IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 November 2022, accepted 20 December 2022, date of publication 22 December 2022,
date of current version 29 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3231681

RESEARCH ARTICLE

Sound Event Detection for Human Safety and Security in Noisy Environments

MICHAEL NERI[1], (Graduate Student Member, IEEE), FEDERICA BATTISTI[2], (Senior Member, IEEE), ALESSANDRO NERI[1], (Life Member, IEEE), AND MARCO CARLI[1], (Senior Member, IEEE)

[1]Department of Industrial, Electronic and Mechanical Engineering, Roma Tre University, 00146 Rome, Italy
[2]Department of Information Engineering, University of Padova, 35131 Padua, Italy

Corresponding author: Michael Neri (michael.neri@uniroma3.it)

This work was supported in part by the H2020 Electronics Components and Systems for European Leadership (ECSEL) European Union (EU) Project Intelligent Secure Trustable Things (INSECTT), Italy, under Agreement 876038.

ABSTRACT The objective of a sound event detector is to recognize anomalies in an audio clip and return their onset and offset. However, detecting sound events in noisy environments is a challenging task. This is due to the fact that in a real audio signal several sound sources co-exist. Moreover, the characteristics of polyphonic audios are different from isolated recordings. It is also necessary to consider the presence of noise (e.g. thermal and environmental). In this contribution, we present a sound anomaly detection system based on a fully convolutional network which exploits image spatial filtering and an Atrous Spatial Pyramid Pooling module. To cope with the lack of datasets specifically designed for sound event detection, a dataset for the specific application of noisy bus environments has been designed. The dataset has been obtained by mixing background audio files, recorded in a real environment, with anomalous events extracted from monophonic collections of labelled audios. The performances of the proposed system have been evaluated through segment-based metrics such as error rate, recall, and F1-Score. Moreover, robustness and precision have been evaluated through four different tests. The analysis of the results shows that the proposed sound event detector outperforms both state-of-the-art methods and general purpose deep learning-solutions.

INDEX TERMS Audio processing, deep learning, human safety, sound event detection, spatial filters.

I. INTRODUCTION

Hearing is a mechanical sense that converts physical movements, i.e. sounds, into nerve impulses by detecting pressure variations of the surrounding medium [1]. This process is performed by the human ear that acts as transducer and amplifier for the human brain. Hence, an audio signal conveys relevant information on the surrounding environment. For this reason, many efforts have been devoted to audio pattern recognition [2], [3], [4], [5], [6], [7]. Several applications of audio pattern recognition are in the audio security domain. This includes tasks such as audio classification and tagging [8], [9], [10], [11], [12], [13], [14], sentiment analysis [15], [16], audio source separation [17], [18], [19], Anomalous Sound Detection (ASD) [20], [21], [22], and Sound Event Detection (SED) [23].

In more detail, differently from monophonic audio classification, the objective of a SED system is to identify both the type of event and the exact time of its onset and offset [24]. However, recent state-of-the-art models trained on AudioSet [8] have shown to be unsuitable for human-security and safety oriented applications [24]. In fact, generally, SED systems provide accurate start and end time identification of acoustic events, while their recall, that is the percentage of the fraction of relevant elements that are successfully retrieved, may be unsatisfactory. Moreover, the lack of large annotated audio datasets has a significant impact on the performance of SED models, leading to weak generalization capabilities of deep learning-based systems.

To address these problems, a new lightweight SED approach, AuSPP, which aims to detect audio events that

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

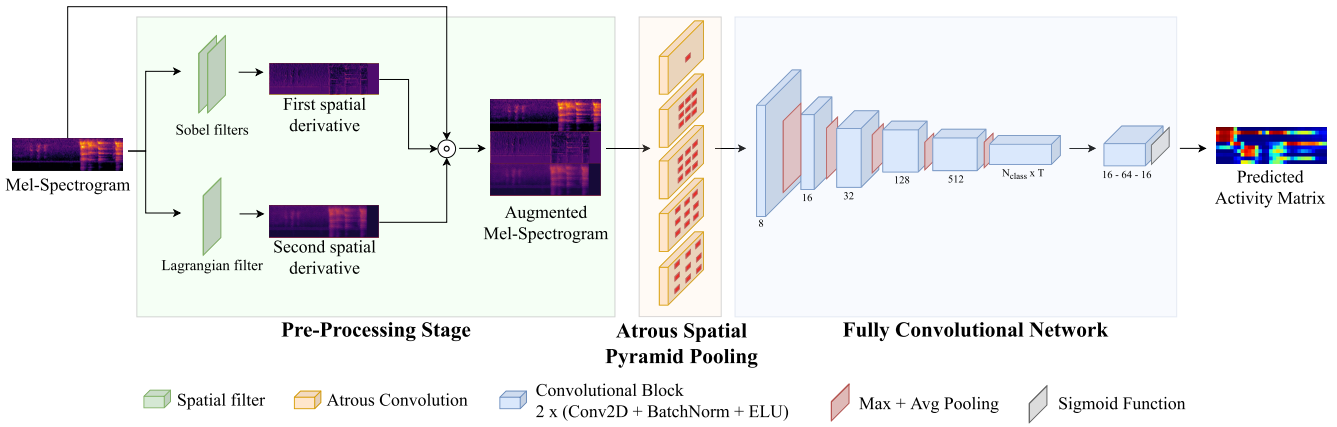


FIGURE 1. Structure of AuSPP, the proposed model for SED.

potentially denote the presence of circumstances threatening public safety and security (e.g., broken glass, gunshots, or shouting) has been proposed. AuSPP exploits state-of-the-art spatial filters on Mel-Spectrograms and introduces the Atrous Spatial Pyramid Pooling (ASPP) module [25] for the first time in a SED system. In addition, in the proposed approach, the number of learnable parameters depends on the number of class anomalies and on the time resolution. Hence, with respect to state-of-the-art SED approaches, it is possible to customize the behaviour of AuSPP, drastically reducing its complexity and detecting more sound events than state-of-the-art models.

The proposed model has been tested in a public transportation vehicle. The motivation of the choice is twofold. First, modern buses are equipped with on-board sensors able to collect heterogeneous data that can be employed for maintenance, i.e., the Automatic Vehicle Monitoring (AVM) paradigm [26]. Second, the same raw audio information can be exploited for granting passenger security and safety in a noisy urban environment.

In this context, an annotated audio SED dataset, Sound Event Detection Dataset On Bus (SEDDOB), specifically designed for the bus environment, has been devised. In more detail, labelled audios with onset and offset time of different types of audio events are provided. To the best of our knowledge, this is the first contribution of an audio dataset for human safety in the public transport system.

To summarize, the contributions of the work are:

- the definition of a new augmented spectrogram that exploits spatial filters to enhance time-frequency patterns by means of the ASPP module;
- the design of an end-to-end SED that is more lightweight than state-of-the-art approaches. Moreover, the number of parameters depends on the time resolution and on the number of classes, thus being customizable;
- the generation of a new SED dataset for the bus environment. Synthesis of real background recording with anomalous events from state-of-the-art monophonic datasets have been performed.

This work is organized as follows: Section II reviews state-of-the-art approaches for sound feature extraction and classification, deep learning models, and audio datasets; Section III contains the description of the proposed SED model; Section IV details the synthetic dataset called SEDDOB with its statistics; Section V describes the experimental results of general-purpose deep learning methods, state-of-the-art SED models, and the proposed solution. Finally, in Section VI the conclusions are drawn.

II. RELATED WORKS

A. DEEP LEARNING MODELS

One of the first approaches to SED was the use of Hidden Markov Model (HMM) combined with Gaussian Mixture Models (GMMs) with Mel-Frequency Cepstral Coefficient (MFCC) as features [27], [28]. In these approaches, each anomaly class is modeled by an HMM and the maximum likelihood is obtained by exploiting the Viterbi algorithm. Nonetheless, together with Non-negative Matrix Factorization (NMF) approaches [29], HMM-based methods show limited performance and can hardly generalize.

Deep learning-based SED techniques have reached good performance. Ding et al. [30] proposed a Convolutional Recurrent Neural Network (CRNN) able to identify short-term dependencies of audio patterns by applying a multi-scale detection method. With a similar approach, Chatterjee et al. [31] introduced a Transposed Convolutional recurrent Neural Network (TCRNN) that incorporates Mel Instantaneous Frequency spectrogram (mel-IFgram) features. However, recurrent layers in neural networks are subject to the vanishing gradient problem, leading to lack of generalization capabilities. Moreover, the training process is slower since the data has to be fed sequentially and parallel computing cannot be exploited.

To handle the lack of annotated audio datasets, self-supervised, active learning and attention-based techniques have been recently proposed [32], [33], [34], [35]. However, large scale Convolutional Neural Networks (CNNs) [11], pre-trained on AudioSet [8], have achieved

top performance on most of audio pattern recognition challenges.

B. SED DATASETS

From monophonic collections of labelled audios, i.e., audio classification database such as UrbanSound8k [9], FSD50K [36], ESC50 [10], AudioSet [8], and FreeSound [37], [38], the research community has started to generate SED datasets by synthesizing normal background recordings with audio anomalies. Turpault et al. [39] proposed Domestic Environment Sound Event Detection (DESED) that contains precisely labelled audio (with precise onset and offset times) with the aim of recognizing sound event classes in domestic environments. In more details, the authors injected recorded anomalies on background soundscapes of Detection and Classification of Acoustic Scenes and Events (DCASE) challenges. The process is performed by using the open source software Scaper [40]. From UrbanSound8k [9], in [40] the authors proposed URBAN-SED that contains 10.000 soundscapes with sound event annotations for urban sound monitoring use-cases. With the same technique, the USM-SED [41] dataset based on sounds taken from the FSD50k dataset, consisting in 20.000 polyphonic soundscapes, has been built.

III. PROPOSED MODEL

One of the peculiarities of the proposed model, AuSPP, is the exploitation of a larger dimensional input than state-of-the-art Mel-Spectrogram based methods. This choice allows the extraction of a larger number of semantic audio features. More specifically, the information provided by the concatenation of spatial derivatives of the Mel-Spectrogram helps the model to learn frequency patterns for jointly classifying the audio events and their corresponding onset and offset times.

AuSPP can be partitioned into three main blocks: a pre-processing stage, the ASPP, and a Fully Convolutional Network (FCN). The first stage is responsible for extracting a time-frequency audio representation, applying spatial filters to the input audio spectrum, and arranging the output into an augmented Mel-Spectrogram. Subsequently, the ASPP module is introduced to combine the output of dilated convolutional filters. Finally, a FCN processes the ASPP output to obtain the predicted activity heatmap. The overall architecture is depicted in Figure 1.

A. PRE-PROCESSING STAGE

Let $\mathbf{x} : [0, \ldots, L-1] \rightarrow \mathbb{R}$ be a raw audio signal with L samples and f_s be the sampling frequency in Hertz. Moreover, let $\mathbf{w} : [0, \ldots, N-1] \rightarrow \mathbb{R}$ be a window function of N samples and $H \in \mathbb{N}$ be the hop size which determines the number of overlapped samples between time segments. Then, the discrete Short-Time Fourier Transform (STFT) $X_{STFT} \in \mathbb{C}$ of the input signal $\mathbf{x}$ is given by

$$X_{STFT}[m,k] = \sum_{n=0}^{N-1} \mathbf{x}[n+mH]\mathbf{w}[n]e^{\frac{-2\pi ikn}{N}}, \tag{1}$$

where $m \in [0, \ldots, M-1]$ and $k \in [0, \ldots, K]$ are the time and frequency bins, respectively. The number $M = \lfloor \frac{L-N}{H} \rfloor$ represents the maximum time frame index in which the input signal $\mathbf{x}$ is present. The maximum frequency index $K = \frac{N}{2}$ represents the Nyquist frequency $\frac{f_s}{2}$.

The Mel-Spectrogram X is computed by means of the Mel-Filterbank $H_{mel}(\cdot)$ on the squared magnitude of the STFT

$$X[m,k] = H_{mel}(|X_{STFT}|^2). \tag{2}$$

In more detail, the Mel-filterbank groups the spectral values of each time frame into n_{mels} logarithmic bins in order to model the human sound perception. Hence, the Mel-Spectrogram X has size $M \times n_{mels}$.

To enhance the audio patterns on the Mel-Spectrogram, Sobel [42] and Langrangian [43] operators have been employed. Let H_u and H_v be the horizontal and vertical first order spatial derivatives, respectively. Furthermore, let H_l be the second order spatial derivative

$$H_u = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad H_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \tag{3}$$

$$H_l = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4}$$

The first Mel-Spectrogram spatial derivative is calculated by convolution:

$$X'_u = X * H_u, \tag{5}$$

$$X'_v = X * H_v, \tag{6}$$

$$X' = \sqrt{X'^2_u + X'^2_v}. \tag{7}$$

Similarly, the second spatial derivative is obtained as:

$$X'' = X * H_l. \tag{8}$$

Finally, the Mel-Spectrogram is concatenated on the frequency axis with its derivatives in an augmented Mel-Spectrogram $\hat{X}$ with size $M \times 3n_{mels}$:

$$\hat{X} = X \circ X' \circ X'', \tag{9}$$

where $\circ$ denotes the concatenation function. An example of an augmented spectrogram $\hat{H}$ is depicted in Figure 2.

B. ATROUS SPATIAL PYRAMID POOLING

In this work, the ASPP module [25] is employed. In more detail, it applies atrous convolutions to the augmented Mel-Spectrogram in order to extract more relevant spatial features. Considering two-dimensional signals, given the augmented Mel-Spectrogram $\hat{X}$, a convolutional kernel W, and a region of interest I, the output of the atrous convolution for the selected region, $Y[i]$, is:

$$Y[i] = \sum_k \hat{X}[i + (r \cdot k)]W[k], \tag{10}$$

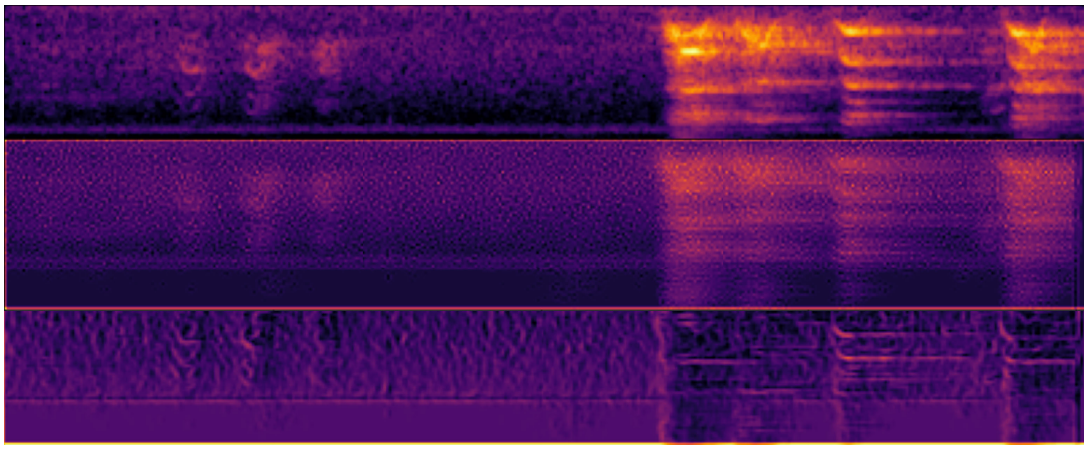


FIGURE 2. Augmented spectrogram $\hat{X}$ of size $M \times 3n_{mels}$. The x-axis corresponds to the time domain whereas the y-axis represents the frequency domain. It is worth noticing that the output of the two spatial filters emphasises the edges of the Mel-Spectrogram, i.e., the most intense frequency bins in terms of magnitude.

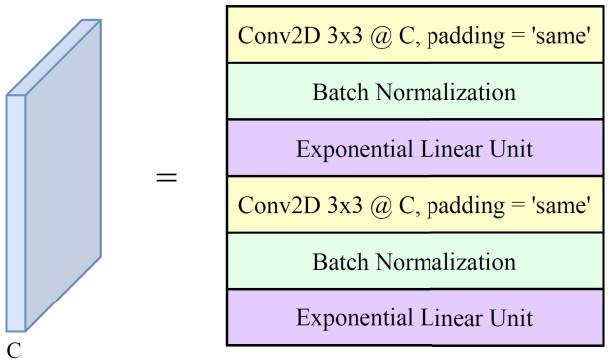


FIGURE 3. General convolutional block with C output channels.

where the atrous rate r is the stride parameter of the convolutional layer of the network, which allows to apply convolutions to the input spectrogram $\hat{X}$ with upsampled zero-padded filters. The variable k accounts for all the possible regions of the image. In the proposed model, the ASPP module is composed of 5 atrous convolutions with kernels of size 3×3 with an atrous rate of $1, 2, 4, 8$, and 16, respectively. This design choice have been considered for capturing new time-frequency pattern across the augmented spectrogram.

C. FULLY CONVOLUTIONAL NETWORK

Inspired by the network architecture proposed in [11], let ConvBlock(C) be the generic convolutional block with C output channels, shown in Figure 3. It is composed of two consecutive Conv2D-BatchNormalization-Exponential Linear Unit (ELU) [44], [45] blocks with 3×3 kernels. Variable pooling sizes are applied to intermediate ConvBlock($\cdot$) outputs to reduce the size of the feature maps. Moreover, a Dropout [46] layer is applied at the end of each ConvBlock($\cdot$) to reduce overfitting.

Finally, the output of the module, that is the predicted activity matrix $\hat{Y}$, is obtained by applying the sigmoid activation function. Details about the configurations of AuSPP convolutional blocks are reported in Table 1. It is worth noticing that the number of parameters changes with the size of the predicted activity matrix. More specifically, the model becomes more complex if a finer time resolution and/or a larger number of audio classes are required.

TABLE 1. Description of the proposed FCN of AuSPP.

Input: $\hat{X}$ Augmented Mel-Spectrogram $T \times 3n_{mels}$
ConvBlock(8) Max + Average Pooling 4×2
ConvBlock(16) Max + Average Pooling 4×2
ConvBlock(32) Max + Average Pooling 2×2
ConvBlock(128) Max + Average Pooling 2×2
ConvBlock(512) Max + Average Pooling 2×2
ConvBlock($T \times N_{class}$) Global Max + Average Pooling
Reshape to 1 channel $T \times N_{class}$
ConvBlock(16)
ConvBlock(64)
ConvBlock(16)
ConvBlock(1)
Sigmoid activation function
Output: $\hat{Y}$ Binary Activity Matrix $T \times N_{class}$

D. LOSS FUNCTION

The loss function is a fundamental component of a deep learning-based approach. The training process aims at finding the best configuration of weights that minimizes the training loss. This procedure is performed by means of the mini-batch gradient descent algorithm, exploiting the derivative of the loss function with respect to the weights of the model. Let $\hat{Y}$ and Y be the predicted and ground truth binary activity matrices, respectively, with size $T \times N_{class}$. Then, we employ

TABLE 2. SEDDOB characteristics.

Audio Settings	
Number of soundscapes	10000 samples
Fixed duration	$4s$
Sampling frequency f_s	16000 Hz
Anomalies Statistics	
Minimum number of events	0
Maximum number of events	4
Distribution of the number of events	Uniform
Distribution of class anomalies	Uniform
Augmentation Statistics	
Minimum Signal-to-Noise Ratio (SNR)	-5dB
Maximum SNR	0dB
Distribution SNR	Uniform
Minimum pitch shift	-0.5 semitones
Maximum pitch shift	0.5 semitones
Distribution pitch shift	Uniform
Minimum time stretch	0.9
Maximum time stretch	1.1
Distribution stretch	Uniform

the Binary Cross Entropy (BCE) loss, that is

$$\mathcal{L}_{BCE}(\hat{Y}, Y) = -\sum_{j=1}^{J}(Y_j \ln \hat{Y}_j + (1 - Y_j)\ln(1 - \hat{Y}_j)), \quad (11)$$

where J is the number of training audio samples in a batch. An example of the two activity matrices is shown in Figure 5.

E. DATA AUGMENTATION

Time and time-frequency audio augmentation have been exploited. This procedure helps the model to generalize and to increase the overall performance by exploiting modified versions of the training dataset. As a drawback, the training process becomes slower compared to models not adopting data augmentation strategies. We employ SpecAugment [47] that randomly drops the time and frequency stripes of the initial Mel-Spectrogram X. In addition, before spectrum extraction, we apply pitch shift, frame shift, polarity inversion, and gain augmentations [48] to the input batch.

IV. SYNTHETIC DATASET-SEDDOB

In this work, a synthetic audio dataset SEDDOB has been designed. As previously mentioned, a large number of high quality annotated audios is required for training data-driven approaches. In the following we describe the recording setup and the synthesis of the sound classes.

A. BACKGROUND AUDIO

In order to collect a typical bus background, a microphone array and a recording unit have been deployed inside a bus. The microphone array has been positioned in 3 different locations (Figure 4). This choice accounts for the observation that in the bus used for the recordings, the engine is located in the rear. Therefore, the background sound pressure level and its spectral characteristics change with distance from the engine, which is the main source of background noise. The total length of the recorded background noise is 1 hour. All the recordings have been acquired in dedicated runs in compliance with General Data Protection Regulation (GDPR).

TABLE 3. STFT parameters and training hyperparameters for all the models.

STFT Settings	
Minimum frequency	50 Hz
Maximum frequency	8000 Hz
Hop Size H	160 samples
Window Size N	512 samples
Window function $\mathbf{w}$	Hanning
n_{FFT}	512 samples
Mel-Filterbank Settings	
n_{mels}	64 bins
Training Hyperparameters	
Learning rate λ	0.001
Batch size	16 samples
Maximum epochs	50 epochs
Learning rate scheduler	Reduce on Plateau of Validation Loss
Scheduler patience	4 epochs
Scheduler factor	0.1

B. FOREGROUND EVENTS

The audio events selected for the classification task are extracted from existing monophonic datasets: UrbanSound8k [9], ESC50 [10], and AudioSet [8]. 10 audio classes which can occur in a bus environment and that can relate to threatening events for public safety and security, have been selected: *breaking_glass, car_horn, gunshot, siren, slap, scream, cry, jackhammer, car_alarm, smoke_alarm*.

C. DATASET GENERATION

Scaper [40] has been used to generate the proposed dataset by adopting the parameters configuration reported in Table 2. The audio length is selected based on the characteristics of the human auditory perception. In fact, tests on human subjects confirm that 4 seconds are sufficient to correctly classify events [9]. Furthermore, the distribution of class events, together with their number, onset, and offset times, is set as uniform to create a balanced dataset. In addition, augmentation strategies such as pitch shift, time stretch, and variable SNR, have been introduced for generalization purposes.

To assess the performance of the proposed SED, two datasets (denoted as *full* and *reduced* dataset) with 10 and 4 audio classes have been generated. Both datasets are split into 10 folds to use different portions of the data for training and testing.

V. EXPERIMENTAL RESULTS

The AuSPP weights are updated by means of the Adam optimizer with learning rate λ. The STFT, Mel-Filterbank, and training hyperparameters for each model are listed in Table 3. The model implementation is based on Pytorch-Lightning and it is available at the following GitLab repository.

A. METRICS

The validation of an audio classification system is usually based on the accuracy score. More specifically, let *TP* be the number of true positive, *FP* be the number of false positive,

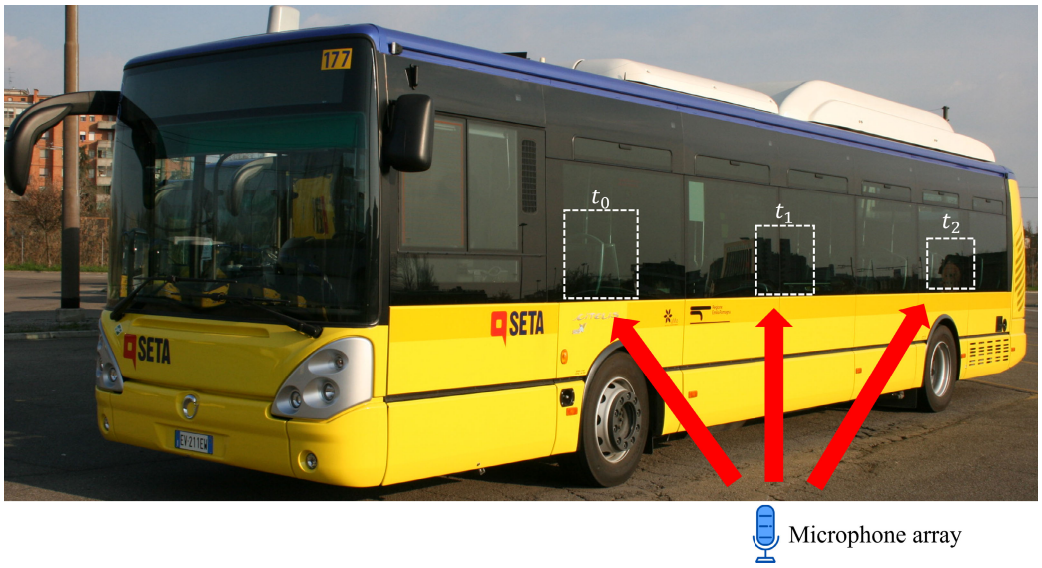


FIGURE 4. Picture of the background recording setup where (t_0, t_1, t_2) are the microphone positions.

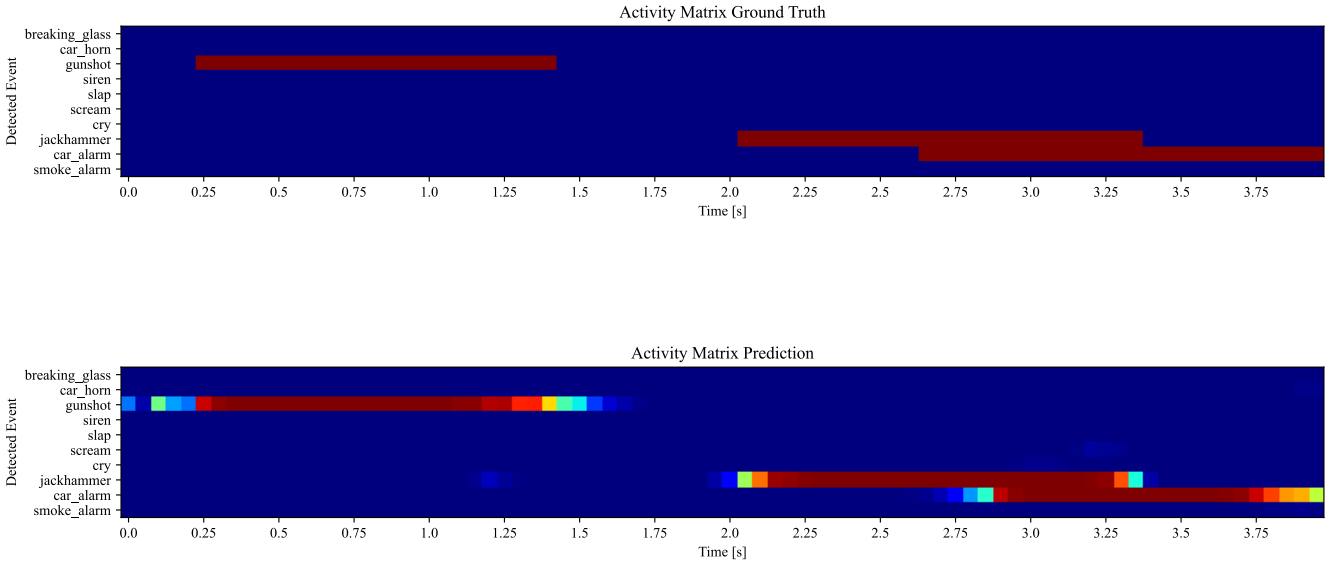


FIGURE 5. Example of a ground truth Y and a successful predicted $\hat{Y}$ from AuSPP activity matrices, respectively. In this example, two events are overlapping but the proposed model succeeds in distinguishing them with high probabilities.

FN be the number of false negatives, and TN be the number of true negatives, the classification accuracy is evaluated as:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}, \tag{12}$$

which can be decomposed into Sensitivity (S_t) and Specificity (S_p):

$$S_t = \frac{TP}{TP + FN}, \tag{13}$$

$$S_p = \frac{TN}{TN + FP}. \tag{14}$$

Inspired by the state-of-the-art [53], we discard TN from Equation (12) and we obtain:

$$Acc_M = \frac{TP}{TP + FP + FN}. \tag{15}$$

At the moment, a rigorous quantitative evaluation of SED systems is still not universally accepted from the research community. For this reason, the comparison between the output of the SED algorithm and the ground truth is performed also on fixed length time intervals, thus measuring segment-based metrics [54].

Segment-based metrics evaluate the system prediction and the reference in fixed short time segments. Thanks to this

TABLE 4. Performance of SoA models and of the proposed approach on 4 classes with threshold $\delta = 0.8$ and time frame $T = 20ms$. We denote with ↑ when the performance is better when the metric is high and ↓ otherwise.

		Segment-based							
	Parameters ↓	$F2_c \uparrow (\%)$	$ER \downarrow$	$F2_g \uparrow (\%)$	$R_c \uparrow (\%)$	$P_c \uparrow (\%)$	$Acc \uparrow (\%)$	$Acc_M \uparrow (\%)$	$S_t \uparrow (\%)$
General-purpose models									
VGG16 [49]	145.36M	68.11	0.45	70.48	56.23	95.99	94.83	55.16	99.77
VGG19 [49]	150.67M	65.18	0.48	52.91	52.91	95.84	94.46	51.91	99.77
ResNet18 [50]	14.93M	73.79	0.38	75.69	63.24	95.16	95.54	61.81	99.69
ResNet34 [50]	25.04M	67.61	0.44	69.79	56.86	94.80	94.83	55.53	99.69
ResNet50 [50]	26.21M	73.33	0.38	75.32	62.84	95.44	95.50	61.42	99.70
ResNet101 [50]	45.20M	64.72	0.48	66.60	53.28	95.28	94.47	52.10	99.73
MobileNetV2 [51]	**7.55M**	71.37	0.41	73.48	59.95	96.47	95.25	58.90	99.78
MobileNetV3 [51]	9.53M	41.60	0.69	44.38	31.67	86.01	92.16	31.03	99.83
DenseNet121 [52]	11.76M	61.96	0.49	63.83	51.27	95.31	94.29	50.25	99.76
DenseNet169 [52]	18.60M	63.83	0.48	65.84	52.89	92.04	94.42	51.70	99.73
SED models									
CNN14 [11]	83.46M	69.00	0.44	71.22	56.89	**97.12**	94.96	56.13	99.83
Pre-trained CNN14 [8], [11]	83.46M	71.09	0.42	73.16	59.34	96.26	95.22	58.59	99.81
Wavegram-LogMel-CNN [11]	82.69M	70.78	0.41	72.96	59.70	95.11	95.14	58.27	99.69
Our Model									
AuSPP w/o ASPP	7.78M	74.48	0.36	76.48	65.13	93.50	95.60	62.78	99.53
AuSPP + ASPP	7.78M	**76.12**	**0.34**	**77.67**	**67.02**	92.95	**95.76**	**64.46**	99.47

TABLE 5. Performance of SoA models and of the proposed approach on 10 classes with threshold $\delta = 0.8$ and time frame $T = 20ms$. We denote with ↑ when the performance is better when the metric value is high and ↓ otherwise.

		Segment-based							
	Parameters ↓	$F2_c \uparrow (\%)$	$ER \downarrow$	$F2_g \uparrow (\%)$	$R_c \uparrow (\%)$	$P_c \uparrow (\%)$	$Acc \uparrow (\%)$	$Acc_M \uparrow (\%)$	$S_t \uparrow (\%)$
General-purpose models									
VGG16 [49]	145.36M	49.63	0.61	54.39	37.01	89.58	97.08	36.06	99.90
VGG19 [49]	150.67M	43.46	0.67	48.56	31.53	86.27	96.85	30.81	99.92
ResNet18 [50]	14.93M	60.42	0.51	62.88	48.25	88.79	97.54	46.87	99.87
ResNet34 [50]	25.04M	28.25	0.78	30.15	21.58	51.44	96.32	20.78	99.88
ResNet50 [50]	26.21M	53.43	0.57	55.52	42.47	88.13	97.27	41.17	88.13
ResNet101 [50]	45.20M	28.10	0.78	30.44	21.30	64.67	96.36	20.56	99.92
MobileNetV2 [51]	**7.55M**	55.38	0.56	59.52	42.35	93.42	97.31	41.19	99.88
MobileNetV3 [51]	9.53M	1.19	0.99	1.48	0.74	8.35	95.47	0.72	99.99
DenseNet121 [52]	11.76M	33.94	0.74	36.24	25.70	64.88	96.51	24.48	99.86
DenseNet169 [52]	18.60M	26.21	0.80	28.04	19.50	53.42	96.28	18.97	99.94
SED models									
CNN14 [11]	83.46M	61.16	0.51	64.15	47.59	94.60	97.53	46.36	99.88
Pre-trained CNN14 [8], [11]	83.46M	**69.99**	**0.42**	**71.93**	57.05	**94.92**	**97.95**	**55.73**	99.88
Wavegram-LogMel-CNN [11]	82.69M	63.22	0.49	65.72	50.19	92.73	97.60	48.48	99.84
Our model									
AuSPP w/o ASPP	16.67M	64.14	0.47	66.48	52.57	86.85	97.56	49.32	99.69
AuSPP + ASPP	16.67M	68.21	**0.42**	70.23	**57.31**	88.15	97.77	53.77	99.69

activity representation, it is possible to define intermediate statistics like TN, TP, FP, and FN. To evaluate the robustness of the system, an activity threshold δ is introduced to the predicted activity matrix $\hat{Y}$. In the following we denote with *loose* threshold the case $\delta = 0.8$ and with *strict* threshold the case with $\delta = 0.9$. By applying one of the thresholds to the predicted activity matrix $\hat{Y}$, the resulting matrix is binary and intermediate statistics can be evaluated. Precision and Recall [55] are used for measuring the effectiveness of the retrieval.

For a generic i-th class event, the precision (P_i) and recall (R_i) are evaluated as:

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i}. \tag{16}$$

Then, class-wise Precision (P_c) and Recall (R_c) are calculated by averaging with respect to the number of class events N_{class}:

$$P_c = \mathbb{E}\left[\sum_{i=1}^{N_{class}} P_i\right], \quad R_c = \mathbb{E}\left[\sum_{i=1}^{N_{class}} R_i\right], \tag{17}$$

where $\mathbb{E}[\cdot]$ is the expected value. In addition, F-score can be derived as:

$$F_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}. \tag{18}$$

In the literature, two types of averaging approaches for the F-score are proposed [54].

We define the class-wise F_c as the average of all the F-scores:

$$F_c = \mathbb{E}\left[\sum_{i=1}^{N_{class}} F_i\right]. \tag{19}$$

We evaluate the audio-wise F-score F_a by calculating the precision P_j and the recall R_j of a j-th audio recording (ignoring the class events).

Let N_{audio} be the number of audio samples. Then, the metric is computed as:

$$F_a = \mathbb{E}\left[\sum_{j=1}^{N_{audio}} \frac{2 \cdot P_j \cdot R_j}{P_j + R_j}\right]. \tag{20}$$

TABLE 6. Performance of SoA models and of the proposed approach on 10 classes with threshold $\delta = 0.9$ and time frame $T = 20ms$. We denote with ↑ when the performance is better when the metric is high and ↓ otherwise.

	Parameters ↓	$F2_c \uparrow (\%)$	$ER \downarrow$	$F2_g \uparrow (\%)$	$R_c \uparrow (\%)$	$P_c \uparrow (\%)$	$Acc \uparrow (\%)$	$Acc_M \uparrow (\%)$	$S_t \uparrow (\%)$
General-purpose models									
VGG16 [49]	145.36M	33.76	0.77	37.01	21.97	89.02	96.46	21.85	99.98
VGG19 [49]	150.67M	26.70	0.82	29.83	16.96	84.89	96.22	16.86	99.98
ResNet18 [50]	14.93M	55.10	0.59	57.83	40.37	96.95	97.27	39.92	99.95
ResNet34 [50]	25.04M	16.52	0.88	17.56	11.49	43.86	95.95	11.34	99.97
ResNet50 [50]	26.21M	47.58	0.64	49.49	35.22	85.82	97.02	34.76	99.95
ResNet101 [50]	45.20M	11.37	0.92	12.10	7.98	36.98	95.80	7.91	99.99
MobileNetV2 [51]	**7.55M**	41.39	0.71	44.71	28.18	96.15	96.74	27.97	99.97
MobileNetV3 [51]	9.53M	3.76	0.98	4.14	2.23	16.53	95.54	2.23	99.99
DenseNet121 [52]	11.76M	19.74	0.86	21.00	13.98	49.00	96.07	13.84	99.98
DenseNet169 [52]	18.60M	18.30	0.87	19.47	12.61	42.46	96.00	30.36	99.97
SED models									
CNN14 [11]	83.46M	44.58	0.69	47.36	30.57	97.55	96.84	30.36	99.97
Pre-trained CNN14 [8], [11]	83.46M	56.37	0.58	58.87	41.31	**97.58**	97.32	41.06	99.97
Wavegram-LogMel-CNN [11]	82.69M	48.09	0.65	51.00	33.93	96.73	96.97	33.57	99.95
Our model									
AuSPP w/o ASPP	16.67M	58.59	0.54	61.17	45.12	90.35	97.35	43.47	99.82
AuSPP + ASPP	16.67M	**59.49**	**0.53**	**61.98**	**46.05**	91.13	**97.40**	**44.40**	99.83

TABLE 7. Performance of SoA models and of the proposed approach on 10 classes with threshold $\delta = 0.9$ and time frame $T = 50ms$. We denote with ↑ when the performance is better when the metric is high and ↓ otherwise.

	Parameters ↓	$F2_c \uparrow (\%)$	$ER \downarrow$	$F2_g \uparrow (\%)$	$R_c \uparrow (\%)$	$P_c \uparrow (\%)$	$Acc \uparrow (\%)$	$Acc_M \uparrow (\%)$	$S_t \uparrow (\%)$
General-purpose models									
VGG16 [49]	145.36M	35.28	0.76	38.69	23.30	87.77	96.42	23.13	99.97
VGG19 [49]	150.67M	31.93	0.78	35.34	20.95	88.04	96.30	20.79	99.98
ResNet18 [50]	14.93M	49.55	0.63	52.27	36.18	91.85	97.00	35.82	99.96
ResNet34 [50]	25.04M	20.21	0.86	21.39	14.12	48.25	95.92	13.90	99.93
ResNet50 [50]	26.21M	53.39	0.59	55.51	39.97	94.83	97.15	39.47	99.94
ResNet101 [50]	45.20M	31.86	0.76	33.47	23.17	66.65	96.39	22.92	99.97
MobileNetV2 [51]	**7.55M**	42.26	0.70	45.58	28.85	95.93	95.68	28.63	99.97
MobileNetV3 [51]	9.53M	1.79	0.989	2.00	1.07	12.38	95.36	1.07	99.99
DenseNet121 [52]	11.76M	32.33	0.77	34.61	21.92	81.71	96.34	21.74	99.97
DenseNet169 [52]	18.60M	33.28	0.76	35.34	23.60	70.91	96.41	23.39	99.97
SED models									
CNN14 [11]	83.46M	44.66	0.68	47.49	30.66	**97.91**	96.76	30.47	99.97
Pre-trained CNN14 [8], [11]	83.46M	56.42	0.58	58.80	41.34	97.68	97.25	41.09	99.97
Wavegram-LogMel-CNN [11]	82.69M	46.24	0.67	48.75	32.06	96.86	96.80	31.78	99.96
Our model									
AuSPP w/o ASPP	7.78M	59.89	**0.53**	62.57	**46.50**	91.84	97.36	44.94	99.84
AuSPP + ASPP	7.78M	**59.96**	**0.53**	**62.61**	46.41	92.40	**97.37**	**44.97**	99.85

Furthermore, we use the Error Rate (ER) metric to account the amount of wrong predictions in terms of substitution, deletion, and insertion errors. More precisely, let k be a specific time-segment, with its intermediate statistics (TP_k, TN_k, FP_k, and FN_k), in the j-th audio file with K time-frames. We can define the errors as:

$$\begin{aligned} S_k &= \min(FN_k, FP_k), \\ D_k &= \max(0, FN_k - FP_k), \\ I_k &= \max(0, FP_k - FN_k). \end{aligned}$$

The total ER of the j-th audio file is evaluated as:

$$ER_j = \frac{\sum_{k=1}^{K} S_k + \sum_{k=1}^{K} D_k + \sum_{k=1}^{K} I_k}{\sum_{k=1}^{K} N_k}. \tag{21}$$

Finally, ER is averaged with respect to the number of audio files in the validation set:

$$ER = \mathbb{E}\left[\sum_{j=1}^{N_{audio}} ER_j\right]. \tag{22}$$

It is worth noticing that the ER is not a probability so its value can be bigger than 1. However, thanks to the use of the activity threshold δ and the fact that a zero predicted activity matrix yields unit ER, all the approaches used for comparisons have an ER lower or equal than 1.

B. RESULTS

We present the results of four tests where the ability to generalize and the robustness of the proposed framework is compared with respect to state-of-the-art techniques [11], [49], [50], [51], [52]. Since SEDDOB is split into 10 folds, we provide the mean values for the aforementioned metrics.

As shown in Tables 4, 5, 6, 7, some metrics such as Acc, S_t, P_t are not meaningful for rare anomalous events activity since they are biased by the high value of TN. Moreover they do not take into account: i) the fact that multiple class events must be recognized in the same time frame (*class errors*) and ii) the wrong predictions of onset and offset timestamps for a correct class event (*time errors*).

1) REDUCED DATASET, SMALL TIME FRAME, AND LOOSE THRESHOLD

This test is performed for evaluating the AuSPP performance on a reduced number of audio classes with an high time-resolution. In more details, 10.000 audio tracks with 4 anomalous events (*breaking_glass, car_horn, gunshot, and siren*) are used for training and testing. The temporal resolution is set to 20*ms* and the *loose* threshold δ is set to 0.8 to filter low probabilities from the predicted activity matrix $\hat{Y}$. As shown in Table 4, AuSPP outperforms both general and SED deep learning approaches for all the metrics, except for the precision metric.

However, for a security-driven system, the proposed model is preferable since it has a larger recall than the state-of-the-art CNN14 [11], with an improvement of 7.68%. From the results, the model pre-trained on AudioSet [8] does not outperform the other models even if it has been trained on a large annotated audio dataset. This behaviour could be caused by the reduced number of audio events to be classified. As a drawback, AuSPP shows a smaller value of precision than the other models. More specifically, predicted onset and offset timestamps of events from AuSPP are less accurate, as it can be seen in Figure 5. This can be due to the low complexity of the model with respect to state-of-the-art models.

2) FULL DATASET, SMALL TIME FRAME, AND LOOSE THRESHOLD

In this case, a dataset of 10.000 audio samples is generated with all 10 audio class events. Hence, AuSPP ability to recognize multiple class anomalies is tested together with the other models. The results reported in Table 5 show that AuSPP achieves comparable results with respect to the version of CNN14 [11] pre-trained on AudioSet [8]. It is worth noticing that AuSPP, with 75% less parameters than CNN14, does not require additional data. Therefore, AuSPP can be employed in edge computing devices where computational resources are limited.

3) FULL DATASET, SMALL TIME FRAME, AND STRICT THRESHOLD

Exploiting the aforementioned extended dataset, this test allows to analyze the predictions of all the models by adopting the *strict* threshold ($\delta = 0.9$). With this test, we evaluate which model is more confident, i.e. higher probability, of its predicted activity matrix. Table 6 shows that the proposed AuSPP outperforms state-of-the-art approaches with a Recall improvement of 4.74% over CNN14 [11], the second best model. Similarly with the other tests, our model is less precise with respect to the state-of-the-art pre-trained model on AudioSet.

4) FULL DATASET, LARGE TIME FRAME, AND STRICT THRESHOLD

Finally, we increase the time-frame from 20*ms* to 50*ms* and re-train all models in order to assess their performances. As shown in Table 7, AuSPP outperforms state-of-the-art SED models with a recall improvement of 5.16%. It is worth noticing that the proposed method achieves better performance without the ASPP module. Hence, using a larger time frame, it is preferable to avoid using the module since no improvements can be obtained.

VI. CONCLUSION

In this paper we propose AuSPP, a lightweight deep learning model that applies spatial filters to Mel-Spectrogram in order to predict different anomalies class with the corresponding onset and offset time. Moreover, we introduce SEDDOB, a human safety-oriented dataset which provides high quality annotated audio waveforms for detecting anomalies on buses. To assess the performance of AuSPP, four tests on SEDDOB demonstrate that our AuSPP outperforms state-of-the-art general-purpose deep learning approaches with a reduced number of audio event classes. In addition, AuSPP achieves comparable results with respect to SED models pre-trained on large scale audio datasets with fewer learnable parameters.

However, the results show that significant improvements can be achieved. All considered models suffer from false alarms on normal recordings, significantly impacting on the task of ensuring human safety. As a possible solution, further studies on a coarse-to-fine approach for increasing the quality of audio prediction - in terms of recall and F-score metrics - could be performed. Moreover, tests could be conducted on a real scenario exploiting edge computing devices. Finally, model interpretability could be performed.

ACKNOWLEDGMENT

The article reflects only the authors' view and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] National Institutes of Health, Bethesda, MD, USA. (2007). *Information About Hearing, Communication, and Understanding*. Accessed: Jul. 27, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK20366

[2] A. Mesaros, A. Diment, and B. Elizalde, "Sound event detection in the DCASE 2017 challenge," *IEEE Trans. Audio, Speech, Language Process.*, vol. 27, no. 6, pp. 992–1006, Jun. 2019.

[3] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in DCASE 2019," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 684–698, 2021.

[4] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 2, pp. 379–393, Feb. 2018.

[5] F. Ronchini and R. Serizel, "A benchmark of state-of-the-art sound event detection systems evaluated on synthetic soundscapes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 1031–1035.

[6] X. Xia, R. Togneri, F. Sohel, Y. Zhao, and D. Huang, "Multi-task learning for acoustic event detection using event and frame position information," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 569–578, Mar. 2020.

[7] S. Park, D. K. Han, and M. Elhilali, "Cross-referencing self-training network for sound event detection in audio mixtures," *IEEE Trans. Multimedia*, early access, May 27, 2022, doi: 10.1109/TMM.2022.3178591.

[8] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 776–780.

[9] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 1041–1044.

[10] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 1015–1018.

[11] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2880–2894, 2020.

[12] H. Malik, "Acoustic environment identification and its applications to audio forensics," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1827–1837, Nov. 2013.

[13] D. M. Ballesteros, Y. Rodriguez-Ortega, D. Renza, and G. Arce, "Deep4SNet: Deep learning for fake speech classification," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115465. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421008770

[14] M. Simonović, M. Kovandžić, I. Ćirić, and V. Nikolić, "Acoustic recognition of noise-like environmental sounds by using artificial neural network," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115484. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421008952

[15] S. Mao, P. C. Ching, and T. Lee, "Enhancing segment-based speech emotion recognition by iterative self-learning," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 123–134, 2022.

[16] S. Sardari, B. Nakisa, M. N. Rastgoo, and P. Eklund, "Audio based depression detection using convolutional autoencoder," *Expert Syst. Appl.*, vol. 189, Mar. 2022, Art. no. 116076. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421014147

[17] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 9, pp. 1652–1664, Sep. 2016.

[18] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[19] H. Zhao and H. Malik, "Audio recording location identification using acoustic environment signature," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1746–1759, Nov. 2013.

[20] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2019, pp. 313–317.

[21] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," in *Proc. Detection Classification Acoust. Scenes Events*, 2019, pp. 209–213.

[22] Z. Zhang, D. Liu, J. Han, K. Qian, and B. W. Schuller, "Learning audio sequence representations for acoustic event classification," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 115007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421004486

[23] P. Laffitte, Y. Wang, D. Sodoyer, and L. Girin, "Assessing the performances of different neural network architectures for the detection of screams and shouts in public transportation," *Expert Syst. Appl.*, vol. 117, pp. 29–41, Mar. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418305657

[24] T. K. Chan and C. S. Chin, "A comprehensive review of polyphonic sound event detection," *IEEE Access*, vol. 8, pp. 103339–103373, 2020.

[25] L. C. Chen, G. Papandreou, and I. Kokkinos, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[26] D. Symes, "Automatic vehicle monitoring: A tool for vehicle fleet operations," *IEEE Trans. Veh. Technol.*, vol. VT-29, no. 2, pp. 235–237, May 1980.

[27] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP J. Audio, Speech, Music Process.*, vol. 2013, no. 1, pp. 1–13, Dec. 2013.

[28] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, and K. Takeda, "BLSTM-HMM hybrid system combined with sound activity detection network for polyphonic sound event detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 766–770.

[29] V. Bisot, S. Essid, and G. Richard, "Overlapping sound event detection with supervised nonnegative matrix factorization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 31–35.

[30] W. Ding and L. He, "Adaptive multi-scale detection of acoustic events," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 294–306, 2020.

[31] C. C. Chatterjee, M. Mulimani, and S. G. Koolagudi, "Polyphonic sound event detection using transposed convolutional recurrent neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 661–665.

[32] Z. Shuyang, T. Heittola, and T. Virtanen, "Active learning for sound event detection," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2895–2905, 2020.

[33] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with CNN-transformer and automatic threshold optimization," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2450–2460, 2020.

[34] K. Wakayama and S. Saito, "CNN-transformer with self-attention network for sound event detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 806–810.

[35] T. K. Chan and C. S. Chin, "Multi-branch convolutional macaron net for sound event detection," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 2972–2985, 2021.

[36] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 829–852, 2022.

[37] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. 21st ACM Int. Conf. Multimedia*, Oct. 2013, pp. 411–412.

[38] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound Datasets: a platform for the creation of open audio datasets," in *Int. Soc. for Music Inf. Retr. Conf.*, 2017.

[39] N. Turpault, R. Serizel, S. Wisdom, H. Erdogan, J. R. Hershey, E. Fonseca, P. Seetharaman, and J. Salamon, "Sound event detection and separation: A benchmark on desed synthetic soundscapes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 840–844.

[40] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2017, pp. 344–348.

[41] J. Abeßer, "USM-SED—A dataset for polyphonic sound event detection in urban sound monitoring scenarios," 2021, *arXiv:2105.02592*.

[42] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE J. Solid-State Circuits*, vol. SSC-23, no. 2, pp. 358–367, Apr. 1988.

[43] X. Wang, "Laplacian operator-based edge detectors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 886–890, May 2007.

[44] C. Djork-Arne, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–14.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[47] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "SpecAugment on large scale datasets," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6879–6883.

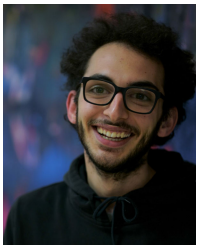
[48] I. Jordal, K. Nishi, H. Bredin, F. Lata, H. C. Blum, P. Manuel, A. Raj, K. Choi, P. Zelasko, M. La Quatra, and E. Schmidbauer, "Torch-audiomentations," Zenodo, Tech. Rep., 2022, doi: 10.5281/zenodo.6778064.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[51] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[53] S. Dixon, "On the computer recognition of solo piano music," in *Proc. Arts Cultural Manage. Conf.*, 2000, pp. 31–37.
[54] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Appl. Sci.*, vol. 6, no. 6, p. 162, May 2016.
[55] C. J. Van Rijsbergen, *Information Retrieval*. Newton, MA, USA: Butterworth, 1979.



MICHAEL NERI (Graduate Student Member, IEEE) received the Laurea (B.Sc.) degree in information engineering and the Laurea Magistrale (M.Sc.) degree from the University of Padova, in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree in applied electronics with the Department of Industrial, Electronic, and Mechanical Engineering, Roma Tre University. His main research interests include computer vision, deep learning, and audio processing.

FEDERICA BATTISTI (Senior Member, IEEE) is an Associate Professor with the Department of Information Engineering, University of Padova. Her research interests include multimedia quality assessment and security. She serves as an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA, *EURASIP Journal on Image and Video Processing*, and *Signal Processing: Image Communication* (Elsevier).

ALESSANDRO NERI (Life Member, IEEE) received the doctoral degree (cum laude) in electronic engineering from the Università degli Studi di Roma "La Sapienza", Roma, Italy, in 1977. In 1978, he joined the Research and Development Department, Contraves Italiana S.p.A., Rome. In 1987, he joined as an Associate Professor of signal and information theory at the INFOCOM Department, Engineering Faculty, Sapienza University of Rome. In November 1992, he joined the Department of Electronic Engineering, Roma Tre University, Rome, as an Associate Professor of electrical communications, and became a Full Professor of telecommunications, in September 2001.

MARCO CARLI (Senior Member, IEEE) received the Laurea degree in telecommunication engineering from the Università degli Studi di Roma "La Sapienza", Rome, Italy, and the Ph.D. degree from theTampere University of Technology, Tampere, Finland. He was a Visiting Researcher at the Image Processing Laboratory, UCSB, University of California, Santa Barbara, CA, USA, from 2000 to 2004. He is an Associate Professor with the Università degli Studi Roma Tre, Rome. He has been an Associate Editor of *EURASIP Journal on Image and Video Processing*, since 2011, and an Area Editor of *Signal Processing: Image Communication* (Elsevier).

Open Access funding provided by 'Università degli Studi Roma Tre' within the CRUI CARE Agreement