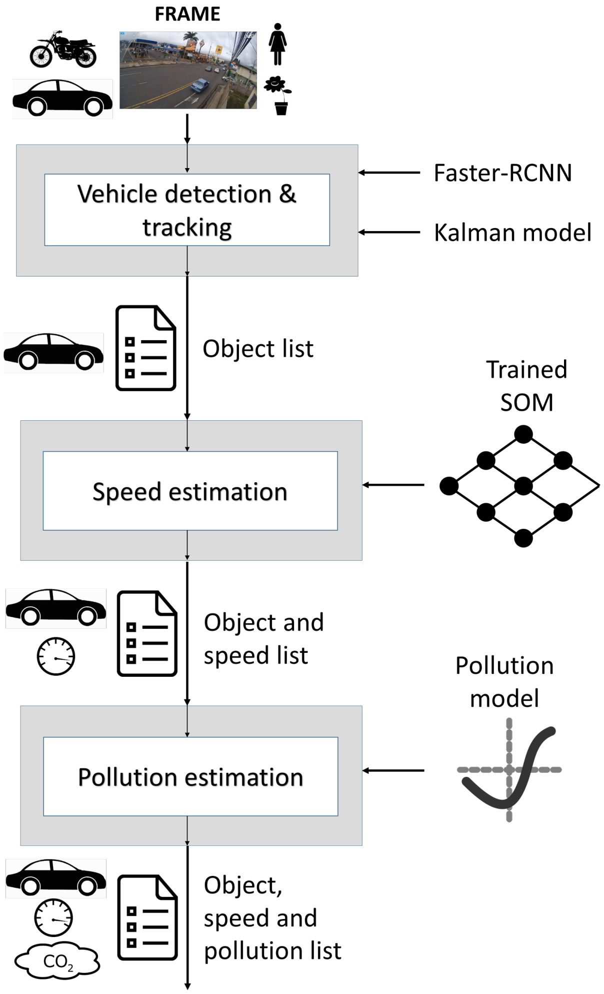


Figure 1. Schema of the operation of the developed approach.

velocity of each detected vehicle. And finally, we can estimate the contamination produced by each detected vehicle considering its speed with a pollution estimation module.

2.1. Vehicle detection and tracking

The Vehicle detection and tracking module is composed by two steps. The first one is an object detection and classification process, and after that, we have a tracking process in order to manage the trajectories of the detected vehicles.

The object detection and classification process is based on a deep learning architecture. In this case, we have used the Faster-RCNN model [11], which employs Convolutional Neural Networks and provides the area and the class of the detected objects. We have considered a pretrained model to detect the 20 objects included in the PASCAL VOC 2007 dataset [12].

Given an image corresponding to the frame t of a video, the output of the model is a set of detected objects, with their area and their probabilities of belonging to each possible class. Let i be one of the detected objects in the frame t, the output of the object detection module is:

$$\mathbf{h}_{i,t} = (h_{i,t,1}, h_{i,t,2}, h_{i,t,3}, h_{i,t,4}) \in \mathbb{R}^3 \tag{1}$$

$$\mathbf{q}_{i,t} = (q_{i,t,1}, ..., q_{i,t,K}) \in \mathbb{R}^K \tag{2}$$

where $(h_{i,t,1}, h_{i,t,2})$ are the upper-left corner location of the bounding box corresponding to the i-th detected object in the current frame and $(h_{i,t,3}, h_{i,t,4})$ are the width and height, respectively, of this bounding box, expressed in pixels. Associated to each detection there is a probability of belonging to each object class, $q_{i,t,k} \in [0, 1]$, where $C_k \in Classes$ and the possible number of object classes is K. In this case, $K = 20$.

After that, in a frame t we have applied a threshold τ and we just consider those objects with $q_{i,t,k}$ higher than these threshold, in order to consider only the vehicles that appear in the frame.

When we have the detected vehicles, a tracking stage is required in order to obtain their partial trajectories. This information is crucial to determine the speed and pollution in each frame. The Kalman filter is applied to perform a correspondence among the detected objects in a frame and the tracked objects. This technique is based on a prediction - correction scheme of the object centroids along the sequence [13].

2.2. Speed estimation

Most of the cameras located on highways capture the images with perspective, which causes that there is no homogeneity in the distances in each part of the frame. Thus, in order to estimate the real distances in the scenario and the speed of the vehicles, a Self-Organizing Map (SOM) model is considered [14]. A feature vector $\mathbf{z} \in \mathbb{R}^D$ is extracted from each detected object where D is the number of chosen features. In this case, geometric information represented by the area and the height and width of its bounding box is sufficient to estimate the distances in pixels of each vehicle. These values form the feature vector. Thus, the aim of the network is to learn a smooth function:

$$\mathcal{F} : \mathbb{R}^2 \to \mathbb{R}^D \quad \mathbf{y} = \mathcal{F}(\mathbf{x}) \tag{3}$$

where $\mathbf{x}$ is a pixel location in the video frame.

The M units of the self-organizing map, which are arranged in a rectangular topology of size $a \times b$, are represented by two prototypes, one for input vectors $\mathbf{w} \in \mathbb{R}^2$ (pixel coordinates) and one for output vectors $\mathbf{v} \in \mathbb{R}^D$ (typical object features at pixel coordinates $\mathbf{w}$). The input prototype is used to compute the winning unit, whereas the output prototype is used to estimate the smooth function $\mathcal{F}$. Therefore, the vehicles that are associated with the neuron i will use the output $\mathbf{v_i}$ to estimate the speed at which they are driving at that moment, and whose location within the frame will be close to $\mathbf{w_i}$.

The training of the SOM is based on the size of the motorcycles, since the are very similar between them according

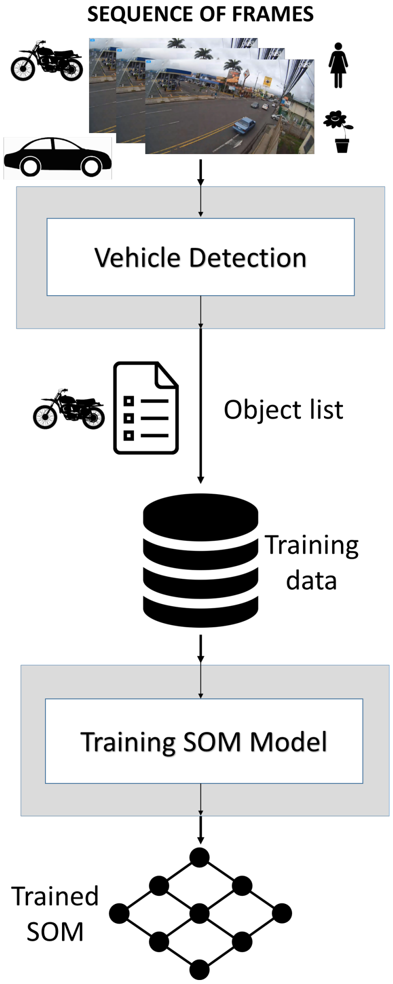


Figure 2. Schema of the training SOM process.

to their aspect ratio. In the case of a car, it exists a higher difference between the shortest and the largest car.

First of all, in order to generate a training SOM dataset, we have carried out the object detection and classification module with a video from the selected scenario. For each frame, we have obtained the list of the detected object with their classes and bounding box (area and position), and we have just considered those object indicated as motorcycle and with a high probability to belong to the motorcycle class. So that, the object i in the frame t will be considered to belong to the training dataset if:

$$\forall k \in K(\mathbf{q}_{i,t,k} \leqslant \mathbf{q}_{i,t,m}) \wedge (\mathbf{q}_{i,t,m} > \tau) \wedge (C_m = motorcycle) \tag{4}$$

In addition, if a side of the bounding box corresponds to a border of the image, we have discarded this vehicle.

After that, with the bounding box of each detected motorcycle we have trained a SOM model, considering a standard weight of a motorcycle. In order to estimate this value, we can select the number of licensed motorcycles by engine size. Then, choosing the most licensed known model nowadays with this engine size and taking the weight of it. Additionally, the height of a person driving a motorcycle is also considered. Thus, we can estimate in each area of the frame the correspondence between meters and pixels.

A schema of the training SOM process is shown in Figure 2.

2.3. Pollution estimation

We have considered an estimation of the pollution based on the emission factor (EF), which is a measure in units of g/km for PM_{10} and litre/100km for fuel consumption. The model to estimate the emission factor is based on the curves from the Production of Updated Emission Curves for Use in the National Transport Model report, which is available in its website[1]. This curves are defined by the following equation:

$$y(x) = \frac{a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6}{x} \tag{5}$$

where x is the speed in kph.

In order to estimate the pollution produced by a car, because we cannot assure its fuel type, we have considered the proportion of car by fuel type (petrol or diesel) on motorway roads. Thus, the emission factor produced by a car, considering the emission curves for the petrol and diesel fuel vehicle type (y_p and y_d, respectively), is defined as follow:

$$\begin{aligned} EF(x) = {} & petrol\_car\_proportion * y_p(x) \\ & + diesel\_car\_proportion * y_d(x) \end{aligned} \tag{6}$$

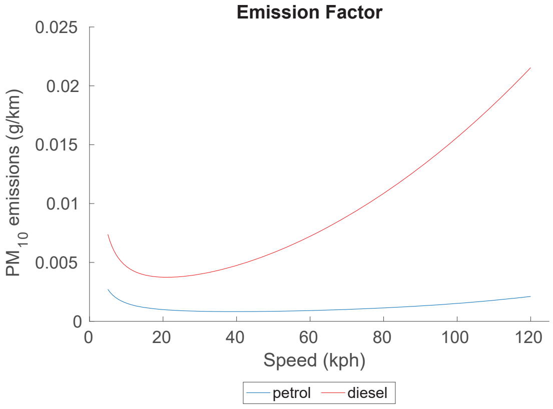


Figure 3. Emission factor curves for petrol and diesel car types corresponding to the pollution estimation module.

[1] https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/662795/updated-emission-curves-ntm.pdf

TABLE 1. CONSIDERED PARAMETER VALUES.

Method	Parameters
Faster RCNN	threshold $\tau = 0.50$
	$model\_dir =$ faster_rcnn_VOC0712_vgg_16layers
	$per\_nms\_topN = 6000$
	$nms\_overlap\_thres = 0.7$
	$after\_nms\_topN = 300$
	$test\_scales = 600$
SOM	$num\_steps = 100000$
	$num\_steps\_per\_epoch = 10000$
	$num\_neurons = 25$
	$num\_rows\_map = 4$
	$num\_cols\_map = 4$
	$initial\_learning\_rate = 0.4$
	$max\_radius = sqrt(num\_neurons)/8$
	$convergence\_learning\_rate = 0.01$
	$convergence\_radius = 1$
	$usual\_moto\_lenght = 2.080$
	$usual\_person\_driving\_moto\_lenght = 1.700$
EF	$petrol\_car\_proportion = 0.29$
	$diesel\_car\_proportion = 0.71$
	$a\_petrol = 0.01185628$
	$b\_petrol = 0.00034047$
	$c\_petrol = 1.2576E - 06$
	$d\_petrol = 1.0462E - 07$
	$e\_petrol = -7.216E - 10$
	$f\_petrol = 6.0976E - 12$
	$g\_petrol = 0$
	$a\_diesel = 0.02918783$
	$b\_diesel = 0.0013909$
	$c\_diesel = 2.8984E - 05$
	$d\_diesel = 6.175E - 07$
	$e\_diesel = 9.9971E - 09$
	$f\_diesel = -7.31E - 11$
	$g\_diesel = 2.1786E - 13$

3. Experimental Results

The computational experiments that we have carried out and their results are shown in this section. First of all, Subsection 3.1 exhibits the software and hardware that have been used. Then, in Subsection 3.2, we have specified the tested video sequences. The tuned parameters of the software can be observed in Subsection 3.3. Finally, the obtained results from the experiments are described in Subsection 3.4.

3.1. Methods

Our implementation of the system is written in Matlab. The speed estimation and the pollution estimation modules are implemented by our group, while the vehicle detection module is based on the Faster R-CNN library, which can be accesible in its website[2].

The reported experiments have been carried out on a 64-bit Personal Computer with an eight-core Intel i7 3.60 GHz CPU, 32 GB RAM and a Titan X GPU.

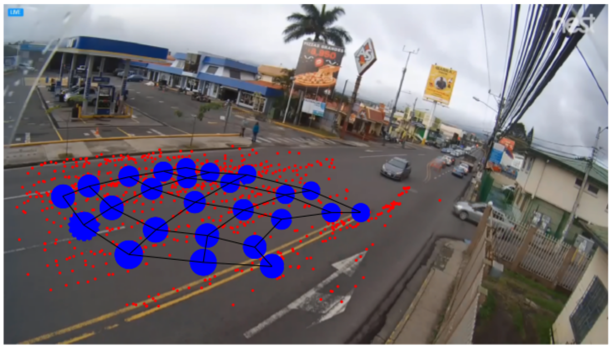


Figure 4. Training data and trained SOM with them shown on a frame of the background of the selected video. The red points are the training samples, the blue circles are the prototypes of the SOM and the lines between the prototypes are the connections between them. It can be observed that the prototypes closer to the camera are larger than other prototypes, which means that a higher number of pixels corresponds to one real meter.

3.2. Sequences

We have considered an specific scenario, which is available on the *camaras viales* website[3], in order to carry out the experiments. The selected scenario is *camara-guadalupe*, where a camera is recording a road 24 hours a day[4]. We have taken a video from this scenario, and it is composed by 36005 frames with a size of 1080x1920 pixels. This video can be downloaded from our website[5].

3.3. Parameters selection

The configuration of the parameters of the different modules that compose our system are described in this subsection.

First of all, the values of the parameters of the Faster-RCNN module are those that their authors recommend as default values. We only have changed the threshold in order to recognize the more possible number of cars.

On the other hand, the most important parameter of the SOM module is the number of neurons and it is selected to cover the regions of the video with more activity. Additionally, in the motorcycle detection process in order to obtain the training SOM dataset, we have employed a Faster-RCNN threshold with $\tau = 0.90$ due to achieve a robust training data. In addition, in order to estimate the value of the standard weight of a motorcycle, we have chosen the number of licensed motorcycles by engine size in UK[6] and it is said that the average engine size is approximately 600 cc. After that, we have selected the most licensed known model nowadays with this engine size[7] and it corresponds to the

[2] https://github.com/ShaoqingRen/faster_rcnn

[3] https://www.camarasviales.com

[4] https://www.camarasviales.com/camara-guadalupe

[5] https://www.lcc.uma.es/~miguelangel/resources/fixed_camera/camarasviales-guadalupe_2018-01-18_23-30-00.mp4

[6] https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/608185/veh0306.ods

[7] https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/666910/veh0120.ods

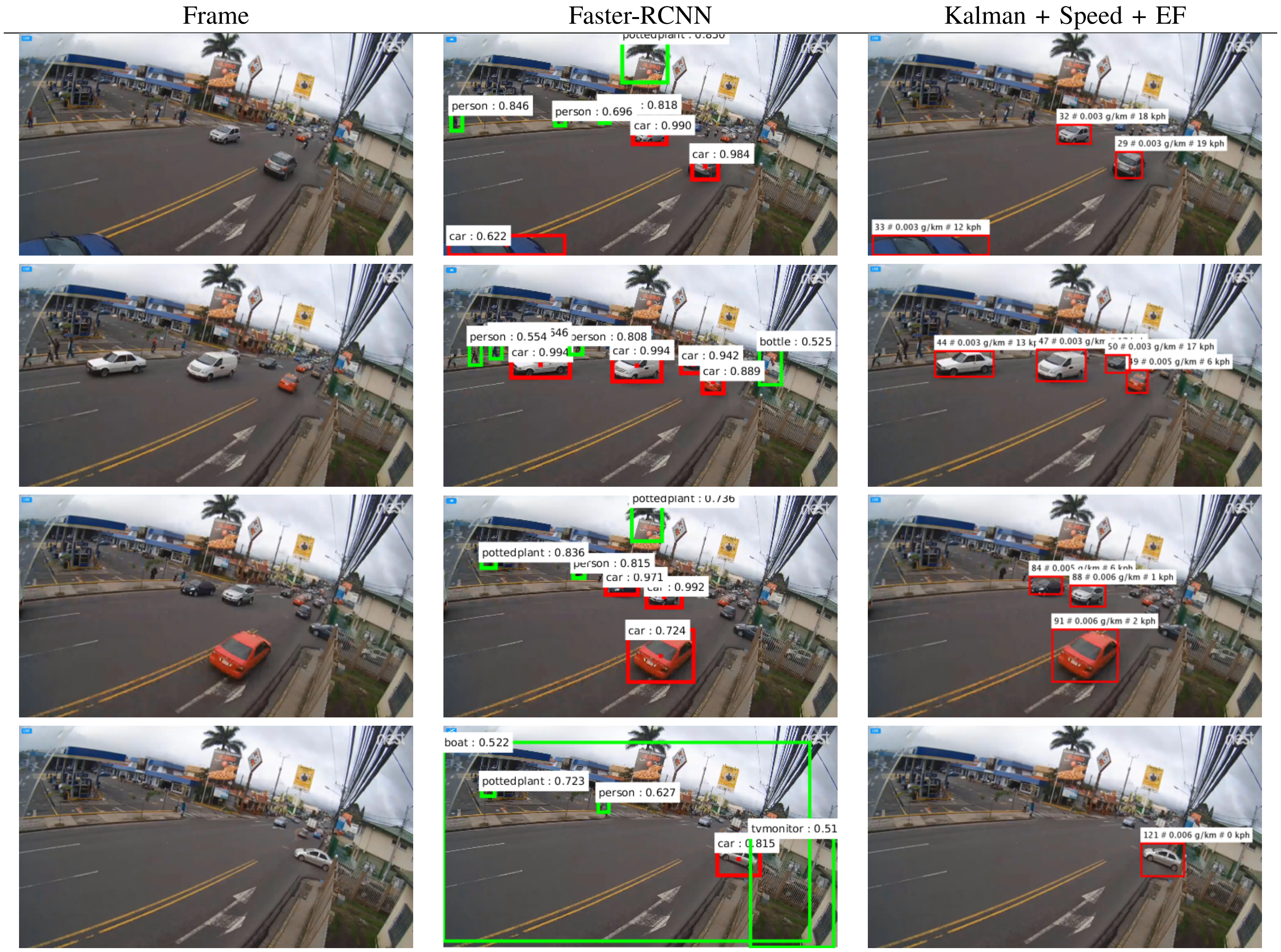


Figure 5. Graphical description of the operation of the proposed method. From left to right, the columns show a frame of a sequence, the objects detected on it provided by the Faster-RCNN module, and the detected cars with the trajectory given by the Kalman model, and speed and pollution estimation information. The rows show frames 698, 1000, 1651 and 2563 of camara-guadalupe. Note that the red bounding boxes correspond to undesired object detected and the green bounding boxes correspond to the objects which belong to the *car* class.

YAMAHA FZS 600. Finally, we have taken the weight of the YAMAHA FZS 600, and it is 2080 mm (2.080 meters)[8]. In addition, we have considered the height of a person driving a motorcyle and we have estimated is with a value of 1700 mm (1.70 meters).

Finally, the parameter values we have used in the pollution estimation module are those obtained from the Production of Updated Emission Curves for Use in the National Transport Model report, which is available in its website [9] and we have selected those configuration corresponding to the year 2020. In addition, according to these report, the emission curves are suitable for speed values between 5 and 120 kph. Figure 3 exhibits the considered emission curves for the petrol and diesel fuel vehicle type (y_p and y_d, respectively).

The value of all the parameters are shown in Table 1.

3.4. Results

In order to test the suitability of our developed approach, we have studied the obtained result from a qualitative and a quantitative point of view.

We have used the same *camara-guadalupe* sequence in the training SOM step and the pollution estimation process.

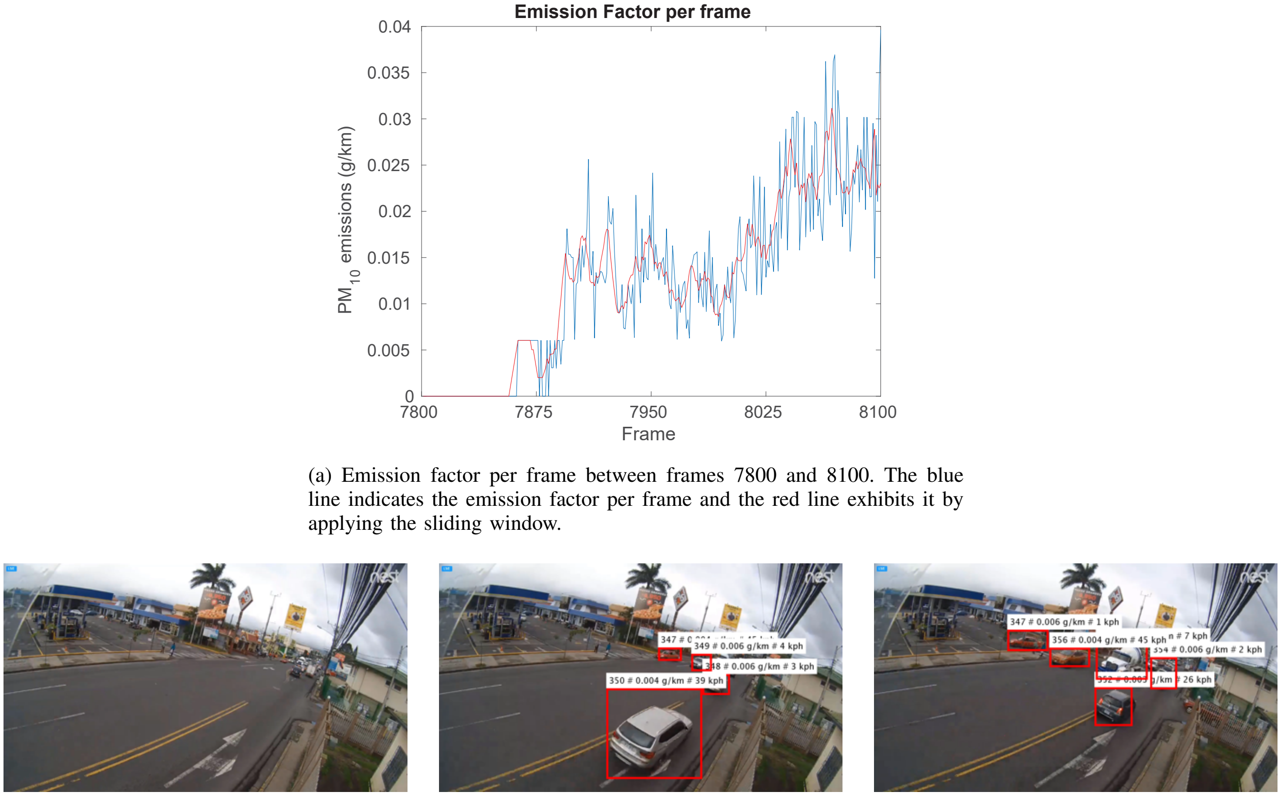
First of all, we have carried out the training SOM process in order to obtain the SOM model which will provide an estimation of the pixel-meter relationship in each area of the frame. The trained SOM and the training dataset can be observed in Figure 4.

After that, we have carried out our proposal in order to estimate the pollution in the selected video. From a qualitative point of view, the operation of our system can be observed in Figure 5. It shows some random frames (first column) and their corresponding output after the application of the Faster-RCNN network (second column) and Kalman model with the estimated pollution and speed (third column). As we can see, given a frame of the sequence as

[8] https://en.wikipedia.org/wiki/Yamaha_FZS600_Fazer

[9] https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/662795/updated-emission-curves-ntm.pdf

Figure 6. Emission factor per frame.



(a) Emission factor per frame between frames 7800 and 8100. The blue line indicates the emission factor per frame and the red line exhibits it by applying the sliding window.

(b) Frames 7800, 7948 and 8048 with the output information provided by the proposal: the trajectory of the detected vehicles and their speed and pollution estimations in these frames.

input, the output with the information obtained from the Faster-RCNN (second column) corresponds to the object detection and classification step, inside of the vehicle detection and tracking module, and it produces several undesired detections (red bounding box) and we only choose those corresponding to the *car* class (green bounding box). Finally, considering this information, the system produces the output corresponding to the input frame providing the information of the trajectory of each detected vehicle and its speed and pollution estimation (third column).

On the other hand, the quantitative analysis is shown in Figure 6 (a), which exhibits the emission factor per several selected frames of the chosen sequence. As it can be observed, our proposal estimates the pollution of each frame and with this information we can obtain those moments with a high or low traffic level, as we can see in Figure 6 (b). Another important point to be highlighted is the ups and downs of the estimation, which corresponds to the blue line from (a). This is because if the bounding box corresponding to the vehicle i returned as output by the Faster-RCNN network in the frame t is not practically the same than the previous frame $t-1$, this error produces a bad calculation of each centroid, and in most of cases both centroids will be far away, so the system will provide a high speed for this vehicle i. Thus, in order to avoid this ups and downs, we have employed a sliding window with a size of 5 frames to show the smoothed emission factor per frame, which is represented by a red line in (b).

4. Conclusion

This work has presented a methodology for estimating road pollution using static traffic cameras. Initially it was necessary to detect the vehicles present on the road (using the Faster RCNN network) to later estimate their speed and pollution level. Because practically all scenes have a perspective view, a self-organized neural model has been used to correct and homogenize the correspondence between physical distance and number of pixels in each region of the image.

Experiments show that there is a clear correlation between the pollution estimation on each frame and the number of vehicles displayed, as can be seen in Figure 6. These promising results allow us to make more extensive comparative studies with other existing techniques. It is also possible to study the feasibility of using this proposal combined with other types of sensors that increase its effectiveness.

Acknowledgments

This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grant TIN2014-

53465-R, project name Video surveillance by active search of anomalous events and TIN2016-75097-P. It is also partially supported by the Autonomous Government of Andalusia (Spain) under projects TIC-6213, project name Development of Self-Organizing Neural Networks for Information Technologies; and TIC-657, project name Self-organizing systems and robust estimators for video surveillance. All of them include funds from the European Regional Development Fund (ERDF). 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 used for this research. The authors would like to thank the grant of the Universidad de Málaga.

References

[1] S. Yang, Y.-J. Wu, and J. Woolschlager, “Integrated modeling framework for highway traffic pollution estimation and dispersion,” *American Journal of Environmental Sciences*, vol. 12, no. 3, pp. 140–151, 2016.

[2] 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,” *Atmospheric Environment*, vol. 104, pp. 264–272, 2015.

[3] O. V. Lozhkina and V. N. Lozhkin, “Estimation of road transport related air pollution in saint petersburg using european and russian calculation models,” *Transportation Research Part D: Transport and Environment*, vol. 36, pp. 178–189, 2015.

[4] “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Computer Science Review*, vol. 11-12, pp. 31 – 66, 2014.

[5] H. Xue, Y. Liu, D. Cai, and X. He, “Tracking people in rgbd videos using deep learning and motion clues,” *Neurocomputing*, vol. 204, pp. 70–76, 2016.

[6] Z. Zivkovic and F. Van Der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction,” *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.

[7] R. Luque, E. Domínguez, E. Palomo, and J. Muñoz, “An art-type network approach for video object detection,” 2010, pp. 423–428.

[8] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 5 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[11] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[13] Y. Bar-Shalom, *Tracking and Data Association*, 1987.

[14] R. Luque-Baena, E. López-Rubio, E. Domínguez, E. Palomo, and J. Jerez, “A self-organizing map to improve vehicle detection in flow monitoring systems,” *Soft Computing*, vol. 19, no. 9, pp. 2499–2509, 2015.