Sensor applications

Traffic Intensity Detection Using General-Purpose Sensing

Aung Kaung Myat[1*], Roberto Minerva[1**], Attaphongse Taparugssanagorn[2**], Praboda Rajapaksha[1**], and Noel Crespi[1**]

[1] *Samovar, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France*
[2] *Internet of Things System Engineering, Asian Institute of Technology, Pathum Thani 12120, Thailand*
[*] *Member, IEEE*
[**] *Senior Member, IEEE*

Manuscript received 30 June 2023; revised 7 September 2023; accepted 10 September 2023. Date of publication 14 September 2023; date of current version 10 October 2023.

Abstract—The conventional approaches for traffic intensity (TI) detection in smart cities use specialized sensors, such as loop detectors, cameras, and radar. These sensors come with high costs, limited reuse, and specialized maintenance. Thus, we propose the utilization of general-purpose sensing, which involves the use of cost-effective and easily deployable sensors, such as microphones, and air quality sensors that are cheaper, reusable, and easily manageable sensors. A general-purpose sensing infrastructure can be used for several applications including measuring TI. The main objective of this letter is to demonstrate how noise signatures can be leveraged to measure TI. Traditionally, image classification techniques are used for audio data analysis and vision-transformers (ViT) have shown remarkable results in this area. However, there is limited research that explores ViT models for traffic intensity detection. For TI detection, two approaches: vehicle count prediction and vehicle type detection (VTD), which use ViT models, are proposed. VTD approach performed better and it achieved an F1-score of 0.95 and a mean vehicle counting error of 0.032. In addition, the computational complexity of the approach was evaluated by implementing an edge solution. The proposed approaches can be effectively used even on resource-limited edge devices, with a notable increase in detection time.

Index Terms—Sensor applications, acoustic, audio, general-purpose sensing, Internet-of-Things (IoT), traffic detection (TI), vision-transformers (ViT).

I. INTRODUCTION

Among the many challenges of a smart city, traffic congestion is a significant problem. Intelligent transportation systems aim to optimize traffic flow, reduce congestion, and lower greenhouse gas emissions. In the literature, specialized traffic sensors [1], such as video cameras, radars, and other "reusable" sensors are used to measure traffic intensities (TI). These specialized sensors are expensive and require specialized maintenance, and often they are used for a single purpose. Moreover, additional sensors are used for monitoring other city phenomena, e.g., noise and air pollution. There is an opportunity to exploit reusable sensing capabilities to avoid building "sensing silos" where specialized sensing is used for each specific purpose. General-purpose sensing [2], [3] is an attempt to use low-cost, easy-to-deploy, and reusable sensors, such as microphones, cameras, or air quality sensors, to build cheaper, nonintrusive, easily manageable, and reliable measuring infrastructure that enables sensing of several phenomena. This letter aims to show how to use noise detection capabilities to infer traffic intensity levels.

To build an effective general-purpose sensing platform, computing and AI capabilities are essential. Computing is an enabler for AI tools and techniques to be applied to sensed and "fused" data. The transformation of these raw data into usable information can be seen as synthetic sensing [2]. Data gathered from general-purpose sensors must undergo analysis using machine learning (ML) techniques. However, these detection methods often demand substantial computing and memory resources, sometimes requiring dedicated graphics processing units. Moreover, although video can be used for monitoring traffic intensity, it raises privacy concerns. The usage of audio data (noise levels of nearby traffic) can be a less privacy-invasive alternative. In addition, microphones are cheaper than cameras and could be easier to deploy (considering present regulations for videos in many countries).

Image classification techniques have been widely used for audio analysis but there has been limited research exploring the use of vision transformers (ViT) for audio analysis. Gong et al. [4] proposed a knowledge transfer approach from ViT and evaluated their models on various audio benchmark datasets, yielding promising performance. However, to the best of the author's knowledge, no research has specifically investigated the use of ViT for audio-based traffic monitoring.

The objective of this article is to investigate the viability of an edge-computing traffic monitoring system utilizing microphones and general-purpose sensors to achieve sufficient efficiency and accuracy in traffic monitoring in urban areas. By leveraging these sensors and employing edge computing techniques, the proposed system aims to provide an alternative to the use of specialized single-purpose sensors, breaking the silos approach and pushing synthetic sensing [2] a step further.

II. LITERATURE REVIEW

In a previous study, we used air pollutants and atmospheric data in conjunction with available traffic data to improve prediction of TI [5], and proved that additional features helped to achieve better model performances. Gatto and Forster [6], proposed an audio-based classification method to determine the traffic intensity into noncongestion and congestion by using a random forest classifier. Compared with video

Corresponding author: Aung Kaung Myat (e-mail: aung-kaung.myat@telecom-sudparis.eu).
Associate Editor: R. Vida.
Digital Object Identifier 10.1109/LSENS.2023.3315251

Table 1. Edge Device Components List for RLD and PED

Components Type	RLD	PED
Computer	Raspberry Pi Model B	Nvidia Jetson Nano
Microphones	IM69D130	AOS3729A-T42
Environmental sensor	BME280	BME280
Light sensor	LTR-559	LTR-559
Gas sensor	MIC6814	MIC6814
PM sensor	PMS5003	PMS5003

data, noise data is less privacy invasive, and many cities, including Madrid [7] and Dublin [8], have already collected these data for noise pollution evaluation. ML techniques can be applied to these noise data for TI monitoring. Slobodan et al. [9] proposed a vehicle counting method using neural networks and achieved a mean vehicle counting error within the range of 0.28%–0.55% [10]. Inspired by this study, we adopt a different strategy to predict vehicle count from the audio feature.

Jakob et al. [11] researched vehicle type classification and direction estimation using VGGNet, ResNet, and SqueezeNet models on two-s audio samples. Similarly, Yermakov [12] employed ResNet variants for vehicle detection and counting, annotating the audio samples with video recordings. In our vehicle detection approach, inspired by Yermakov's work [12], we combine vehicle type detection (VTD) and a nonoverlapping time window approach to count the number of passing vehicles. Our approach enables the prediction of four vehicle types and the determination of the total count of passing vehicles. We utilize an object detection model for annotating our audio data.

III. PROPOSED TI DETECTION METHODS

Assumptions: This letter focuses on 1-D two-lane roads considering all vehicles. Regarding the speed of the vehicles, no assumptions are made. The captured audio contains a mixture of vehicle sounds and environmental noises, including sounds from nearby roads, weather conditions, and reflected sound from the vehicles to be counted.

Utilized Edge Devices: Two types of edge devices are utilized for TI detection: *Resource-Limited device (RLD)* and *Powerful Edge device (PED)*. Both devices are equipped with micro-electromechanical systems (MEMS) microphones for sound recording. In addition, various sensors including temperature, pressure, gas, light intensity, and air quality sensors are used to capture the environmental road conditions. Many of these features are used for a bottom-up digital twin creation of the road and visualization purposes. More details about the components in each edge device are given in Table 1.

A. TI Detection System Architecture

We propose to utilize an edge-cloud architecture, with sensors and edge computer in the edge layer for traffic detection onsite while a message broker, flow controller, and databases on the cloud for visualization and digital twin creation. The system architecture and components included are shown in Fig. 1. The system can be divided into three modules.

The training module (gray area in Fig. 1) encompasses three key processes: audio dataset creation (Section III-B); feature extraction (Section III-C); and AI model training (Section III-D). This module prepares AI models for real-time TI detection on edge devices.

The inferencing module (red area in Fig. 1) on edge devices handles real-time TI detection and road environment monitoring. It utilizes the pretrained model from the training module to predict vehicle counts and gather sensor data. This information is subsequently transmitted to the cloud layer via message queuing telemetry transport (MQTT).

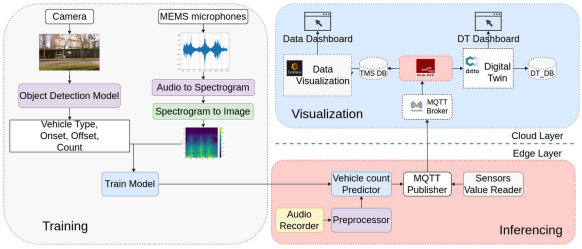

Fig. 1. Traffic intensity detection system comprised of three modules: *Training* (to train detection model), *Inferencing* (to do TI detection in real-time on the road), and *Visualization* (to visualize the traffic intensity and road condition).

Table 2. Numbers of Sample According to Vehicle Counts and Numbers of Vehicles According to Vehicle Types

Vehicle Counts	No. of Samples	Vehicle Types	No. of Vehicles
0	709	car	504
1	220	bus	20
2	85	truck	10
3	30	motorcycle	19
4	12	Total	553
5	5		
Total	1061		

The visualization module (blue section in Fig. 1) handles data visualization and digital twin updates. MQTT messages from edge devices are managed by the Node-RED flow controller, which stores data in databases and keeps the digital twin updated. Grafana is used for data visualization.

B. Dataset

An audio dataset for VTD and vehicle counting is created as the dataset comprising both tasks is not available.

Dataset Description: The dataset is collected at Boulevard Coquibus, Evry, France for 3 h. This location was chosen as it has a one-direction two-lane road and is one of the busy roads in Evry. The edge devices are set up on the sidewalk of the road, maintaining a distance of 1 m, to capture both audio and video. The audio was recorded using two MEMS microphones with a sampling rate of 48 kHz and two channels, while the video was captured at 640 × 480 resolution with 30 frames per second. The recorded audio and video pair is split into 10 s samples. In total, the dataset consists of 1061 samples encompassing recordings of 553 passing vehicles.

Audio Samples Annotation: To train the model, we need to annotate the audio data with timestamps for vehicle passages and their types. These annotations cannot be directly extracted from the audio. Instead, we leverage concurrent video data, which is recorded alongside the audio. We employ YOLOv8, an object detection model, to analyze the video, identifying passing vehicles (cars, buses, trucks, and motorcycles), and tracking them to determine their entry and exit times, as well as total counts. This YOLO model output, containing vehicle type, entry and exit times, and vehicle count, serves as the annotation for the audio samples. The details of the datasets are given in Table 2.

C. Preprocessing and Feature Extraction

In this section, how audio data are preprocessed, and how features are extracted and prepared for training AI models will be discussed. These preprocessing and feature extraction tasks are done on edge devices.

Preprocessing: Each audio sample has two channels as two microphones are used. Initially, the average values of two audio channels

are extracted. Subsequently, trimming or zero-padding is applied to achieve the desired size. This process ensures that each audio sample has a consistent size.

Audio to Spectrogram: Spectrograms, like the log-Mel spectrogram (LMS), are commonly employed for audio analysis. In LMS, the y-axis depicts frequency, the x-axis represents time, and colors signify amplitude in decibels. When a vehicle passes the microphone, it appears as a bright spot in the LMS. To extract LMS, first, short-time Fourier transform (STFT) (1) is applied to the audio data. This is a widely used technique for analyzing the time-varying frequency content of audio signals to obtain information about the spectral characteristics of the audio data at different time intervals. A sampling rate of 48 kHz, frame length of 2048, and hop length of 1024, which means 50% overlap between frames, is employed to extract spectrogram using STFT as follows:

$$X(t,\omega) = \int_{-\infty}^{\infty} x(\tau) w(\tau - t) e^{-j\omega\tau}\, d\tau. \tag{1}$$

Then, the spectrogram is converted into the Mel scale using (2) because it is a representation that better aligns with human auditory perception. To account for the human perception of loudness, its values are logarithmically converted to the decibel scale. Subsequently, a background noise removal algorithm is applied to eliminate environmental noise. Finally, the LMS feature with reduced background noise is obtained from the audio sample as follows:

$$\text{Mel}(f) = 2596 \log_{10}\left(1 + \frac{f}{700}\right). \tag{2}$$

Spectrogram to Image: As we want to leverage the capabilities of image classification techniques for audio analysis, LMS values are converted into color image(s). This conversion to image process differs depending on the proposed TI detection methods. For the first approach, LMS values are directly converted into color images with 224×224 pixels resolution. For the second approach, LMS values are divided into smaller chunks along the time axis using a nonoverlapping time window. Afterward, each chunk is converted into a color image with 224×224 pixels resolutions. To find the optimal window size, 1, 1.5, 2, 2.5, and 3 s window sizes are tested.

D. Proposed Methods

Image classification techniques have been widely used in audio classification and event detection tasks where ViT models have shown remarkable capabilities. Nevertheless, the extensive parameter count of ViT models hinders applicability on resource-constrained devices, such as edge devices. In this article, light-weight ViT-based models, such as TinyViT [14], EdgeViT [15], and MobileViT [16] are tested for audio classification and event detection while compared with baseline models, such as MobileNetV3 [17] and ResNet [18]. Two approaches for audio-based traffic intensity detection will be presented in the following. AI models for both approaches are trained on the cloud where computing resources are abundant.

Vehicle Count Prediction (VCP): In VCP, the output layer of the models mentioned earlier is replaced with a sequential layer containing fully connected and activation layers, culminating in a final layer with six output units. These six units correspond to vehicle counts: 0, 1, 2, 3, 4, and 5, aligning with the observed data that a maximum of five vehicles can pass through the microphones in a 10 s window. In this approach, the single-color LMS image extracted from the audio sample serves as input, yielding the vehicle count as output.

VTD: In VTD, the output layer is replaced by a sequential layer comprising fully connected and activation layers. The four output units correspond to vehicle types (car, bus, truck, and motorcycle). Differing from the VCP approach, this model does not directly predict the vehicle count. Instead, it divides the 10 s audio into smaller chunks, detecting the vehicle type in each chunk. The total count is derived by summing the vehicle counts from these chunks. The VTD approach takes images extracted from LMS, segmented based on the window size, as input. The model processes these images, providing both vehicle type and count as output.

Training Parameters and Evaluation Metrics: The models are learned using the Adam optimizer with a learning rate of 0.0001. Due to the dataset's imbalance, a weighted random sampler is employed for training data sampling. Cross-entropy is used as the loss function which is defined as follows:

$$L_{\text{CE}}(p, y) = -\sum_{c=1}^{C} y_i \log p_i \tag{3}$$

where C is the number of classes, y_i equals 1 if the sample belongs to class i and 0 otherwise, and p_i are probabilities for each class given by the softmax activation function. We used two metrics to evaluate the performance of each model: F1 score and relative vehicle counting error (RVCE) [9] which is given as follows:

$$\text{RVCE}(v) = \frac{|n_v^{\text{true}} - n_v^{\text{pred}}|}{n_v^{\text{true}}} \tag{4}$$

where n^{true} and n^{pred} represent the true and predicted number of vehicles in 10 s audio data. A higher F1 score indicates better performance, while a lower RVCE error reflects greater accuracy.

TI Calculation: The traffic intensity is derived from the vehicle count considering a threshold value (five in this research) as given in (5), where x is the number of passing by vehicles and TI is the traffic intensity as follows:

$$\text{TI} = \begin{cases} \text{High} & \text{if } x \geq 5 \\ \text{Medium} & \text{if } 2 \leq x \leq 5 \\ \text{Low} & \text{if } x \leq 2. \end{cases} \tag{5}$$

IV. EXPERIMENTAL RESULTS

The proposed two approaches are compared in terms of F1-score, RVCE, and computation time. Totally, 14 models including different variants of ViT models and baseline models are evaluated. In addition, edge-centric or cloud-centric deployment options are also compared by measuring bandwidth requirements.

A. Evaluation on VCP Approach

Among the models, the EdgeViT(size s) model achieves the highest F1-score of 0.767 and the lowest RVCE of 0.185, as can be seen in Fig. 2. In TinyViT and ResNet models, the performance of the model decreases as the model size becomes larger, which indicates overfitting. On the contrary, the models' performance improves as the size increases in EdgeViT, MobileViT, and MobileNetV3. In this approach, the transformer-based model outperforms traditional models, such as MobileNetV3 and ResNet, which achieve F1-scores of 0.74 and 0.75 and RVCE errors of 0.42 and 0.22.

B. Evaluation on VTD Approach

Among the different window sizes tested, the window size of 1 s achieves the highest F1-score and lowest RVCE compared with other window sizes. A 1 s window size is sufficient to effectively detect the

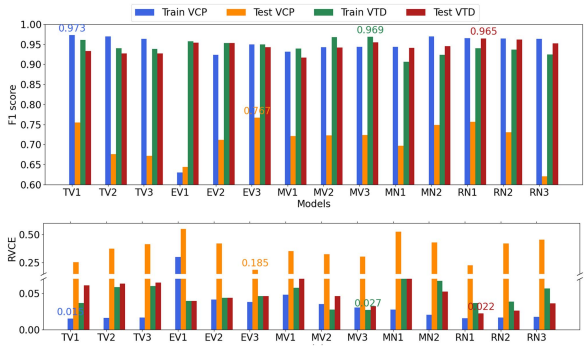

Fig. 2. This figure depicts the performances of both TI detection methods. TV represents TinyViT, EV stands for EdgeViT, MV represents MobileViT, and 1,2,3 in these models denotes xxs, xs, and s model sizes, respectively. RN stands for ResNet and 1,2,3 indicates 18, 34, and 50 layers in RN, respectively. MN refers to MobileNetV3, with 1 and 2 representing small and large model sizes, respectively. The highest F1-score on train and test datasets are annotated. Similarly, the lowest RVCE values are annotated.

vehicle presence and classify them into different categories. Thus, 1 s is chosen as the window size to train models. As shown in Fig. 2, ResNet(18 layers) achieved the lowest F1-score of 0.965 and RVCE of 0.022 on the testing set outperforming the ViTs. Among the ViTs, MobileViT (size s) achieves a comparable result, in which the F1-score is 0.955 and RVCE is 0.032. In the dataset, there are only 553 types of vehicles which means the sample size to train models is very small. ViTs tend to overfit on small datasets, which could be the reason ResNet outperforms ViTs.

C. Evaluation of Computation Time on Edge Devices

Both proposed approaches perform faster on PED with VCP approach taking 1.47 s and the VTD approach taking 2.53 s. However, on RLD, it takes 4.96 and 3.12 s, respectively. The memory of RLD is not sufficient to process the whole 10 s audio feature, which causes memory swapping operations that lead to more computation time in the VCP approach while the VTD audio feature is split into 1 s chunks, which are small enough to process on RLD memory. ResNet variant models exhibit significantly longer prediction and detection times on the resource-limited edge device (over 37.82 and 9.64 s, respectively), with minimal impact on the powerful edge device. Both the proposed approaches can be deployed on RLD with a marginal increase in detection time.

Furthermore, the bandwidth requirements for cloud-centric and edge-centric deployment are compared. To ensure real-time performance, the transfer time must be under 100 ms. The cloud-centric approach necessitates 148 Mb/s to transfer 1.8 MB of audio data every 10 s, whereas the edge-centric approach only requires 0.0244 Mb/s to transfer a 338-byte MQTT message. Consequently, the edge-centric approach demands less bandwidth, resulting in lower power consumption by edge devices.

D. Practical Challenges

Superposition: When two vehicles pass simultaneously, the combined noise is higher than a single vehicle, which enables the model to distinguish. However, it struggles when one of them is a low-noise vehicle like an electric car.

Recording hardware quality: Depending on the recording device quality, the characteristics of the audio data captured can differ which poses a problem when training the models.

V. CONCLUSION AND FUTURE WORK

In this letter, we have tackled vehicle detection and counting tasks with general-purpose sensing focusing on audio data. The VCP approach and VTD approach are explored. The VTD approach performs better than the VCP approach and achieves an F1-score of 0.955 and an RVCE of 0.032. Powerful-edge devices reduce computation time for detection and prediction. Even low-cost, resource-limited devices can achieve acceptable performance with slightly longer computation time. Finally, the edge-centric approach requires less bandwidth for transmission and lower power dissipation which are important factors for edge devices.

In future work, we aim to enhance our model capabilities by adding vehicle direction detection and detecting crucial audio events like emergency sirens and accidents. This will enable us to compile an audio-signature dataset for further refinement. In addition, we plan to leverage historical audio event data to simulate and predict potential traffic congestion and road blockages using digital twin technology. These capabilities will be used to incrementally build a digital twin representation of the road, comprising the abilities to observe, represent, and classify anomalies. In this way, we will use general-purpose sensing to support situation awareness experiments through a digital twin representation.

REFERENCES

[1] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo1, "Sensor technologies for intelligent transportation systems," *Sensors J.*, vol. 18, 2018, Art. no. 1212.
[2] G. Laput, Y. Zhang, and C. Harrison, "Synthetic sensors: Towards general-purpose sensing," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 3986–3999.
[3] R. Minerva, F. M. Awan, and N. Crespi, "Exploiting digital twins as enablers for synthetic sensing," *IEEE Internet Comput.*, vol. 26, no. 5, pp. 61–67, Sep.-Oct.– 2022.
[4] Y. Gong, Y. A. Chung, and J. R. Glass, "AST: Audio spectrogram transformer," in *Proc. Interspeech*, 2021, pp. 571–575.
[5] F. Awan, R. Malik Minerva, and N. Crespi, "Improving road traffic forecasting using air pollution and atmospheric data: Experiments based on LSTM recurrent neural networks," *Sensors J.*, vol. 20, no. 13, 2020, Art. no. 3749.
[6] R. C. Gatto and C. H. Q. Forster, "Audio-based machine learning model for traffic congestion detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7200–7207, Nov. 2021.
[7] A. Recio, C. Linares, J. R. Banegas, and J. Díaz, "Impact of road traffic noise on cause-specific mortality in Madrid (Spain)," *Sci. Total Environ.*, vol. 590, pp. 171–173, 2017.
[8] B. Basu et al., "Effect of COVID-19 on noise pollution change in Dublin, Ireland," 2020, *arXiv:2008.08993*.
[9] S. Djukanovic, J. Matas, and T. Virtanen, "Robust audio-based vehicle counting in low-to-moderate traffic flow," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, 2020, pp. 1337–1343.
[10] S. Djukanovic, Y. Patel, J. Matas, and T. Virtanen, "Neural network-based acoustic vehicle counting," in *Proc. 29th Eur. Signal Process. Conf.*, 2021, pp. 561–565.
[11] J. Abeßer, S. Gourishetti, A. Kátai, T. Clauß, P. Sharma, and J. Liebetrau, "IDMT-Traffic: An open benchmark dataset for acoustic traffic monitoring research," in *Proc. 29th Eur. Signal Process. Conf.*, 2021, pp. 551–555.
[12] B. Andrii Yermakov, "Audio-based vehicle recognition," M.S. thesis, Dept. Compute. Sci., Czech Tech. Univ., Prague, 2022.
[13] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," 2023, *arXiv:2305.09972*.
[14] K. Wu et al., "TinyViT: Fast pretraining distillation for small vision transformers," 2022.
[15] J. Pan et al., "EdgeViTs: Competing light-weight CNNs on mobile devices with vision transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2022.
[16] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.
[17] M. Chen et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.
[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.