Neural network-based acoustic vehicle counting

1st Slobodan Djukanović
Faculty of Electrical Engineering
University of Montenegro
Podgorica, Montenegro
slobdj@ucg.ac.me

2nd Yash Patel
Faculty of Electrical Engineering
Czech Technical University
Prague, Czech Republic
patelyas@fel.cvut.cz

3rd Jiří Matas
Faculty of Electrical Engineering
Czech Technical University
Prague, Czech Republic
matas@fel.cvut.cz

4th Tuomas Virtanen
Faculty of Information Technology and Communication Sciences
Tampere University
Tampere, Finland
tuomas.virtanen@tuni.fi

***Abstract*—This paper addresses acoustic vehicle counting using one-channel audio. We predict the pass-by instants of vehicles from local minima of clipped vehicle-to-microphone distance. This distance is predicted from audio using a two-stage (coarse-fine) regression, with both stages realised via neural networks (NNs). Experiments show that the NN-based distance regression outperforms by far the previously proposed support vector regression. The 95% confidence interval for the mean of vehicle counting error is within $[0.28\%, -0.55\%]$. Besides the minima-based counting, we propose a deep learning counting that operates on the predicted distance without detecting local minima. Although outperformed in accuracy by the former approach, deep counting has a significant advantage in that it does not depend on minima detection parameters. Results also show that removing low frequencies in features improves the counting performance.**

***Index Terms*—Vehicle counting, log-mel spectrogram, neural network, peak detection, deep learning**

I. Introduction

Traffic monitoring (TM) systems use different traffic data to improve the use and performance of roadway systems, transportation safety, law enforcement, prediction of future transportation needs etc. TM data include estimates of vehicle count, traffic volume, speed of vehicles and of various vehicle parameters (length, weight, class) [1].

Current TM systems use diverse sensors and technologies, including induction loops, vibration, piezoelectric, infrared, ultrasonic, magnetic and acoustic sensors and cameras [1]. Vision-based TM systems have recently become popular due to breakthroughs in object detection, tracking and classification tasks provided by deep learning methods [2]. In addition, a single camera suffices to cover multiple lanes for the TM tasks, which is not the case for other sensor technologies. However, high computational complexity, partial occlusion, shadows and illumination variation limit the performance of vision-based TM systems [1], [3].

The research was supported by Research Center for Informatics (project CZ.02.1.01/0.0/0.0/16019/0000765 funded by OP VVV) and CTU student grant (SGS OHK3-019/20). Slobodan Djukanović was supported by the OP RDE programme of project International Mobility of Researchers MSCA-IF III at CTU in Prague No. CZ.02.2.69/0.0/0.0/19_074/0016255.

Acoustic TM has several advantages in comparison with the vision-based one [1]. Microphones are less expensive than cameras, consume less energy, require less storage space, are easier to install and maintain with low wear and tear. In addition, acoustic TM is not affected by visual occlusions and lighting conditions, and has less privacy issues.

This paper addresses acoustic vehicle counting using one-channel audio. The standard approach is to detect temporal variation of the sound power due to vehicles passing by microphone [4]–[6], which is performed using state transitions of a hidden Markov model [4] or by a peak-picking algorithm [5], [6]. Maximal frequency at which the power of a time-frequency representation reaches a predefined threshold, a.k.a. top-right frequency, also enables detecting vehicles passing by the microphone [7]. The method [8] uses a prediction of *clipped vehicle-to-microphone distance*, a form of pseudo-distance between a vehicle and the microphone. Vehicle counting [8], carried out by counting local minima in the predicted distance, outperforms those based on peak detection in the sound power and top-right frequency. However, the optimal, false negative-false positive compensating, detection threshold in [8] can only be imprecisely estimated *a priori*. Moreover, the distance regression [8] is computationally demanding.

In this paper, we significantly improve the distance regression, and thus vehicle counting accuracy, compared with [8] by a computationally less demanding approach. We first overview clipped vehicle-to-microphone distance (Section II) and then propose new counting method (Section III). Experimental results are given in Section IV, whereas Section V concludes the paper.

II. Clipped vehicle-to-microphone distance and vehicle counting

In [8], clipped vehicle-to-microphone distance[1] of the k-th vehicle is defined as

$$d^{(k)}(t) = \begin{cases} \left|t - t^{(k)}\right|, & \left|t - t^{(k)}\right| < T_D \\ T_D, & \text{elsewhere}, \end{cases} \tag{1}$$

[1]We will refer to clipped vehicle-to-microphone distance as the distance.

where $t^{(k)}$ represents the pass-by instant of the vehicle and T_D is the distance threshold. The V-shape of $d^{(k)}(t)$ models approaching and receding of the vehicle from the microphone (see dotted line around $t^{(1)}$ in Fig. 1). When the audio contains N_v vehicles, only the distance of the closest vehicle is taken into account, so the overall distance is defined as minimum of all separate distances (dotted line in Fig. 1):

$$D(t) = \min\{d^{(1)}(t), d^{(2)}(t), \dots, d^{(N_v)}(t)\}. \tag{2}$$

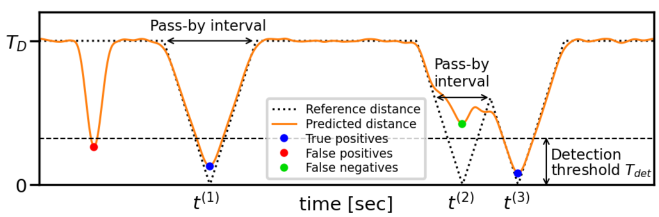

Fig. 1. Illustration of a reference distance, distance predicted from audio and classification of predicted distance minima.

In [8], the vehicle count is equal to the number of detected local minima of the predicted distance (orange line in Fig. 1) that fall below detection threshold. Not every local minimum below the threshold corresponds to a vehicle passing by the microphone. Only minima that occur within the true vehicle pass-by intervals (depicted by horizontal arrows in Fig. 1) represent true positives (TPs). Other minima below the threshold represent false positives (FPs), whereas minima that occur within the corresponding pass-by intervals, but are above the threshold, represent false negatives (FNs).

The method [8] is evaluated using the TP, FP and FN probabilities, p_{TP}, p_{FP} and p_{FN}, calculated for variable detection threshold. The optimal threshold corresponds to a point where $p_{\text{FP}} = p_{\text{FN}}$, since then the FPs and FNs cancel each other in statistical sense and the total number of detected vehicles equals the true number of vehicles. In terms of counting error, the best generalization in [8] is obtained when the distance is predicted using the log-mel spectrogram (LMS) and newly introduced high-frequency power (HFP) as input features. With HFP, the counting error remains low (below 2%) within a wide range of detection threshold values.

Although characterized by low counting error within a wide range of detection thresholds, the optimal threshold of [8] is not known in advance. Our first objective is to extend threshold range with low-error (below 2% or even more), i.e., to make counting more robust to the choice of detection threshold. Another drawback of [8] is the computational complexity. For distance regression, it uses support vector regression (SVR), implemented in the libsvm library [9]. Its complexity scales between $O(n_f n_s^2)$ and $O(n_f n_s^3)$, where n_f and n_s represent the number of features and samples in the dataset, respectively. Therefore, our second objective is to perform distance regression in a computationally more efficient way to enable scaling to larger datasets. A method which fulfills these two objectives is described in the sequel.

III. Neural network-based counting

To address the low-counting error objective, we propose to improve the accuracy of distance regression by a two-stage (coarse-fine) approach. To address the computational complexity objective, we propose to use fully-connected neural networks (NNs) instead of originally used SVR. The block diagram of the proposed method is presented in Fig. 2 (top). $x_i(t_0)$, $i = 1, \cdots, I$ are input features at time instant t_0 and $\check{D}(t_0)$ represents the predicted distance at t_0. The *Vehicle counting* block carries out counting based on the distance prediction of the whole audio file.

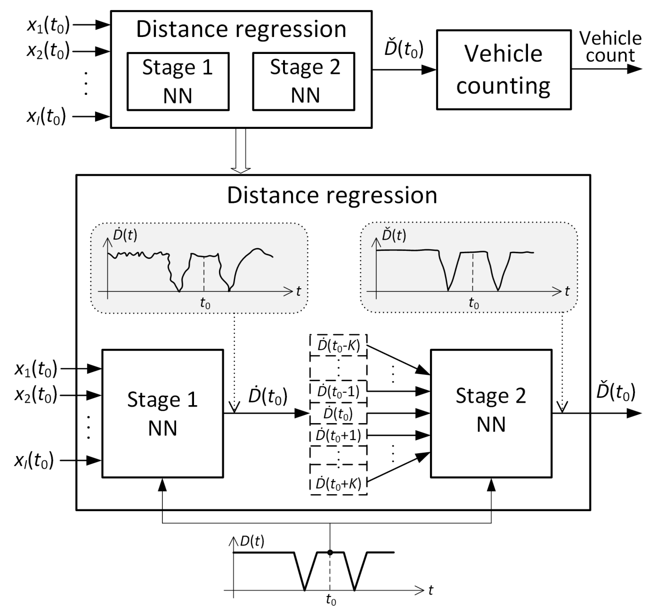

Fig. 2. *Top*: The block diagram of the proposed vehicle counting method. *Bottom*: Distance regression in detail. Stage 2 improves the distance regression output by Stage 1.

A. Distance regression and input features

A detailed representation of the proposed distance regression is given in Fig. 2 (bottom). Stage 1 NN performs regression based on input features $x_i(t_0)$, $i = 1, \cdots, I$, similarly to SVR in [8]. As input features, we propose to combine HFP and LMS, as suggested in [8]. Since HFP represents the power of high-frequency portion of the signal spectrum, we incorporate it into LMS by leaving out a number of filters with the lowest central frequencies in the mel spectrogram filter bank [10]. The resulting LMS, referred to as high-frequency LMS (HF-LMS), does not include low-frequency portion of the spectrum which contains the most significant part of the environment noise. To take into account time dependence between adjacent $D(t)$ values, value $D(t_0)$ will be predicted using the samples of the HF-LMS spectrum from time interval $[t_0 - Q, t_0 + Q]$. Therefore, the dimensionality of the input space is $I = (2Q + 1)N_{mel}$, where N_{mel} is the number of mel bands. For implementation details, see Section III-C.

The task of Stage 2 NN is to refine prediction $\dot{D}(t_0)$ of Stage 1 to obtain the final distance prediction $\check{D}(t_0)$. Stage 2 refines $\dot{D}(t_0)$ using a window of predicted distances centered at t_0, i.e., a vector of $2K+1$ successive predicted distances $\dot{D}(t_0-K), \ldots, \dot{D}(t_0+K)$, represents input features to Stage 2 NN.

Prior to vehicle counting, the predicted distance is low-pass filtered to eliminate high-frequency oscillations, which is discussed in Section IV.

B. Vehicle counting

In this paper, we propose two vehicle counting approaches, both based on the final (Stage 2) predicted distance.

1) Peak properties-based vehicle counting: In [8], local minima of the predicted distance (black dots in Fig. 3 (left)) were detected by detecting local maxima (peaks) of the inverted distance (black dots in Fig. 3 (right)) based on their prominence[2]. Here, we extend this approach by introducing a peak magnitude criterion. If two close peaks have similar magnitudes (corresponding to two close vehicles), the prominence of the weaker one can be much less than that of the stronger one. For example, consider the fourth peak with 0.46 magnitude and 0.2 prominence, shortly $(0.46, 0.2)$, in Fig. 3 (right). It clearly corresponds to a vehicle (see the corresponding reference and predicted distances in Fig. 3 (left)), but due to vicinity of the third peak $(0.74, 0.74)$ its prominence is even smaller than that of the second peak $(0.25, 0.21)$ which has almost two times smaller magnitude.

Since the weaker peak can be left out if it is detected based on prominence only, we define vehicle detection criterion as:

> *A vehicle is detected if detected peak of the inverted predicted distance has magnitude larger than M or prominence larger than P.*

Selection of M and P is discussed in Section IV.

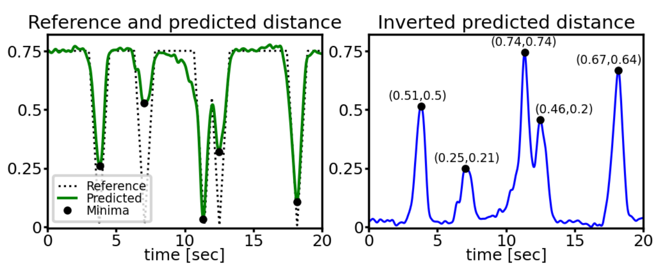

Fig. 3. *Left*: Reference and predicted distances of an audio file. *Right*: Inverted predicted distance is used for vehicle detection. Each detected peak (black dots) is characterized by a magnitude-prominence pair (e.g., first peak $(0.51, 0.5)$).

2) Deep vehicle counting: A convolutional NN [12] is used for counting. The model operates on the raw predicted distance and estimates the vehicle count directly, without an intermediate local minima detection. It consists of 1D convolutional, a global-average pooling and fully-connected layers. The global-average-pooling allows the model to operate on varying length distance [13].

The model is trained to predict the vehicle count directly. We experimented with three different loss functions: L_1, L_2 and smooth L_0 distances. The model trained with smooth L_0 distance gives the best performance. Training with smooth L_0 is performed using a surrogate learned via a deep embedding, where the Euclidean distance between the prediction and the ground truth corresponds to the L_0 distance [14]. The deep embedding is realized using a shallow fully-connected NN. The surrogate and the counting model are trained in parallel.

C. Implementation details

The HF-LMS feature is based on the spectrogram of the input signal. In this paper, the length of sliding (Hamming) window used in spectrogram calculation is $N_w = 4096$ and the stride length is $N_h = 1634$ samples [15], which with 20-second audio files sampled at $f_s = 44100$ Hz[3] gives the time-length of HF-LMS of 540 samples. In addition, $N_{mel} = 48$ mel bands are used in HF-LMS, with $f_{min} = 1000$ Hz and $f_{max} = fs/2 = 22050$ Hz. To form a vector of input features, we take the HF-LMS spectra at $Q = 5$ preceding and following instants with a stride of 2. Therefore, the dimensionality of the input space is $I = (2Q+1)N_{mel} = 528$. The input dimensionality of Stage 2 NN is $2K+1 = 31$. For distance threshold, we take $T_D = 0.75$ s [8]. The selected values of N_{mel}, Q, K and T_D are obtained via cross-validation.

Stage 1 and 2 NNs have four layers with 528-64-64-1 and 31-31-15-1 neurons per layer, respectively. These configurations gave the best regression performance cross-validated on the training data. Both NNs use mean squared error loss, ReLU activation, $L2$ kernel regularization with factors 10^{-4} (Stage 1) and 5×10^{-6} (Stage 2), 100 training epochs. Batch normalization is applied at each layer, except for the output, after activation. The model is implemented in Keras. Peak detection is based on `scipy.signal.find_peaks` procedure (SciPy library for Python).

IV. Experiments

First evaluation metric we use is *normalized area under the p_{TP} curve*, i.e., the average value of p_{TP} over detection threshold $T_{det} \in [0, T_D]$ [8]. p_{TP} is calculated as percentage of minima detected within the pass-by intervals (maximum one minimum per interval) at 100 equidistant T_{det} points.

The second metric is *relative vehicle counting error*

$$\mathrm{RVCE} = \frac{N_v^{true} - N_v^{est}}{N_v^{true}} \times 100\,[\%], \tag{3}$$

where N_v^{true} and N_v^{est} represent the true and estimated vehicle counts. As opposed to (3), the error definition in [8] uses the absolute difference. Here, the signed error enables distinguishing between underestimation (positive error) and overestimation (negative error) in counting.

[2]The prominence of a peak measures how much the peak stands out due to its height and location relative to other peaks, and is defined as the vertical distance between the peak and its lowest contour line [11].

[3]The spectrogram of audio file presented in Fig. 3 in [8] indicates the presence of high-frequency components (around 14 kHz). Therefore, the original sampling rate is retained.

We will use the dataset from [8][4] which contains two parts: VC-PRG-1:5 (250 20-second audio files with 841 vehicles) and VC-PRG-6 (172 audio files with 580 vehicles). Five vehicle classes are considered in the dataset: motorcycles, cars, vans, buses and trucks. No assumptions are made regarding the vehicles' speed, nor it is estimated.

The proposed vehicle counting method (hereafter referred to as VCNN) will be trained and validated (80%-20% training-validation split) using VC-PRG-1:5[5] and tested on VC-PRG-6. We run the method 40 times (training data shuffled each time). Along with Stage 2, we also consider the output of Stage 1 NN and the corresponding vehicle count VCNN$_{S1}$.

Minima detection, and therefore probabilities p_{TP}, p_{FP} and p_{FN}, as well as RVCE, are affected by i) low-pass filtering of the predicted distance, ii) value of M and iii) value of P (Section III-B1). To optimize the count performance with respect to these factors, we carry out the grid search algorithm. We consider all possible combinations of i) low-pass filters (successive moving average (MA) filters with lengths $(5,3)$[6], $(7,3)$ and $(7,5,3)$), ii) $M \in \{35\%\,T_D, 40\%\,T_D, 45\%\,T_D, 50\%\,T_D\}$ and iii) $P \in \{10\%\,T_D, 15\%\,T_D, 20\%\,T_D, 25\%\,T_D\}$. The optimality criterion is averaged absolute RVCE for detection threshold range $T_{det} \in [50\%\,T_D, T_D]$. Table I presents the optimal setups for VCNN$_{S1}$ and VCNN (first two rows).

TABLE I
OPTIMAL MINIMA (VEHICLE) DETECTION PARAMETERS

Setup	Optimal parameters
VCNN$_{S1}$	MA filters $(7,3)$, $M = 45\%\,T_D$, $P = 25\%\,T_D$
VCNN	MA filters $(5,3)$, $M = 40\%\,T_D$, $P = 20\%\,T_D$
VCNN$_{f0}$	MA filters $(5,3)$, $M = 45\%\,T_D$, $P = 20\%\,T_D$

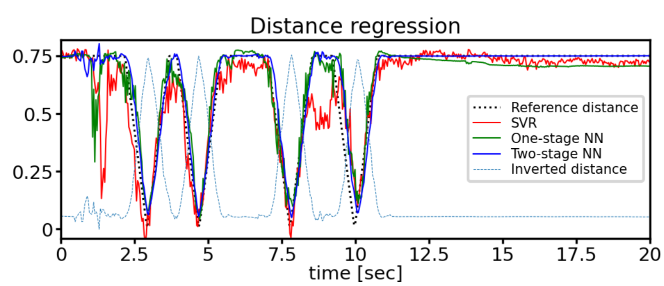

Fig. 4. Distance predictions of an audio file. Minima are detected by detecting peaks of the inverted predicted distance.

Before evaluating the vehicle counting performance, let's analyze improvement in distance regression offered by the proposed approach. Figure 4 compares distance predictions of one audio file carried out via SVR, one-stage NN and two-stage NN. The two-stage approach (blue solid line) follows the reference distance (dotted line) the most faithfully. The mean square regression errors for the testing set are 9.55×10^{-3} (SVR), 5.27×10^{-3} (one-stage NN) and 2.30×10^{-3} (two-stage NN). We conclude that i) the proposed method significantly outperforms SVR in terms of regression accuracy, and ii) Stage 2 improves the regression accuracy with respect to Stage 1.

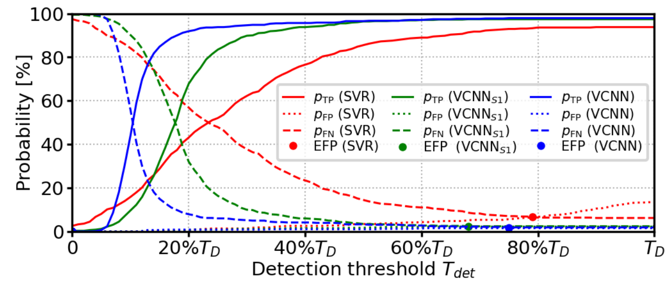

Fig. 5. TP, FP and FN probabilities with equal false probabilities (EFP) points. The lower EFP the better.

Figure 5 compares probabilities p_{TP}, p_{FP} and p_{FN} of the SVR-based counting [8] and of one run of VCNN$_{S1}$ and VCNN. Low T_{det} results in low p_{TP} and high p_{FN} (see detection threshold in Fig. 1). On the other hand, with high T_{det}, majority of detected local minima surpass the threshold, resulting in high p_{TP} and low p_{FN}. Here, we notice notably higher p_{FP} (dotted curves in Fig. 5) in the SVR-based counting. Significant improvement of the proposed approach is reflected in an increase of normalized area under the p_{TP} curve from 0.695 (SVR) to 0.759 (VCNN$_{S1}$) and 0.850 (VCNN), the latter two averaged over all runs. Dots in Fig. 5 represent the points of equal false probabilities (EFPs), which are 6.55%, 2.33% and 1.72%, respectively. The lower EFP the better [8].

The error plots for detection threshold $T_{det} \in [30\%\,T_D, T_D]$ are given in Fig. 6 (top). With the exception of SVR, RVCEs are shown as 95% confidence intervals for the mean (referred to as confidence). In addition to vehicle counting based on SVR, VCNN$_{S1}$ and VCNN, we present RVCE of the deep counting approach (VCNN$_{deep}$) and of VCNN when full-frequency range is used in the LMS input features ($f_{min} = 0$, see Section III-C). The latter is denoted as VCNN$_{f0}$ and its optimal parameters are given in the bottom row in Table I. Confidence of VCNN$_{deep}$ is a horizontal band within $[1.25\%, 2.49\%]$. The deep counting does not depend on minima detection parameters and hence its performance does not depend on T_{det}. The proposed VCNN approach outperforms the other ones by a significant margin. Its 95% confidence goes below 2% for $T_{det} \approx 50\%\,T_D$ and is bounded within $[0.28\%, -0.55\%]$ for $T_{det} > 70\%\,T_D$. In other words, the counting error is less then 2% for $T_{det} > 50\%\,T_D$, which makes our low-error objective fulfilled.

The combined peak magnitude-prominence criterion in vehicle detection (Section III-B1) improves the counting performance with respect to detection based solely on the peak prominence [8], as illustrated in Fig. 6 (bottom). Peak prominences of 5%, 10% and 15% are considered. Prior to peak detection, the predicted distance has been low-pass filtered

[4] The dataset and Python implementation of [8] are available for download at http://cmp.felk.cvut.cz/data/audio_vc.

[5] With the setup given in Section III-C, the number of features and samples in the training/validation dataset are $n_f = 528$ and $n_s = 250 \times 540 = 135000$, respectively. Since the SVR complexity scales between $O(n_f n_s^2)$ and $O(n_f n_s^3)$ [9], we have decided to implement NN-based regression.

[6] (l_1, l_2) denotes filtering first with an MA filter of length l_1 then with one of length l_2.

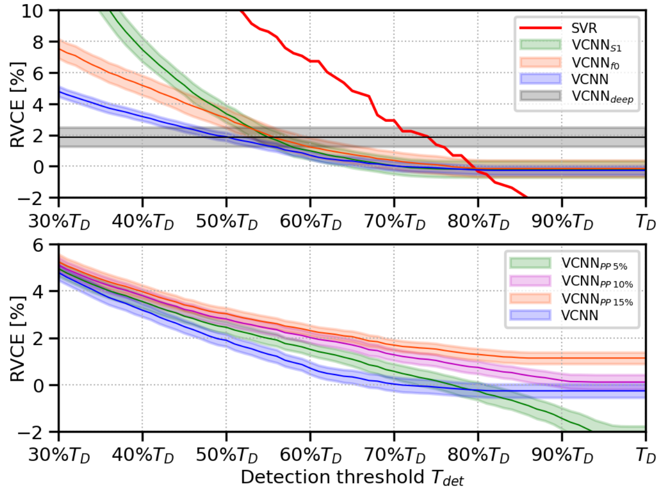

Fig. 6. Relative vehicle counting error, RVCE, as a function of the detection threshold. With the exception of SVR, RVCEs are shown as 95% confidence intervals for the mean. The proposed VCNN (in blue) compared with alternatives (for their description, see text). VCNN$_{PP\,\#\%}$ in the bottom plot correspond to peak detection based only on peak prominence.

with MA filters $(5, 3)$. Introducing magnitude in peak detection is crucial for achieving stable accurate results within a wide range of detection threshold.

V. CONCLUSIONS

We proposed a method for acoustic vehicle counting based on the clipped vehicle-to-microphone distance. The distance was predicted using a two-stage NN-based regression. Significant improvement in regression accuracy with respect to the SVR-based approach resulted in a highly accurate vehicle counting not depending on detection threshold within a wide range of threshold values. Deep counting, an alternative to the local minima-based counting, estimates the vehicle count directly from the predicted distance, without detecting local minima. Although outperformed in accuracy by the latter approach, a significant advantage of deep counting is that it does not depend on minima detection parameters. Our future work will address developing end-to-end vehicle counting method.

REFERENCES

[1] Myounggyu Won, "Intelligent traffic monitoring systems for vehicle classification: A survey," *IEEE Access*, vol. 8, pp. 73340–73358, 2020.

[2] Milind Naphade et al., "The 2019 AI City challenge," in *CVPR Workshops*, 2019, pp. 452–460.

[3] Brendan Tran Morris and Mohan Manubhai Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, 2008.

[4] Jien Kato, "An attempt to acquire traffic density by using road traffic sound," in *Proceedings of the 2005 International Conference on Active Media Technology, 2005.(AMT 2005).* IEEE, 2005, pp. 353–358.

[5] Jobin George, Leena Mary, and KS Riyas, "Vehicle detection and classification from acoustic signal using ANN and KNN," in *2013 international conference on control communication and computing (ICCC)*. IEEE, 2013, pp. 436–439.

[6] Jobin George, Anila Cyril, Bino I Koshy, and Leena Mary, "Exploring sound signature for vehicle detection and classification using ANN," *International Journal on Soft Computing*, vol. 4, no. 2, pp. 29, 2013.

[7] Sugang Li, Xiaoran Fan, Yanyong Zhang, Wade Trappe, Janne Lindqvist, and Richard E Howard, "Auto++: Detecting cars using embedded microphones in real-time," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 70, 2017.

[8] Slobodan Djukanović, Jiří Matas, and Tuomas Virtanen, "Robust audio-based vehicle counting in low-to-moderate traffic flow," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1608–1614.

[9] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[10] Romain Serizel, Victor Bisot, Slim Essid, and Gaël Richard, "Acoustic features for environmental sound analysis," in *Computational Analysis of Sound Scenes and Events*, pp. 71–101. Springer, 2018.

[11] "Topographic prominence," https://en.wikipedia.org/wiki/Topographic_prominence, Accessed: 2020-10-19.

[12] Xiang Zhang, Junbo Zhao, and Yann LeCun, "Character-level convolutional networks for text classification," in *NeurIPS*, 2015.

[13] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," *ICLR*, 2014.

[14] Yash Patel, Tomáš Hodaň, and Jiří Matas, "Learning surrogates via deep embedding," in *European Conference on Computer Vision*. Springer, 2020, pp. 205–221.

[15] Ljubiša Stanković, Igor Djurović, Srdjan Stanković, Marko Simeunović, Slobodan Djukanović, and Miloš Daković, "Instantaneous frequency in time–frequency analysis: Enhanced concepts and performance of estimation algorithms," *Digital Signal Processing*, vol. 35, pp. 1–13, 2014.