FREQUENCY TRACKING FEATURES FOR DATA-EFFICIENT DEEP SIREN IDENTIFICATION

Stefano Damiano[1]*, *Thomas Dietzen*[1], *Toon van Waterschoot*[1]*,

[1] KU Leuven, Dept. of Electrical Engineering (ESAT-STADIUS), Leuven, Belgium,
{stefano.damiano, thomas.dietzen, toon.vanwaterschoot}@esat.kuleuven.be

ABSTRACT

The identification of siren sounds in urban soundscapes is a crucial safety aspect for smart vehicles and has been widely addressed by means of neural networks that ensure robustness to both the diversity of siren signals and the strong and unstructured background noise characterizing traffic. Convolutional neural networks analyzing spectrogram features of incoming signals achieve state-of-the-art performance when enough training data capturing the diversity of the target acoustic scenes is available. In practice, data is usually limited and algorithms should be robust to adapt to unseen acoustic conditions without requiring extensive datasets for re-training. In this work, given the harmonic nature of siren signals, characterized by a periodically evolving fundamental frequency, we propose a low-complexity feature extraction method based on frequency tracking using a single-parameter adaptive notch filter. The features are then used to design a small-scale convolutional network suitable for training with limited data. The evaluation results indicate that the proposed model consistently outperforms the traditional spectrogram-based model when limited training data is available, achieves better cross-domain generalization and has a smaller size.

Index Terms— siren detection, frequency tracking, data-efficient learning, convolutional neural network

1. INTRODUCTION

The increasing level of automation of road vehicles requires robust systems that enable cars to understand their surroundings and either provide feedback to human drivers or autonomously interact with other road users. Environmental awareness is obtained by collecting information using multi-modal sensors including cameras, radar, lidar and acoustic sensors [1]. With the rich urban soundscape containing information on events happening on a road, sound detection has been widely explored for both monitoring purposes [2, 3] and to identify emergency or harmful situations that require attention [4, 5, 6, 7, 8, 9, 10, 11]. In particular, emergency vehicles (EV) are usually announced by the sound of their siren that can often be detected from a distance, before they become visible to the driver or can be identified using other sensing modalities (e.g., when obstacles occlude the line of sight or the EV is behind a corner).

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956962, from KU Leuven internal funds C3/23/056, and from FWO Research Project G0A0424N. This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

Several siren identification algorithms have been proposed, with deep learning models achieving state-of-the-art performance thanks to their robustness to the diversity of siren signals (three classes of sirens exist, namely two-tone, wail and yelp, and a large variability can be observed even between sirens of the same type) and to the prominent and non-stationary traffic background noise [4, 7, 12, 13, 14, 15]. Most state-of-the-art solutions rely on a spectrogram-based time-frequency representation of sound signals fed to 2D convolutional neural networks (CNN) [13, 12, 15]. These vision-inspired architectures process the spectrogram as a 2D image and achieve high accuracy when (diverse) enough training data is at disposal. Siren identification systems are faced with several use-case specific challenges. First, models to be deployed on-vehicle should have a low complexity to run on resource-constrained embedded devices. Second, models should have a vast generalization ability to face the diverse urban soundscape: not only the background noise can significantly differ based on factors such as the landscape (e.g., urban vs. rural), the region, or the time of the day (and day of the year), but also the characteristics of siren sounds can strongly vary among different countries. Finally, in practice, the amount of available data can be limited and datasets are unlikely to capture the diversity of the target scenes. In [15], the generalization ability of state-of-the-art siren identification networks is investigated, showing that models trained on one dataset do not always generalize well to unseen domains (cross-dataset setting): using synthetic data for training purposes is thus proposed to enhance data diversity. In [14], instead, data-efficient learning is achieved by fine-tuning a pre-trained environmental audio classification model in a *few-shot* setting to identify a specific type of two-tone siren.

Aiming for data-efficiency and low complexity, in this work, we propose novel features for siren identification based on frequency tracking. In contrast to the unstructured nature of traffic noise, sirens are artificial signals generated with a simple process: all types of sirens have a harmonic behavior characterized by a periodically evolving fundamental frequency, that can be tracked over time by means of an adaptive notch filter (ANF) [16, 17]. Adopting the single-parameter ANF design proposed in [17] (KalmANF), we design a CNN model using two features, namely the tracked fundamental frequency and the power ratio between the tracked sinusoidal component, extracted by the ANF, and the full audio signal. This allows to drastically reduce the input feature size compared to using the full spectrogram, and to thus adopt low-complexity networks. In the experimental evaluation, we show that the proposed model is suitable for training with a limited amount of data, consistently outperforming a spectrogram-based CNN [13] when small training sets are used. Moreover, the proposed model is 7 times smaller than the baseline [13] and achieves improved performance in a cross-dataset setting. Accompanying code is available at [18].

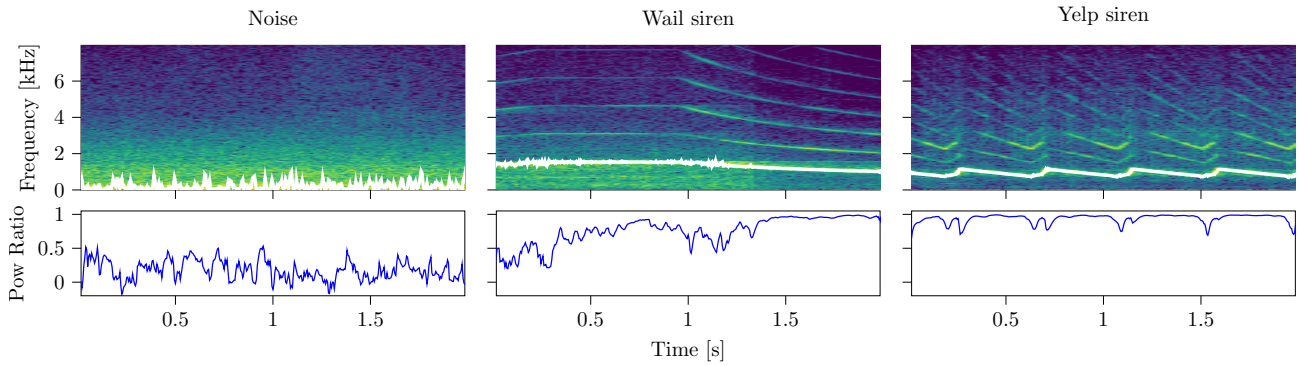

Figure 1: Proposed features for three audio samples: frequency tracked by the ANF algorithm (above, highlighted in white and overlaid to the full spectrogram) and power ratio (below).

Layer	Kernel Size	Filters/Neurons
Conv1D, stride 2	16	10
MaxPool 2x1	-	-
Conv1D, stride 2	8	20
MaxPool 2x1	-	-
Conv1D, stride 2	4	40
GlobAvgPool	-	-
Fully Connected	-	40
Fully Connected	-	20
Output	-	1

Table 1: Proposed ANFNet architecture, taking as input the two frequency tracking features.

2. PROBLEM STATEMENT AND BASELINE

We cast siren identification as a binary classification problem, where the goal is to assign a unique label (*siren* or *noise*) to a 2 s audio segment. The task is solved using the proposed architecture (ANFNet), introduced in Sec. 3, that we compare with the spectrogram-based baseline [13], denoted as VGGSiren. The network is a VGG-inspired [19] 2D-CNN composed of three blocks, each containing two 2D convolutional layers and a max pooling operation, followed by a 10-neurons FC layer and the single-neuron output layer. The network takes as input the mel-spectrogram of a 2 s-long single-channel audio segment.

3. PROPOSED METHOD

In this section, we summarize the KalmANF frequency tracking algorithm described in [17], underlining the modifications introduced to obtain the proposed features; we then present the proposed ANFNet siren identification network.

An ANF is a type of notch filter [20] whose notch frequency is recursively updated in order to suppress a high-energy sinusoidal component while leaving nearby frequencies relatively unaffected. The KalmanANF in [17] is expressed as a time-varying single-parameter bi-quadratic infinite impulse response (IIR) filter

$$H(q^{-1}, n) = \frac{1 - a(n)q^{-1} + q^{-2}}{1 - \rho a(n)q^{-1} + \rho^2 q^{-2}}\,, \tag{1}$$

where n is the time index, q denotes the discrete-time shift operator defined such that, for an input signal $y(n)$, $q^{-k}y(n) = y(n-k)$ [21]; $\rho < 1$ is a fixed hyperparameter denoting the radius of the complex conjugate pole pair and $a(n) = 2\cos[2\pi f(n)/f_s]$ is the single filter parameter, $f(n)$ being the notch frequency and f_s the sampling frequency. In the KalmANF, the time-varying coefficient $a(n)$ represents the state that is adaptively estimated in order to track the variations of $f(n)$ over time, as outlined in the following. The given N-samples long input signal $y(n)$ is filtered by the direct-form II [22] of $H(q^{-1}, n-1)$ using a joint delay line for the feedforward and the feedback paths of the IIR filter. The delay line signal $s(n)$ is defined as [17]

$$s(n) = y(n) + \rho a(n-1)s(n-1) - \rho^2 s(n-2)\,. \tag{2}$$

In the KalmANF, $s(n)$ represents the measurement. This results in the state space model (see [17] for the detailed derivation)

$$\begin{bmatrix} a(n) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a(n-1) \\ 1 \end{bmatrix} + \begin{bmatrix} w(n) \\ 0 \end{bmatrix} \tag{3}$$

$$s(n) = \begin{bmatrix} s(n-1) & -s(n-2) \end{bmatrix} \begin{bmatrix} a(n) \\ 1 \end{bmatrix} + e(n)\,, \tag{4}$$

where $e(n)$ is the residual signal obtained at the output of the notch filter and $w(n)$ is the process noise. Based on this state-space model, $a(n)$ can be estimated in a recursive manner using the Kalman filter [23]. The estimation procedure consists in the recursive update of the covariance of the prediction error $\hat{p}(n)$, the Kalman gain $k(n)$, and the parameter estimate $\hat{a}(n)$. These steps involve scalar operations and require a memory of 2 past samples. The filter relies on tuning three hyperparameters, namely the pole radius ρ, the variance σ_e of the residual $e(n)$ and the variance σ_w of the process noise $w(n)$. First, given the previous estimate $\hat{a}(n-1)$, the measurement $s(n)$ is computed according to (2). Then, the covariance of the prediction error is computed as

$$\hat{p}(n|n-1) = \hat{p}(n-1) + \sigma_w \tag{5}$$

and is used to obtain the Kalman gain

$$k(n) = \frac{s(n-1)}{s^2(n-1) + \frac{\sigma_e}{\hat{p}(n|n-1)}}\,. \tag{6}$$

The update equation to estimate the current value of the parameter $\hat{a}(n)$ can then be expressed as

$$\hat{a}(n) = \hat{a}(n-1) + k(n)e(n)\,, \tag{7}$$

where, from eq. (4), the residual takes the value

$$e(n) = s(n) - \hat{a}(n-1)s(n-1) + s(n-2)\,. \tag{8}$$

Finally, the covariance of the prediction error is updated by

$$\hat{p}(n) = \left(1 - \frac{s^2(n-1)}{s^2(n-1) + \frac{\sigma_e}{\hat{p}(n|n-1)}}\right)\hat{p}(n|n-1)\,. \tag{9}$$

At each time step, the estimated parameter $\hat{a}(n)$ contains information on the frequency tracked by the ANF, that is retrieved as $\hat{f}(n) = (f_s/2\pi)\arccos[\hat{a}(n)/2]$ and will be used as a first feature for the siren identification network. It is important to notice that the tracked frequency is not necessarily the fundamental frequency, but the one with the highest energy. This comes with the advantage that, if the fundamental of a siren is missing or hidden in the background noise, the higher harmonics could still be tracked by the ANF.

We then expand the above formulation to introduce a second feature that we call *power ratio*, expressing the ratio between the power of the suppressed sinusoidal component and of the input signal. At each time step, the power of the input signal, the notched signal (after $\hat{f}$ has been suppressed) and the suppressed frequency component can be estimated recursively as

$$P_y(n) = \lambda P_y(n-1) + (1-\lambda)y^2(n) \tag{10}$$

$$P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n)\,, \tag{11}$$

$$P_f(n) = P_y(n) - P_e(n)\,, \tag{12}$$

where $\lambda = e^{-1/(\tau f_s)}$, with τ constituting a first additional hyperparameter representing the time constant for recursive averaging. The power ratio is finally computed as

$$P_{\text{ratio}}(n) = P_f(n)/P_y(n)\,. \tag{13}$$

To reduce the feature size, we finally downsample P_f and $\hat{f}$ by a factor q_{down}, the second additional hyperparameter of the proposed method. The procedure is summarized in Algorithm 1.

In Fig. 1 the $\hat{f}$ and P_{ratio} features are shown for a noise sample and two different siren samples (wail and yelp) extracted from the sireNNet dataset [24], that will be used for the experimental evaluation. In the top row, the tracked $\hat{f}$ is overlaid to the full spectrogram: nevertheless, we remark that the frequency estimate is obtained directly from the time-domain signal without computing the spectrogram. The visualization shows the effectiveness of the tracking algorithm when applied to siren signals, and underlines the clear contrast between the features extracted from a (structured) siren and the (unstructured) traffic noise.

We solve the siren identification problem using the ANFNet network, that processes the $\hat{f}$ and P_{ratio} features extracted from a single channel, 2 s-long audio sample: the two features are stacked into a 2-channel vector provided as input to the first layer. The architecture (see Tab. 1) contains three 1D convolutional layers (Conv1D) with, respectively, 10, 20 and 40 filters having kernel size 16, 8 and 4. Each of the first two Conv1D layers is followed by a max pooling operation (MaxPool) for dimensionality reduction. After the third one, a global average pooling operation (GlobAvgPool) is used as interface between the convolutional part and the classification head, composed of two fully connected (FC) layers with 40 and 20 neurons, respectively, and a single neuron output layer. We use the ReLU activation function in each hidden layer and the sigmoid activation in the output layer, and introduce dropout layers with 0.25 drop probability after each FC layer to prevent overfitting. The network has 7.7 k floating-point 32-bit parameters.

Algorithm 1: The modified KalmANF algorithm

Initialize $s(0), s(1), \hat{a}(1), \hat{p}(1) = 0$
Set $\sigma_e, \sigma_w, \rho, \tau, q_{\text{down}}$
for $n = 2$ to $N-1$ **do**
  $\hat{p}(n|n-1) = \hat{p}(n-1) + \sigma_w$
  $s(n) = y(n) + \rho\hat{a}(n-1)s(n-1) - \rho^2 s(n-2)$
  $k(n) = \frac{s(n-1)}{s^2(n-1) + \frac{\sigma_e}{\hat{p}(n|n-1)}}$
  $e(n) = s(n) - \hat{a}(n-1)s(n-1) + s(n-2)$
  $\hat{a}(n) = \hat{a}(n-1) + k(n)e(n)$
  $\hat{p}(n) = \left(1 - \frac{s^2(n-1)}{s^2(n-1) + \frac{\sigma_e}{\hat{p}(n|n-1)}}\right)\hat{p}(n|n-1)$
  if $|\hat{a}(n)| > 2$ **then**
    $\hat{a}(n) = 2\text{sgn}(\hat{a}(n))$
  end
  $\hat{f}(n) = (f_s/2\pi)\arccos[\hat{a}(n)/2]$
  $P_y(n) = \lambda P_y(n-1) + (1-\lambda)y^2(n)$
  $P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n)$
  $P_f(n) = P_y(n) - P_e(n)$
  $P_{\text{ratio}}(n) = P_f(n)/P_y(n)$
end
Downsample $\hat{f}$ and P_{ratio} by factor q_{down}

4. EVALUATION

We run an evaluation campaign to assess the effectiveness of the proposed method: to promote reproducibility, the code is available at [18]. For training, we use the sireNNet dataset [24], containing a total of 421 noise and 1254 siren samples including different types of sirens. All samples have a duration of 3 s, and since half of the siren files are artificially generated for data augmentation purposes, we exclude them and use only the 627 non-augmented siren samples. We divide this dataset into training, validation and test data with ratios $[0.8, 0.1, 0.1]$. In order to perform a data-efficient evaluation, we split the training set into subsets of different size similarly to [25]: in particular, we create subsets containing an increasing percentage of the full training set, with ratios $0.25\%, 0.5\%, 1\%, 2\%, 4\%, 8\%, 16\%, 32\%, 64\%$ and 100% (i.e., the entire training set). 10 folds are randomly generated for each subset, in order to compute the mean and standard deviation of the results. The subsets are created such that (i) smaller splits are subsets of larger ones; (ii) the data distribution is kept similar to that of the entire training set; (iii) overlapping folds are allowed. The validation and test sets are always used without additional splitting. To further evaluate the generalization performance in a cross-dataset setting, we also use a subset of 210 audio files randomly extracted from the dataset [26] (that we will call LSSiren) for testing; this dataset contains siren and noise files with lengths between 3 s and 15 s. All files of both datasets have been re-sampled to 16 kHz and converted to mono; moreover, since we use 2 s samples as input, we take only the first two seconds of each file of the sireNNet dataset, and divide the LSSiren files in non-overlapping 2 s segments. Both datasets include real recordings, with background traffic noise, moving sirens and Doppler effect.

We implement the proposed ANFNet and the baseline VGGSiren using Pytorch Lightning [27]: for the KalmANF algorithm, we set the hyperparameters $\rho = 0.99, \sigma_w = 10^{-5}, \sigma_e = 0.66, q_{\text{down}} = 5, \tau = 0.02$, all chosen by manual tuning based on the best loss obtained on the validation set. For VGGSiren, to com-

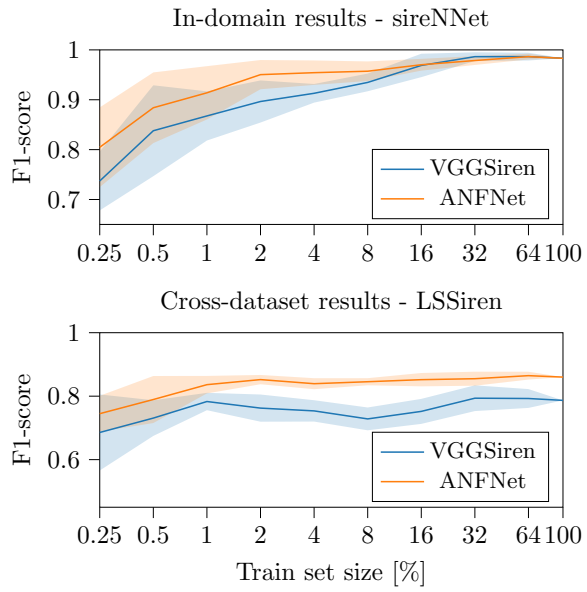

Figure 2: Comparison of the average (solid line) and standard deviation (shaded area) of the F1-score for the baseline VGGSiren and the proposed ANFNet, trained with an increasing amount of data: in-domain evaluation (above) and cross-dataset evaluation (below).

pute the mel-spectrogram we use a 1024 samples Hann window with 512 samples overlap, and 128 mel channels. As a result, VGGSiren has a total of 53.9 k floating point 32-bits parameters, thus being 7 times larger than the proposed ANFNet. For VGGSiren, we apply peak normalization to the mel-spectrograms, whereas for ANFNet we normalize the $\hat{f}$ feature to $f_s/2$ (the P_{ratio} feature is normalized by definition). In all experiments we train both models for 400 epochs using the binary cross-entropy loss function, the Adam optimizer [28] with learning rate between 0.001 and 0.005, a batch size between 2 and 32, both depending on the size of the training split, and select the best model based on the validation loss. To evaluate the performance, we use the F1-score [29] and the area under the precision-recall curve (AUPRC) [29, 30] metrics, chosen to deal with non-balanced datasets.

We train both models on the 10 folds of each sireNNet subset. Note that the $0.25\%, 0.5\%$ and 1% splits contain, respectively, only $2, 4$ and 9 samples, making the problem extremely challenging and comparable to that of few-shot learning (without pre-training). First, we evaluate in-domain performance on the sireNNet test set and report in Fig. 2 the average and standard deviation (shaded area) of the F1-score. In Tab. 2 we report the average F1-score and AUPRC obtained with the two models for each training split. As expected, the performance of both networks degrades as the amount of training data decreases; nevertheless, ANFNet outperforms the baseline when trained using smaller subsets, and reaches a comparable performance on the larger ones (with a lower complexity).

We then evaluate the models on the LSSiren data (cross-dataset setting) and report the results in the bottom plot of Fig. 2 and in Tab. 3. Again, the performance of both decreases as the training dataset size decreases. In this case, the proposed ANFNet significantly outperforms the baseline on all subsets. These results indicate that the proposed features help the network capture the difference between siren and noise classes also when limited data

	F1-score		**AUPRC**	
	VGGSiren	ANFNet	VGGSiren	ANFNet
0.25	0.7372	**0.8047**	0.8120	**0.8471**
0.5	0.8379	**0.8840**	0.8844	**0.9162**
1	0.8676	**0.9139**	0.9602	**0.9658**
2	0.8965	**0.9504**	0.9745	**0.9787**
4	0.9130	**0.9543**	**0.9781**	0.9772
8	0.9348	**0.9572**	**0.9860**	0.9796
16	0.9688	**0.9702**	**0.9949**	0.9904
32	**0.9864**	0.9787	**0.9990**	0.9962
64	**0.9865**	**0.9865**	**0.9996**	0.9966
100	**0.9833**	0.9831	**0.9995**	**0.9995**

Table 2: In-domain evaluation: average F1-score and AUPRC metrics computed on the sireNNet test set for VGGSiren and ANFNet.

	F1-score		**AUPRC**	
	VGGSiren	ANFNet	VGGSiren	ANFNet
0.25	0.6856	**0.7448**	0.6962	**0.7364**
0.5	0.7309	**0.7895**	0.7512	**0.8447**
1	0.7831	**0.8362**	0.8607	**0.9169**
2	0.7624	**0.8522**	0.8426	**0.9272**
4	0.7536	**0.8393**	0.8349	**0.9147**
8	0.7284	**0.8455**	0.7914	**0.9180**
16	0.7520	**0.8520**	0.8100	**0.9247**
32	0.7936	**0.8550**	0.8492	**0.9302**
64	0.7926	**0.8646**	0.8229	**0.9355**
100	0.7867	**0.8601**	0.8052	**0.9384**

Table 3: Cross-dataset evaluation: average F1-score and AUPRC metrics computed on LSSiren data for VGGSiren and ANFNet.

is available, suggesting their potential for data-efficient learning. Moreover, the evaluation underlines that the proposed features ensure an enhanced robustness to domain shift compared to the mel-spectrogram. In Fig. 2 it is also visible that the standard deviation is reduced compared to VGGSiren, showing that ANFNet is less sensitive to the choice of training samples. Finally, ANFNet has a lower complexity, with a 7 times smaller network size (7.7 k parameters vs. the 53.9 k of VGGSiren). Note that, thanks to time-domain processing and downsampling, the feature extraction procedure has also a reduced complexity and the ANFNet input features have a smaller size compared to the mel-spectrograms used by VGGSiren.

5. CONCLUSIONS

In this work, we investigated two novel features based on frequency tracking for training a siren identification model. Given the harmonic nature of siren signals, as opposed to the unstructured background noise, the features are effective for learning in a data-efficient setting, when limited data is available. The proposed system outperforms a spectrogram-based baseline on in-domain test data, when limited training data is available, and always achieves better performance in a cross-dataset setting. Moreover, its reduced complexity promotes its adoption in the automotive domain. Future work will focus on extending the frequency tracker to include higher harmonics, further investigating the generalization performance of the proposed system and optimizing the model for complexity.

6. REFERENCES

[1] L. Marchegiani and X. Fafoutis, "How Well Can Driverless Vehicles Hear? A Gentle Introduction to Auditory Perception for Autonomous and Smart Vehicles," *IEEE Intell. Transp. Syst. Mag.*, pp. 92–105, 2022.

[2] M. Won, "Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey," *IEEE Access*, vol. 8, pp. 73 340–73 358, 2020.

[3] S. Damiano, L. Bondi, S. Ghaffarzadegan, A. Guntoro, and T. van Waterschoot, "Can synthetic data boost the training of deep acoustic vehicle counting networks?" in *Proc. 2024 Int. Conf. Acoust. Speech Sig. Process. (ICASSP)*, Seoul, South Korea, 2024, pp. 631–635.

[4] F. Walden, S. Dasgupta, M. Rahman, and M. Islam, "Improving the Environmental Perception of Autonomous Vehicles using Deep Learning-based Audio Classification," *arXiv:2209.04075*, Sept. 2022.

[5] J. Yin, S. Damiano, M. Verhelst, T. van Waterschoot, and A. Guntoro, "Real-Time Acoustic Perception for Automotive Applications," in *2023 Design, Automation & Test in Europe Conf. Exhib. (DATE)*, Antwerp, Belgium, Apr. 2023, pp. 1–6.

[6] Y. Furletov, V. Willert, and J. Adamy, "Auditory Scene Understanding for Autonomous Driving," in *2021 IEEE Intell. Vehicles Symp. (IV)*, Nagoya, Japan, July 2021, pp. 697–702.

[7] L. Marchegiani and P. Newman, "Listening for Sirens: Locating and Classifying Acoustic Alarms in City Scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17 087–17 096, 2022.

[8] M. K. Nandwana and T. Hasan, "Towards Smart-Cars That Can Listen: Abnormal Acoustic Event Detection on the Road," in *Proc. 2016 Interspeech*, San Francisco, USA, Sept. 2016, pp. 2968–2971.

[9] D. Lee, S. Lee, Y. Han, and K. Lee, "Ensemble of Convolutional Neural Networks for Weakly-Supervised Sound Event Detection Using Multiple Scale Input," in *Proc. Detection Classification Acoust. Scenes Events 2017 Workshop (DCASE2017)*, Munich, Germany, Nov. 2017.

[10] A. E. Ramirez, E. Donati, and C. Chousidis, "A siren identification system using deep learning to aid hearing-impaired people," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105000, 2022.

[11] D. Pramanick, H. Ansar, H. Kumar, S. Pranav, R. Tengshe, and B. Fatimah, "Deep learning based urban sound classification and ambulance siren detector using spectrogram," in *Proc. 12th Int. Conf. Computing Comm. Networking Technologies (ICCCNT)*, Kharagpur, India, 2021, pp. 1–6.

[12] V.-T. Tran and W.-H. Tsai, "Acoustic-Based Emergency Vehicle Detection Using Convolutional Neural Networks," *IEEE Access*, vol. 8, pp. 75 702–75 713, 2020.

[13] M. Cantarini, A. Brocanelli, L. Gabrielli, and S. Squartini, "Acoustic Features for Deep Learning-Based Models for Emergency Siren Detection: An Evaluation Study," in *2021 12th Int. Symp. Image Sig. Process. Anal. (ISPA)*, Zagreb, Croatia, Sept. 2021, pp. 47–53.

[14] M. Cantarini, L. Gabrielli, and S. Squartini, "Few-Shot Emergency Siren Detection," *Sensors*, vol. 22, no. 12, p. 4338, June 2022.

[15] S. Damiano, B. Cramer, A. Guntoro, and T. van Waterschoot, "Synthetic Data Generation Techniques for Training Deep Acoustic Siren Identification Networks," *Frontiers Sig. Process.*, vol. 4, 2024.

[16] D. Rao and Sun-Yuan Kung, "Adaptive notch filtering for the retrieval of sinusoids in noise," *IEEE Trans. Acoust. Speech Sig. Process.*, vol. 32, no. 4, pp. 791–802, Aug. 1984.

[17] R. Ali and T. van Waterschoot, "A Frequency Tracker Based on a Kalman Filter Update of a Single Parameter Adaptive Notch Filter," in *Proc. 26th Int. Conf. Digital Audio Effects (DAFx)*, Copenhagen, Denmark, Sept. 2023.

[18] S. Damiano and T. Dietzen, "An ANF-based siren identification system," Github Repository, 2024. [Online]. Available: https://github.com/steDamiano/anf-siren-identification

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[20] K. Hirano, S. Nishimura, and S. Mitra, "Design of Digital Notch Filters," *IEEE Trans. Commun.*, vol. 22, no. 7, pp. 964–970, July 1974.

[21] T. van Waterschoot and M. Moonen, "Fifty years of acoustic feedback control: State of the art and future challenges," *Proc. IEEE*, vol. 99, no. 2, pp. 288–327, 2011.

[22] J. Travassos-Romano and M. Bellanger, "Fast least squares adaptive notch filtering," in *Proc. Int. Conf. Acoust., Speech Sig. Process. ICASSP 1988*, New York, NY, USA, 1988, pp. 1391–1394.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[24] A. Shah and A. Singh, "sireNNet-Emergency Vehicle Siren Classification Dataset For Urban Applications," Mendeley Data, 2023, doi: 10.17632/j4ydzzv4kb.1.

[25] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, "Data-efficient low-complexity acoustic scene classification in the dcase 2024 challenge," *arXiv:1706.10006*, 2024.

[26] M. Asif, M. Usaid, M. Rashid, T. Rajab, S. Hussain, and S. Wasi, "Large-scale audio dataset for emergency vehicle sirens and road noises," *Scientific Data*, vol. 9, no. 1, p. 599, Oct. 2022.

[27] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," Github Repository, Mar. 2019. [Online]. Available: https://github.com/Lightning-AI/lightning

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

[29] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[30] Q. Qi, Y. Luo, Z. Xu, S. Ji, and T. Yang, "Stochastic optimization of areas under precision-recall curves with provable convergence," in *Proc. 35th Int. Conf. Neural Information Process. Syst.*, online, 2021.