Applied Acoustics 211 (2023) 109489

Contents lists available at ScienceDirect

Applied Acoustics

journal homepage: www.elsevier.com/locate/apacoust

Lightweight deep neural networks for acoustic scene classification and an effective visualization for presenting sound scene contexts

Lam Pham [a], Dat Ngo [b], Dusan Salovic [c], Anahid Jalali [a], Alexander Schindler [a], Phu X. Nguyen [d], Khoa Tran [e], Hai Canh Vu [f,g,*]

[a] Center for Digital Safety & Security, Austrian Institute of Technology (AIT), Vienna, Austria
[b] School of Computer Science and Electronic Engineering, Essex University, Colchester, UK
[c] Faculty of Informatics, Vienna University of Technology, Vienna, Austria
[d] Department of Computing Fundamentals, FPT University, Ho Chi Minh City 700000, Viet Nam
[e] International Research Institute for Artificial Intelligence and Data Science, Dong A University, Danang, Viet Nam
[f] Laboratory for Applied and Industrial Mathematics, Institute for Computational Science and Artificial Intelligence, Van Lang University, Ho Chi Minh City, Viet Nam
[g] Faculty of Mechanical - Electrical and Computer Engineering, School of Technology, Van Lang University, Ho Chi Minh City, Viet Nam

ARTICLE INFO

Article history:
Received 27 November 2022
Received in revised form 16 May 2023
Accepted 9 June 2023
Available online 24 June 2023

Keywords:
Acoustic scene classification
Sound scene
Sound event
Residual-inception architecture
Deep neural network

ABSTRACT

In this paper, we propose lightweight deep neural networks for Acoustic Scene Classification (ASC) and a visualization method for presenting a sound scene context. To this end, we first propose an inception-based and low-memory footprint ASC model as the ASC baseline. The ASC baseline is then compared with benchmark and high-complexity network architectures. Next, we improve the ASC baseline by proposing a novel deep neural network architecture which leverages a residual-inception architecture and multiple kernels. Given the novel residual-inception (NRI) based model, we apply multiple techniques of model compression to evaluate the trade off between the model complexity and the model accuracy performance. Finally, we evaluate whether sound events detected in a sound scene recording can help to improve ASC accuracy performance and to present the sound scene context more comprehensively. We conduct extensive experiments on various ASC datasets, including sound scene datasets proposed for IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) 2018 Task 1A and 1B, 2019 Task 1A and 1B, 2020 Task 1A, 2021 Task 1A, and 2022 Task 1. Our experimental results on several different ASC challenges highlight two main achievements. First, given the analysis of the trade off between the model performance and the model complexity, we propose two low-complexity ASC models: The medium-size model (MM) presents 4.96 M trainable parameters, 19.3 MB memory occupation, and 7.12 BFLOPs; The small-size model (SM) presents a very low complexity of 120 K trainable parameters, 120 KB memory occupation, and 0.82 BFLOPs. These ASC systems are very competitive to the state-of-the-art systems and compatible for real-life applications on a wide range of edge devices. Secondly, from the analysis of the role of sound events in a sound scene, we propose an effective visualization method for comprehensively presenting a sound scene context. By combining both the sound scene and sound event information, the visualization method not only indicates predicted sound scene contexts with high probabilities but also provides statistics of sound events occurring in these sound scene contexts.

1. Introduction

Acoustic Scene Classification (ASC), one of two main tasks of the machine hearing research [1], aims at detecting surrounding environments such as *'in a bus'*, *'in an shopping mall'*, or *'on a street'*. By detecting the current sound scene context, edge devices could make use of this useful information to enable them to respond appropriately or adjust certain functions, then opening up various applications: to integrate an ASC component into a robotic system

* Corresponding author at: Van Lang University, 69/68, Dang Thuy Tram, Ward 13, Binh Thanh District, Ho Chi Minh City, Vietnam.
E-mail addresses: Lam.Pham@ait.ac.at (L. Pham), dn22678@essex.ac.uk (D. Ngo), dusan.salovic@gmail.com (D. Salovic), SeyedehAnahid.Naghibzadehjalali@ait.ac.at (A. Jalali), Alexander.Schindler@ait.ac.at (A. Schindler), phunx4@fpt.edu.vn (P.X. Nguyen), khoatd@donga.edu.vn (K. Tran), canh.vuhai@vlu.edu.vn (H.C. Vu)

https://doi.org/10.1016/j.apacoust.2023.109489
0003-682X/

[2], a mobile application [3], or a sensor system [4] as one of main functions; to support sound event detection when these sound events are mixed in real-world environments [5]. Considering a general recording of an acoustic environment, it contains not only a background sound field but also various foreground events. Both background and foreground contain true noise–continuous, periodic or aperiodic acoustic signals that interfere with the understanding of the scene. If the background is considered as the noise and the foreground is referred to as the signal, it can be seen that the signal-to-noise ratio exhibits high variability due to the diverse range of environments or recording conditions. To further complicate matters, if a sound event can occur in a long time, it could be considered as background in certain contexts. For example, an audio recording *'on pedestrian street'* may present a quiet background, but the sound of the *'engine'* of traffic passes is considered as the foreground events. However, a lengthy *'engine'* sound in an *'on train'* recording would be considered a background sound. Furthermore, the issue of mismatched recording devices [6,7] causes very different distribution of energy across the frequency dimensions of audio spectrograms from the same acoustic scene [8], which leads classification models to classify incorrectly. All these challenges mentioned make acoustic scene classification (ASC) task particularly challenging.

To deal with the ASC challenges recently mentioned, researches on the ASC task have tended to focus on two main approaches. The first aims at solving the lack of discriminative information by exploiting various methods of low-level feature extraction. In particular, an input audio is transformed into various two-dimensional spectrogram representations. Then, these spectrograms are independently trained with back-end deep learning models. Finally, independent models' results are fused to achieve the best performance. For instances, log-Mel spectrogram was combined with constant-Q transform (CQT) [9], Gammatone-like spectrogram (GAM) [10], or draw audio [11]. To evaluate a wavelet-transform derived spectrogram representation, Ren et al. [12] compared results from STFT spectrograms and a combination of *Bump* and *Morse* scalograms. By exploiting channel information, Sakashita and Aono [13] generated multi-spectrogram inputs from two channels, the average and side channels, and even explored separated harmonic and percussive spectrograms from mono channels. The approach of using multiple spectrograms has proven powerful to tackle the issue of mismatched recoding devices. Indeed, a combination of log-Mel and Mel-based nearest neighbor filter (NNF) spectrograms in [14] helps to achieve the top-1 on DCASE 2018 Task 1B blind Test set and the top-4 on DCASE 2018 Task 1B Development set. Meanwhile, the authors in [15] conducted various ensemble methods on log-Mel, GAM, CQT, and MFCC spectrograms, then achieved the top-6 on DCASE 2020 Task 1A blind Test set and the top-1 on DCASE 2020 Task 1A Development set. Although the approach of multiple spectrogram inputs shows effective to deal with the ASC challenges, it presents the issue of large memory footprint as using ensemble of multiple classifiers (i.e. The memory footprint is the number of Byte on a target device's memory which the trainable parameters of a model occupy).

Instead of using multiple spectrogram inputs, the second approach tends to deploy more complex deep learning architectures, especially focusing on exploring the frequency bands of audio spectrograms. For instances, authors in [16] split the entire log-Mel spectrograms into three sub spectrograms across the frequency dimension. Then, each sub spectrogram was learned by a ResNet-based network architecture before concatenating together. Meanwhile, Phaye et al. [17] proposed a SubSpectralNet network which comprises multiple sub-networks with parallel branches to extract discriminative information from 30 sub log-Mel spectrograms. However, to achieve the best performance, some papers from the second approach have still applied ensemble methods of multiple models [18–22], which increases the model complexity.

It can be seen that although both multiple spectrogram input and complex network approaches show effective to deal with the ASC challenges, these approaches present the issue of large model footprint. These high-complexity ASC models prevent to integrate these models into edge devices or mobiles with a memory limitation. Recently, the issue of low-complexity model within the ASC task have been indicated in [23,24] as a new challenge of the ASC task. To deal with the issue of large memory footprint as using complex network architectures, ensemble of multiple models, or ensemble of multiple spectrogram inputs, researches on the ASC task can be separated into two main groups. The first research group much focuses on the network architecture. For instances, authors in [25] proposed a lightweight TC-SKNet network for the ASC task which takes advantages from temporal convolution and the Selective Kernel Networks [26]. Similarly, a multi-kernel and separable convolution base architecture was proposed in [27], which achieved the top-3 on DCASE 2022 Task 1 blind Test set. Focus on frequency normalization, authors in [28] proposed a novel Residual Normalization method and a residual-based network architecture, which showed effective to improve the ASC performance and achieved the top-1 on DCASE 2021 Task 1A blind Test set and the top-4 on DCASE 2021 Task 1A Development set, but still satisfied the challenge requirement of less than 120 KB memory footprint occupation. Meanwhile, the second group leverages a wide range of model compression techniques to reduce the model size. Among the model compressions, the pruning [18,19,29,30] and quantization [29,20] techniques have been widely applied. While quantization techniques feasibly help the model reduce to 1/4 of the original size (i.e. 32 bit with floating point format presenting for 1 trainable parameter is quantized to 8 bit with integer format [31]), pruning techniques prove that models can be reduced to 1/10 of the original sizes [29]. Recently, teacher-student schemes have proven effective to achieve a low-complexity student which still performs well the ASC task. Indeed, ASC systems [32,33], which achieved the top-1 and top-2 of DCASE 2022 Task 1, made use of this scheme.

Looking at the recent approaches surveyed above, we can see that: **(I)** While ensembles of multiple spectrograms or complex network architectures can help to enhance the ASC performance as well as effectively to deal with the mismatched recording devices, these approaches present large memory footprint models which are not compatible for applications on edge devices or mobiles. However, recent research [30,34], which provided the analysis of trade off between the ASC system performance and the ASC system complexity, much focused on pruning techniques rather than other model compression techniques. Although pruning techniques prove to reduce the model complexity significantly, the pruning parameters are not removed from the proposed network architecture and they still occupy the memory of edge devices that leads the cost computation same as the non-pruning parameters. Therefore, the recent DCASE 2021 Task 1A and DCASE 2022 Task 1 challenges [24], which focus on the issue of low-complexity ASC model, require not to use pruning techniques. As a result, an analysis of the trade off between the ASC system performance and the ASC system complexity without using the pruning techniques is necessary. **(II)** While target devices integrating ASC function present a wide range memory capacity (i.e. High-performance computers present large memory with more than GB; Applications on mobiles require a memory occupation of around 20 MB [35,36]; Embedded devices such as STM32L496@80 MHz or Arduino Nano 33@64 MHz show a limitation memory with the maximum 256 KB), recently proposed low-complexity ASC systems have not been indicated to be compatible

for certain target devices. Additionally, although both the number of trainable parameters and the number of floating point operations (FLOPs) reflect the model complexity on a certain device, almost state-of-the-art ASC systems report the number of trainable parameters without FLOPs number. **(III)** As a sound scene can contain different types of sound events and some sound events are distinct for certain sound scene, this inspires that exploring sound event information in a sound scene recording can help to further improve the ASC performance. However, just a few of researches [37,38] leveraged sound event information for enhancing ASC systems and none of research has deeply analyzed the relationship and correlation between sound scene and sound events.

Therefore, in this work, we aim to fill these three gaps of the ASC research and further describe our main contributions below:

1. By evaluating various neural network architecture, we indicate that a shallow and wider inception based network is more effective rather than other deeper architectures with a trunk of convolutional layers for the ASC task. Inspired by the efficiency of inception based network for the ASC task, we propose a novel residual-inception (NRI) based network architecture. We then combine the NRI model with multiple spectrogram inputs and different model compression techniques to propose a comprehensive analysis of the trade off between the model accuracy performance and the model complexity (Notably, the pruning technique is not used for the model compression in this paper).
2. Given the analysis of the trade off between the model performance and the model complexity, we propose different ASC models presenting a wide range of model complexity, which are very competitive to the state-of-the-art ASC systems and potential to apply on various target devices. In particular, three ASC models are proposed: The first large-size model (LM) with 12.9 M trainable parameters, 49.8 MB memory occupation, and 43.68 BFLOPs is suitable for applications running on high-performance computers; The second medium-size model (MM) with 4.96 M trainable parameters, 19.3 MB memory occupation, and 7.12 BFLOPs which satisfies a wide range of edge devices and mobiles surveyed in [35,36]; The third small-size model (SM) with 120 K trainable parameters, 120 KB memory occupation, and 0.82 BFLOPs which is suitable for very limited-memory embedded systems such as STM32L496@80 MHz or Arduino Nano 33@64 MHz.
3. We comprehensively evaluate the role of sound events in a sound scene recording, indicate how pre-trained models on the task of acoustic event detection (AED) can help to improve the ASC performance. Given the analysis of the role of sound events, we provide an effective visualization method to present a sound scene context more comprehensively. In particular, the visualization method is evaluated in a use case of detecting and presenting a riot context in this paper. In the use case experiment, not only statistics of detected sound events in each 5-s duration of sound scene recordings are presented, but distinct sound events in a riot context are also indicated. From the proposed use case, we proves that our proposed visualization method is very effective for detecting and presenting a sound scene context. Additionally, as sound scene and sound events information used in the proposed visualization method are obtained from the medium-size model (MM) with less than 20 MB memory occupation, the visualization method is also feasible to be integrated in a wide range of edge devices or mobiles.

Rather than selecting a single task, we evaluate over a wide range of datasets of: Crowded-Scene [39], DCASE 2018 Task 1A and 1B [40], DCASE 2019 Task 1A and 1B [41], DCASE 2020 Task1A [42], DCASE 2021 Task 1A [42], and DCASE 2022 Tasks 1 [42]. We will see that the performance of our proposed system is competitive with the state-of-the-art systems.

2. Evaluating datasets

To select ASC datasets for evaluating our proposed models in this paper, we first analyze all published and real-life-recording audio datasets since 2010 as shown in Table 1. As Table 1 shows, the audio datasets can be separated into two main groups. The first dataset group of DEMAND [43], UrbanSound8K [44], Freefield1010 [45], ESC-50 [46], ESC-10 [46], CHIME-Home [47], Youtube-8 M [48], MSoS [49], and AudioSet [50] was mainly proposed for the sound source detection or the sound event detection, referred to as Acoustic Event Detection (AED). In these datasets for the AED task, an audio recording was labeled by one or multiple sound events (i.e. The piano, the human speech, etc.) occurring in the recording. Meanwhile, the remaining datasets in Table 1 were proposed for the task of sound scene classification, referred to as Acoustic Scene Classification (ASC). An audio recording used for the ASC task was labeled by the place (i.e. On bus, in park, etc.) where the audio file was recorded. Among the datasets for the ASC task, DCASE 2013 Scenes, TUT Sound Scenes 2016, and TUT Acoustic Scenes 2017 present a limitation of recording time less than 15 h. Meanwhile, the datasets for the ASC task since the year of 2018 show more than 20-h recording time. As the result, we evaluate our proposed ASC systems on TUT Urban Acoustic Scenes 2018 [40], TUT Urban Acoustic Scenes 2018 Mobile [40], TAU Urban Acoustic Scenes 2019 [41], TAU Urban Acoustic Scenes 2019 Mobile [41], TAU Urban Acoustic Scenes 2020 Mobile [42], and Crowded Scenes [39] (Notably, although LITIS Rouen presents the total recording time of 25.2 h, the link for downloading this dataset has been invalid). Additionally, as each following section in this paper not only describes ASC systems but also evaluates and discusses experimental results, we first present the selected ASC datasets in detail and indicate why and which datasets are evaluated in certain sections.

Crowded Scenes (Cr-Sc) [39]: contains 341 videos collected from YouTube (in-the-wild scenes), which presents a total recording time of nearly 29.1 h. These videos were then split into 10-s video segments, each of which was annotated by one of five categories: *'Riot', 'Noise-Street', 'Firework-Event', 'Music-Event',* or *'Sport-Atmosphere'*. Notably, 10-s video segments split from an original video are not presented in both Train and Test subsets to make the data distribution different between these two subsets. In this paper, we extract audio recordings from these video segments and follow the splitting method as proposed in [39] to evaluate this dataset[1].

TUT Urban Acoustic Scenes 2018 [40] and TAU Urban Acoustic Scenes 2019 [41] Development sets: TUT Urban Acoustic Scenes 2018 Development set, which was proposed for DCASE 2018 Task 1A and referred to as DC-18-1A. The dataset was recorded from one device, referred to as the device A. TAU Urban Acoustic Scenes 2019 Development set, which was proposed for DCASE 2019 Task 1A and referred to as DC-19-1A, reused all DC-18-1A dataset and more data was added (i.e. Additional audio recordings were also recorded on the same device A). As audio recordings from these both datasets are from one device A, they are used to evaluate the ASC task regardless of the issue of mismatched recording devices.

TUT Urban Acoustic Scenes 2018 Mobile [40] and TAU Urban Acoustic Scenes 2019 Mobile [41] Development sets: TUT Urban Acoustic Scenes 2018 Mobile Development set, which was pro-

[1] https://zenodo.org/record/5774751#.Ybc9R5pKhhE

Table 1
Real-life recording and publishing audio datasets since 2010.

Year	Name	Classes	Samples	Duration (hours)
2013	DEMAND [43]	18	18 (300s)	1.5
2013	DCASE 2013 Scenes [51]	10	100 (30s)	0.83
2014	UrbanSound8K [44]	10	8732 (4s)	9.7
2014	Freefield1010 [45]	7	7690	21.3
2015	LITIS Rouen [52]	19	3026 (30s)	25.2
2015	ESC-50 [46]	50	2000 (5s)	2.78
2015	ESC-10 [46]	10	400 (5s)	0.56
2015	CHIME-Home [47]	7	6137 (4s)	6.8
2016	Youtube-8 M [48]	3862	>6.1 M	> 0.3 M
2016	TUT Sound Scene 2016 [53]	15	1170 (30s)	9.75
2017	AudioSet [50]	527	2.1 M	5800
2017	TUT Sound Scene 2017 [53]	15	4680 (30s)	13
2018	MSoS [49]	5	2000 (5s)	2.78
2018	TUT Urban Acoustic Scenes 2018 [40]	10	86400 (10s)	24
2018	TUT Urban Acoustic Scenes 2018 Mobile [40]	10	10080 (10s)	28
2019	TAU Urban Acoustic Scenes 2019 [41]	10	14400 (10s)	40
2019	TAU Urban Acoustic Scenes 2019 Mobile [41]	10	16560 (10s)	46
2020	TAU Urban Acoustic Scenes 2020 Mobile [42]	10	23040 (10s)	64
2022	Crowded Scenes [39]	5	10460 (10s)	29.1

posed for DCASE 2018 Task 1B and referred to as DC-18-1B, reused all DC-18-1A dataset recently mentioned, and added more data recorded from two other devices, referred to as the device B and the device C. Similarly, TAU Urban Acoustic Scenes 2019 Mobile Development set, which was proposed for DCASE 2019 Task 1B and referred to as DC-19-1B, also reused DC-19-1A dataset, and 3 h of recording time on each device B and C were further added. As audio recordings from these both datasets are from three different devices (A, B, and C) with a limitation of recording time on device B and C, these datasets are used to evaluate the ASC task concerning the issue of mismatched recording devices.

TAU Urban Acoustic Scenes 2020 Mobile Development set [42]: was proposed for DCASE 2020 Task 1A and referred to as DC-20-1A. This is currently the largest dataset proposed for the ASC task. In particular, DC-19-1B set with 40 h, 3 h, and 3 h recorded on devices A, B, and C is reused in DC-20-1A. Additionally, synthesized audio recordings namely from S1 to S6 with 3-h recording time for each synthesized device were added. Notably, audio recordings from S4, S5, and S6 are not presented in Training subset to evaluate unseen samples. As audio recordings are from both different real and synthesized devices, this dataset is proposed to evaluate the ASC task with the issue of mismatched recording devices.

This dataset is also used in **DCASE 2021 Task 1A (DC-21-1A) and DCASE 2022 Task 1 (DC-22–1)** challenges for evaluating both issues of mismatched recording devices and low-complexity models. While DCASE 2021 Task 1A challenge evaluates low-complexity models on 10-s segments, it is more challenging in DCASE 2022 Task 1 challenge as this task requires to evaluate on 1-s segments and does not allow to use pruning techniques.

Given these updated and benchmark sound scene datasets mentioned above, we can see that DC-2020-1A dataset includes all DC-18/19-1A/1B datasets and it is proposed to evaluated the issue of low complexity model in DCASE 2021 Task 1A and DCASE 2022 Task 1 challenges. We, therefore, first use DC-2020-1A dataset for: (1) evaluating our proposed ASC baseline in Section 3, (2 evaluating our proposed novel residual-inception neural network architecture in Section 4, (3 analyzing the trade off between ASC system performance and complexity in Section 5, and (4) evaluating how a pre-trained model for the acoustic event detection (AED) task can help to improve ASC accuracy performance in Section 6. These sections recently mentioned focus on how to achieve high-performance and low-complexity ASC models.

Second, while all DCASE datasets present ten daily scenes, Crowded Scenes dataset was proposed to classify five very noise sound contexts. This inspires us to define a new dataset of sound scene contexts which comprises both DC-2020-1A and Crowded Scenes datatsets. As the new dataset presents diverse scene contexts, a wide range of sound events can be observed. Additionally, it is a fact that specific sound events can only occur in certain sound scene such as the *'gun sound'* or *'explosion'* in a *'riot context'* or the *'loud music'* in a *'music event'*. We, therefore, make use of statistic information of sound events in a sound scene context and ASC systems proposed in previous sections (i.e. From Section 3 to Section 6) to develop a visualization method. The proposed visualization method helps to present a sound scene context more comprehensively in Section 7.

Finally, our proposed models are evaluated on all datasets mentioned and compared with the state-of-the-art systems in Section 8. The Training/Evaluating splitting methods used to evaluate DCASE datasets obey DCASE challenges.

3. Propose an ASC baseline system

To evaluate whether extending deep neural network architectures in depth with trunks of convolutional layers is effective for the ASC task, we first propose an ASC baseline system which presents an inception based architecture and low memory footprint. Next, we construct benchmark neural networks of VGG16, VGG19, ResNet50V2, ResNet152V2, DenseNet169, DenseNet201, and Xception, which present much deeper convolutinal layers compared to the ASC baseline. The proposed low-complexity ASC baseline is evaluated and compared with the benchmark and high-complexity architectures on DC-20-1A dataset. As Fig. 1 shows, the proposed ASC baseline framework is separated into three main steps: Front-end spectrogram feature extraction, online data augmentations, and back-end inception based deep neural network for classification.

3.1. Proposed ASC baseline

Front-end spectrogram feature extraction: The input audio recordings are first resampled to 32,000 Hz. Then, the resampled audio recordings are transformed into Mel spectrograms using Librosa toolbox [54]. As we set the Fast Fourier Transform (FFT) number, Hann window size, the hop size, and the filter number to 4096, 2048, 1024, and 128 respectively, a two-dimensional Mel spectrogram of 128×312 is generated from one 10-s audio recording. We finally apply delta and delta-delta on each

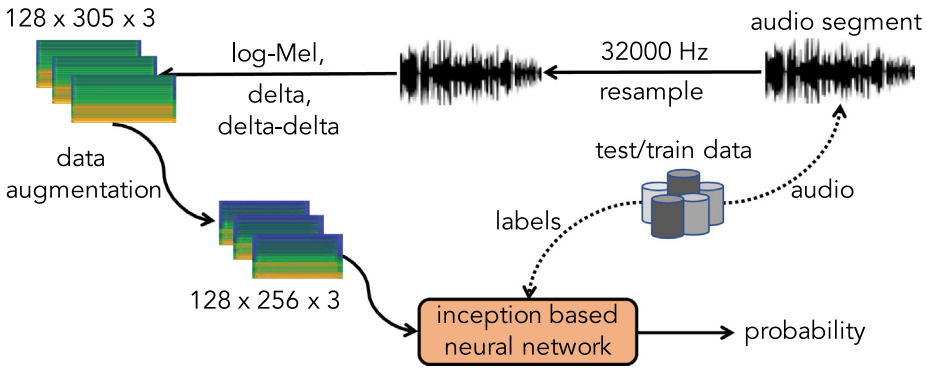

Fig. 1. The high-level architecture of the proposed ASC baseline.

two-dimensional spectrogram, create three-dimensional spectrogram of 128×305×3 (i.e. The channel dimension is three which is created by concatenating the original Mel spectrogram, delta, and delta-delta).

Online data augmentations: In this paper, we apply three data augmentation methods: Random Cropping [55], Specaugment [56], and Mixup [57,58], respectively. First, the temporal dimension of Mel spectrograms of 128×**305**×3 is randomly cropped to 128×**256**×3 (Random Cropping). Next, ten random and continuous temporal and frequency bins of the cropped spectrograms are erased (Specaugment). Finally, the spectrograms are randomly mixed together using different coefficients from both Beta and Uniform distributions (Mixup). As all of three data augmentation methods are applied on each batch of spectrograms in the training process, we refer them to as the online data augmentations.

Back-end inception based deep neural network: As Table 2 shows, the back-end inception based network is separated into two main parts: CNN-based backbone and Multilayer Perception (MLP) based classification. In particular, the CNN-based backbone comprises four Inception Blocks, each of which is performed by an inception layer, followed by batch normalization (BN) [59], Rectified Linear Unit (ReLU) [60], drop out (Dr(drop ratio)) [61], Max Pooling (MP) for the first three Inception Blocks or Global Max Pooling (GMP) for the final Inception Block 04. The inception layer architecture (Inc(*Ch* = The channel number)) is shown in Fig. 2 which is a variant of the naive version of inception layer introduced in [62]. In particular, we use kernel [1×4] instead of [5×5] as usual to enforce the network focus on minor variation across the frequency dimension of audio spectrum. Additionally, we add a convolutional layer with kernel size of [1×1] after the max pooling MP([3×3]) layer.

Regarding the MLP-based classification as shown the lower part in Table 2, it performs two dense blocks (Dense Block 01 and Dense Block 02). While the fully connected layers (FC(*Ch* = The channel number)) in the first Dense Block 01 is followed by Rectified Linear Unit (ReLU) and drop out (Dr(drop ratio), the fully connected layer (FC(*Ch* = *C*)) in the second Dense Block 02 uses Softmax layer (i.e. *C* is set to match the number of categories classified in a target dataset.).

Table 2
The inception based neural network for classification in the ASC baseline system.

Main Blocks	Sub blocks	Layers
CNN-based backbone	Inception Block 01	Inc(*Ch*=128) - BN - ReLU - MP[2×2] - Dr(0.1)
	Inception Block 02	Inc(*Ch*=128) - BN - ReLU - MP[2×2] - Dr(0.15)
	Inception Block 03	Inc(*Ch*=256) - BN - ReLU - MP[2×2] - Dr(0.2)
	Inception Block 04	Inc(*Ch*=256) - BN - ReLU - GMP - Dr (0.25)
MLP-based classification	Dense Block 01	FC(*Ch*=1024) - BN - ReLU - Dr(0.25)
	Dense Block 02	FC(*Ch* = *C*) - Softmax

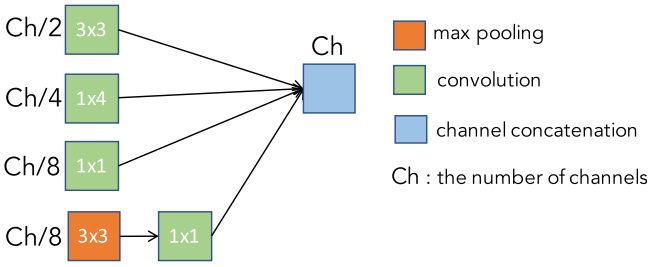

Fig. 2. The architecture of the inception layer (Inc(Ch = The channel number) in the proposed ASC baseline.

3.2. Construct benchmark and high-complexity neural networks for back-end classification

To evaluate whether benchmark and high-complexity deep neural network architectures are effective for the ASC task, we replace the proposed CNN-based backbone in the ASC baseline model by the benchmark architectures of MobileNetV1, MobileNetV2, VGG16, VGG19, ResNet50V2, ResNet152V2, DenseNet169, DenseNet201, and Xception which are available from Keras Application API [63]. In other words, while the proposed MLP-based classification is remained, we evaluate different backbone network architectures. Notably, steps of the front-end spectrogram feature extraction and the online data augmentations used for the ASC baseline are retained during evaluating these benchmark network architectures.

3.3. Dataset and settings for evaluating the proposed ASC baseline and the benchmark neural networks

To evaluate the proposed ASC baseline and the benchmark neural networks, we use DC-20-1A dataset mentioned in Section 2. We obey the challenge and follow the recommended setting as mentioned in Section 2. Regarding the evaluation metric, we use accuracy (Acc.%), which is the most popular and main metric in all ASC challenges [64].

As using the Mixup data augmentation method, labels are not one-hot encoding format. Therefore, we use Kullback–Leibler divergence (KL) loss [65] shown in Eq. (1) below.

$$Loss_{KL}(\Theta) = \sum_{n=1}^{N} \mathbf{y}_n \log\left\{\frac{\mathbf{y}_n}{\hat{\mathbf{y}}_n}\right\} + \frac{\lambda}{2}||\Theta||_2^2 \quad (1)$$

where Θ are trainable parameters, constant λ is empirically set to 0.0001, N is batch size set to 64, $\mathbf{y_i}$ and $\hat{\mathbf{y}}_\mathbf{i}$ denote expected and predicted results, respectively.

Both the proposed ASC baseline and the benchmark neural networks are implemented with Tensorflow framework, using Adam method [66] for optimization. The training and evaluating processes are conducted on GPU Titan RTX 24 GB. The training process is stop after 40 epochs. While the first 30 epochs uses the learning rate of 0.001 and all data augmentation methods mentioned in Section 3, the remaining epochs uses the lower learning rate of 0.00001 with only Random Cropping data augmentation method. The final result, which is reported in this paper, is an average of accuracy results obtained from 10 times of running experiments.

3.4. Performance comparison among DCASE baseline, the proposed ASC baseline, and the benchmark network architectures

As experimental results are shown in Table 3, our proposed ASC baseline system outperforms DCASE baseline and all the benchmark network architectures. Significantly, our proposed ASC baseline helps to improve the DCASE baseline on all seen and unseen recording devices. The results of the proposed ASC baseline also indicate that performances on unseen devices (S4, S5, and S6) are lower than seen devices (A, B, C, S1, S2, and S3) with an average of 10% and the performances on real recording devices (A, B, C) are better than synthesized devices (S1 to S6).

Regarding the model complexity (e.g. The number of trainable parameters with the unit of million (M); The memory footprint presenting the amount of memory with megabyte (MB) unit that a device needs to store the trainable parameters and one trainable parameter is presented by 32 bits using floating point format; The billion of floating point operations (BFLOPs) that a proposed ASC model uses for an inference process over one input sample), we can see that deeper neural networks (MobileNetV1, VGG19, ResNet152V2 or DenseNet201) present low performance than the lower complexity networks (MobileNetV2, VGG16, ResNet50V2, or DenseNet121) from the same architecture groups. Meanwhile, our proposed ASC baseline presents a small trainable parameter number of 0.94 M, a low memory footprint of 3.5 MB, and BFLOPs of 1.52, but achieves the best performance compared to the others. We also see that only DCASE baseline, our ASC baseline, and MobinetV1/V2 network architectures present lower than 5 M trainable parameters while the others show large amount of trainable parameters.

Overall, the experimental results indicate that the shallow inception based network architecture used for the ASC baseline is more effective than deeper architectures for the ASC task with the issue of mismatched recording devices. Although the ASC baseline presents the lowest memory footprint with 3.5 MB which is compatible to a wide range of mobiles or edge devices, this network architecture is still considered too large regarding small and limited-memory devices such as STM32L496@80 MHz or Arduino Nano 33@64 MHz. Additionally, the performance of the proposed ASC baseline (64.6%) is not competitive to the state-of-the-art systems. Therefore, these below sections will show how we improve the ASC baseline accuracy performance, but still satisfy the low-complexity model.

4. A novel residual-inception neural network and multiple spectrograms for ASC

As the performance comparison between the proposed ASC baseline and the benchmark neural networks are shown in Section 3, it indicates that the inception-based architecture shows effective for the ASC task. To further improve the ASC performance, we therefore propose a novel residual-inception (NRI) neural network as shown in Fig. 3.

4.1. Propose a novel residual-inception network architecture

As Fig. 3 shows, the proposed residual-inception network architecture also comprises two main parts: CNN-based deep neural network backbone and multilayer perception (MLP) based classification. In particular, there are four blocks in the proposed CNN-based backbone: one Dob-Inc Block and three Inc-Res Blocks. These four blocks are described at the upper part of Fig. 3. While Dob-Inc Block makes uses of inception-based architecture, both inception-based and residual architectures are leveraged in three Inc-Res Blocks.

Regarding the Dob-Inc Block as shown in the left part of Fig. 3, it reuses the sub-block architecture of Inception Block from the ASC baseline, but using two Inc01(Ch) layers each of which is accompanied with batch normalization (BN). The number of channel (Ch) used at these inception layers is set to 128. Three Inc-Res Blocks as shown in the right part of Fig. 3 present the same network architecture, and the channel numbers are set to 128, 256, and 256, respectively. Each Inc-Res Block presents two data streams. As the first stream is shown on the left, referred to as the shortcut branch, the feature map input goes through (Conv[Ch×1×1]), BN, ReLU, average pooling (AP[3×3]), Residual Normalization (RN ($\lambda = 0.4$)) inspired from [28]. Meanwhile, in the main stream as shown on the right, the feature map input first goes though (Conv[Ch/2×1×1]), BN, and ReLU, then is passed into three sub branches. In each sub branch of the main stream, different kernel sizes, defined by K as shown in the right of Fig. 3, are applied to learn local regions of the feature map input. By using different kernel sizes of [K×1], [K×K], and [1×K] and applying AP layers with the same kernels [K×K], the network is enforced to learn distribution of spectrum in certain frequency bands effectively. This strengthens the network to deal with the different distribution of energy across the frequency dimensions which occurs with mismatched recording devices. Finally, the shortcut stream and three sub branches in the main stream are accumulated before going through ReLU, MP[2×2], Dr(0.1), and RN($\lambda = 0.4$) in the order.

The MLP-based classification as shown in the lower part of Fig. 3 performs a Pooling Block and two fully connected layer blocks. At the Pooling Block, thee types of feature are extracted: (1) global average pooling across the channel dimension (i.e. This is exactly the global average pooling layer (GAP) used in the proposed ASC baseline), (2) global max pooling across temporal dimension, and (3) global average pooling across frequency dimension. We then concatenate the three features before feeding into following fully connected blocks. While the first fully connected layer (FC (Ch = 1024)) is followed by ReLU and Dr(0.2), the second fully connected layer combines with Softmax layer for classifying into C scene categories.

4.2. Further improve ASC performance by an ensemble of multiple spectrogram inputs

As using ensemble is a rule of thumb to improve the ASC performance and shows effective to deal with the issue of mismatched recording devices [14,15,67–69], we therefore apply an ensemble

Table 3
Compare our proposed ASC baseline to DCASE baseline, benchmark network architectures on the DC-20-1A dataset.

	DCASE Baseline	Proposed Baseline	MobileV1	MobileV2	VGG16	VGG19	ResNet50V2	ResNet152V2	DenseNet121	DenseNet201	Xception
A(%)	70.6	73.3	74.2	71.0	68.3	67.1	74.1	74.0	74.1	74.8	75.2
B(%)	60.6	67.0	60.1	56.3	54.5	56.1	57.9	60.8	63.1	58.3	62.0
C(%)	62.6	72.6	63.7	60.6	61.5	61.5	63.7	67.8	63.7	68.6	68.4
S1(%)	55.0	64.2	57.2	52.5	55.3	49.8	60.2	52.5	62.0	57.2	60.1
S2(%)	53.3	64.9	51.4	55.2	54.4	51.4	54.1	52.8	58.9	56.3	54.7
S3(%)	51.7	67.9	55.4	52.5	53.5	52.0	55.6	57.0	60.2	59.6	62.2
unseen-S4(%)	48.2	57.6	43.8	41.3	43.8	38.3	45.6	47.6	51.7	51.5	50.4
unseen-S5(%)	45.2	60.0	44.7	46.1	45.3	44.4	52.0	44.3	53.8	48.7	49.4
unseen-S6(%)	39.6	52.4	32.6	29.5	40.1	31.4	31.7	31.9	40.8	35.7	35.2
Average(%)↑	54.1	**64.6**	53.3	51.6	53.3	50.8	55.1	54.0	58.7	56.7	57.9
Parameters(M)↓	5.0	**0.94**	4.3	3.5	15.3	20.6	25.7	60.5	8.1	20.3	23.0
Memory(MB)↓	19.2	**3.5**	16.4	13.7	58.2	254.8	98.0	230.6	30.9	77.5	87.6
FLOPs (B)↓	13.43	1.52	0.75	**0.40**	20.10	25.49	4.57	14.29	3.73	5.64	5.95

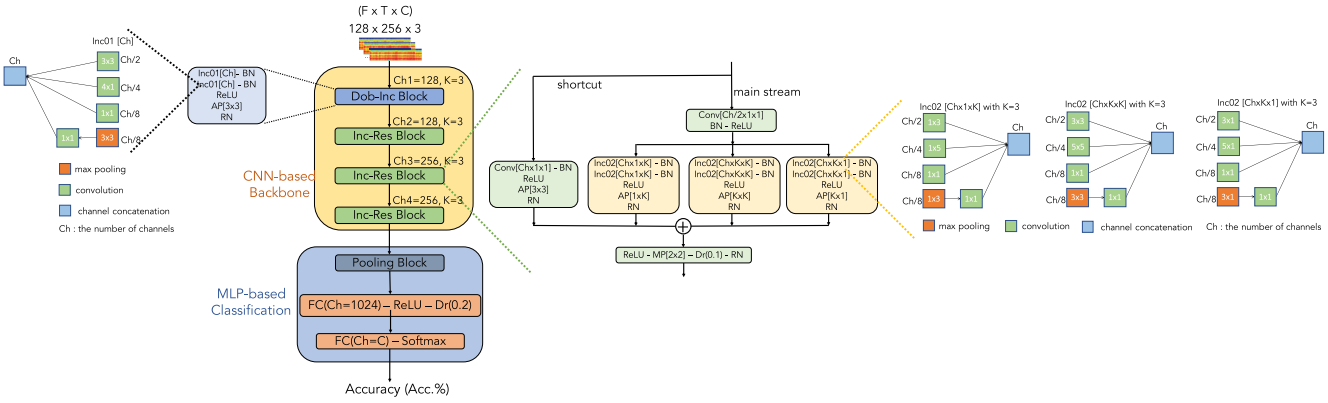

Fig. 3. The proposed novel residual-inception (NRI) deep neural network architecture.

of multiple spectrogram inputs in this paper. In particular, we use three spectrograms: log-Mel [54], GAM [70], and Constant Q Transform (CQT) [54]. To ensure spectrograms present the same size, we reuse the setting parameters of FFT number, Hann window size, the hop size, and the filter number as mentioned in Section 3 and apply for three types of spectrograms. As using multiple spectrogram inputs, each of spectrogram is independently trained with one back-end deep learning model. Then, predicted probabilities obtained from individual models will be fused to achieve the best performance. In this paper, we propose to use late fusion of probabilities, referred to as PROD fusion. Let consider predicted probabilities of each model as $\bar{\mathbf{p}}_s = (\bar{p}_{s1}, \bar{p}_{s2}, \ldots, \bar{p}_{sC})$, where C is the category number and the s^{th} out of S networks evaluated. Next, the predicted probabilities after PROD fusion $\mathbf{p}_{\mathbf{prod}} = (\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_C)$ is obtained by:

$$\bar{p_c} = \frac{1}{S}\prod_{s=1}^{S}\bar{p}_{sc} \; for \; 1 \leqslant c \leqslant C \tag{2}$$

Finally, the predicted label $\hat{y}$ is determined by

$$\hat{y} = argmax(\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_C) \tag{3}$$

4.3. Performance comparison among DCASE baseline, the proposed ASC baseline, the novel residual-inception network architecture with individual spectrograms, and the ensemble of multiple spectrograms

We use DC-20-1A dataset to evaluate the novel residual-inception network architecture and see how an ensemble of multi-spectrogram inputs helps to further improve the ASC performance. All settings and implementation are reused from Section 3.3.

To evaluate the role of individual components in NRI network such as the shortcut branch, the sub-kernel [1×K] and [K×1] branches, the entire Inc-Res Block, and the Pooling Block, we con-Fig. 4 variants of NRI network architectures and evaluate with Mel spectrogram input: (1) the NRI with only using Dob-Inc blocks, referred to as MEL-NRI-Dob (i.e. Three Inc-Res blocks in NRI network are replaced by Dob-Inc block and the channel numbers are remained); (2) the NRI without using sub-kernel [1×K] and [K×1] branches, referred to as MEL-NRI w/o Sub-Ker; (3) the NRI without using the shortcut branch, referred to as MEL-NRI w/o Shortcut; (4) the NRI with only using global average pooling (GAP) at Pooling Block, referred to as MEL-NRI w/ GAP.

As the experimental results show in Table 4, we can see that all 4 variants of NRI architecture outperform the proposed ASC baseline. When Dob-Inc Blocks are replaced by Inc-Res Blocks as comparing between MEL-NRI-Dob and MEL-NRI, it helps to improves by 2.6%. The performances of MEL-NRI w/o Sub-Ker, MEL-NRI w/ o Shortcut, and MEL-NRI w/ GAP with 67.3%, 67.1%, and 68.8%, respectively also proves that it is effective to apply these components for improving the NRI network.

We then evaluate how different spectrogram inputs and ensemble of multiple spectrograms affect NRI network performance. As experimental results are shown in Table 5, we can see that the novel residual-inception (NRI) network trained with Mel spectrogram (MEL-NRI) helps to further improve the proposed ASC baseline in Section 3 by 4.5% and significantly outperform DCASE baseline with an improvement of 15.0%. Compare among spectrograms, the novel residual-inception networks trained with Mel

Table 4
Performance comparison among: The proposed ASC baseline, MEL-NRI with only using Dob-Inc blocks (MEL-NRI-Dob), MEL-NRI without using sub-kernel branches of [1×K] and [K×1] (MEL-NRI w/o Sub-Ker), MEL-NRI without the shortcut branch (MEL-NRI w/o Shortcut), MEL-NRI with only using global average pooling (GAP) at Pooling Block (MEL-NRI w/ GAP), and full MEL-NRI on DC-20-1A dataset.

	Proposed baseline	MEL-NRI-Dob	MEL-NRI w/o Sub-Ker	MEL-NRI w/o Shortcut	MEL-NRI w/ GAP	MEL-NRI
A(%)	73.3	73.6	76.9	73.6	72.1	77.3
B(%)	67.0	72.6	71.5	66.3	71.1	70.5
C(%)	72.6	74.7	74.8	72.0	72.3	75.7
S1(%)	64.2	63.6	68.2	69.1	69.7	69.7
S2(%)	64.9	68.8	67.9	66.4	70.9	70.6
S3(%)	67.9	68.2	70.0	71.2	70.9	71.8
unseen-S4(%)	57.6	60.6	61.5	65.2	66.4	61.5
unseen-S5(%)	60.0	64.2	62.7	64.9	68.2	66.1
unseen-S6(%)	52.4	52.4	53.0	55.2	57.6	58.8
Average(%)↑	64.6	66.5	67.3	67.1	68.8	**69.1**
Parameters (M)↓	**0.94**	1.9	2.8	4.2	4.3	4.3
Memory (MB)↓	**3.5**	7.3	11.0	16.2	16.5	16.6
FLOPs (B)↓	**1.52**	9.51	11.50	14.31	14.50	14.56

spectrogram (MEM-NRI) and GAM (GAM-NRI) are competitive, presenting classification accuracy of 69.1% and 65.8%, respectively. Although the novel residual-inception networks trained on CQT spectrogram (CQT-NRI) presents a low performance of 60.8%, ensemble of three spectrograms (SPECs-NRI) achieves an accuracy of 73.6%, further improve the MEL-NRI by 4.5%.

Regarding performance on different recording devices, SPECs-NRI ensemble model significantly improves the performance on all recording devices compared with DCASE baseline and the proposed ASC baseline. The gap performance between real recording devices (A, B, C) and synthetic devices (from S1 to S6) as well as between unseen devices (S4, S5, S6) and seen devices (A, B, C, S1, S2, S3) are also reduced by using ensemble of three spectrograms (SPECs-NRI). Overall, we have proven that a combination of the novel residual-inception network and the ensemble of multi-spectrogram inputs is effective for ASC system to deal with the issue of mismatched recording devices.

5. The trade off between ASC model complexity and performance

5.1. Propose techniques to reduce the model complexity

As Table 4 and Table 5 show, the complicated architecture of the novel residual-inception (NRI) network and the ensemble of multiple spectrograms lead to increase the number of trainable parameters to 12.9 M, 49.8 MB memory on devices, and 43.68 BFLOPs. As we aim to achieve low-complexity ASC systems which are suitable for various edge devices, two constrains of the maximum memory occupied by proposed models are set: (1) 20 MB for mobiles or large-memory devices basing on surveys in [35,36], and (2) 128 KB for limited-memory devices such as STM32L496@80 MHz or Arduino Nano33@64 MHz. The second constrain of 128 KB memory is also the requirement set to challenges of DCASE 2021 Task 1A and DCASE 2022 Task 1. The models with the memory constrains of 20 MB and 128 KB are referred to as medium-size model (MM) and small-size model (SM) respectively in this paper. Meanwhile, models with unlimited memory occupation is referred to as large-size model (LM).

To achieve these low-complexity models (MM and SM), we apply three techniques of model compression: Channel reduction (CR), channel deconvolution (CD), and quantization (Qu). In particular, we first reduce the channel number in inception layers at each sub block of NRI network to 128, 64, 32 and 16, respectively. To further reduce the model complexity, the channel deconvolution technique is applied on each convolutional layer with a kernel size of [K×K], which is inspired from [71,72]. Let consider *Cin* and *Cout* are the number of input channel and output channel used at a convolutional layer with kernel size of [K×K] as shown in Fig. 4. We then separate the input tensor $\mathbf{X}$ into four sub-tensors of $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}$, and $\mathbf{X_4}$ by splitting the length of channel dimension *Cin* into four same parts (0 to $Cin/4, Cin/4$ to $Cin/2, Cin/2$ to $3Cin/4$, and $3Cin/4$ to Cin) while remaining the other dimensions. Then, convolutional layers with different kernel sizes as shown in Fig. 4 are applied to learn these sub-tensors before concatenating. By using the channel deconvolution (CD), the trainable parameters used for a convolutional layer with kernel [K×K] is reduced to nearly 1/8.5 of the original number of trainable parameters.

By using both channel reduction (CR) and channel deconvolution (CD), we obtain Table 6 which summaries four variants of SPECs-NRI, referred to as SPECs-NRI-RD128, SPECs-NRI-RD64, SPECs-NRI-RD32, and SPECs-NRI-120 KB with 2.62 M, 0.86 M, 0.36 M, and 120 K of trainable parameters, respectively. We can see that these four variants of SPECs-NRI are considered as medium-size model (MM) which meet the requirement of maximum 20 MB of memory occupation on devices. To meet the requirement of maximum 128 KB of memory occupation, we apply the quantization technique to only SPECs-NRI-120 KB (i.e. The quantization technique helps to convert 32-bit floating point to 8-bit integer, which reduce the memory occupation to 1/4 of the original volume), further reduce the occupied memory from 480 KB to 120 KB.

5.2. Performance of the novel residual-inception network architecture with and without using decompression techniques

To evaluate the techniques of model compression applied on the proposed residual-inception neural network, we again conduct experiments on DC-20-1A dataset. All settings and implementation are reused from Section 3.3.

As the experimental results are shown in Table 7, we can see that applying model compression techniques CR and CD on SPECs-NIR leads to decreases the classification accuracy slightly. In particular, SPECs-NRI-RD128, SPECs-NRI-RD64, SPECs-NRI-RD32, SPECs-NRI-120 KB present the accuracy of 72.9%, 72.0%, 71.3%, 71.0% compared with 73.6% of SPECs-NRI model. However, these techniques helps to reduce the model complexity significantly, presenting 9.9 MB and 25.86 BFLOPs, 3.3 MB and 6.72 BFLOPs, 1.4 MB and 1.8 BFLOPs, 480 KB and 0.82 BFLOPs for SPECs-NRI-RD128, SPECs-NRI-RD64, SPECs-NRI-RD32, SPECs-NRI-120 KB, respectively.

Table 5
Performance comparison among: DCASE baseline, our ASC baseline, the novel residual-inception network (NRI) with individual spectrograms of Mel (MEL-NRI), GAM (GAM-NRI), or CQT (CQT-NRI), and ensemble of multiple spectrograms (SPECs-NRI) on DC-20-1A dataset.

	DCASE baseline	Proposed baseline	MEL-NRI	GAM-NRI	CQT-NRI	SPECs-NRI
A(%)	70.6	73.3	77.3	77.3	61.5	80.6
B(%)	60.6	67.0	70.5	67.8	62.3	78.7
C(%)	62.6	72.6	75.7	71.1	57.1	73.9
S1(%)	55.0	64.2	69.7	68.2	62.1	74.6
S2(%)	53.3	64.9	70.6	59.4	63.6	74.2
S3(%)	51.7	67.9	71.8	70.9	61.2	76.4
unseen-S4(%)	48.2	57.6	61.5	61.2	60.9	67.3
unseen-S5(%)	45.2	60.0	66.1	63.3	60.3	71.8
unseen-S6(%)	39.6	52.4	58.8	52.7	58.1	65.2
Average(%)↑	54.1	64.6	69.1	65.8	60.8	**73.6**
Parameters (M)↓	5.0	**0.94**	4.3	4.3	4.3	12.9
Memory (MB)↓	19.2	**3.5**	16.6	16.6	16.6	49.8
FLOPs (B)↓	13.43	**1.53**	14.56	14.56	14.56	43.68

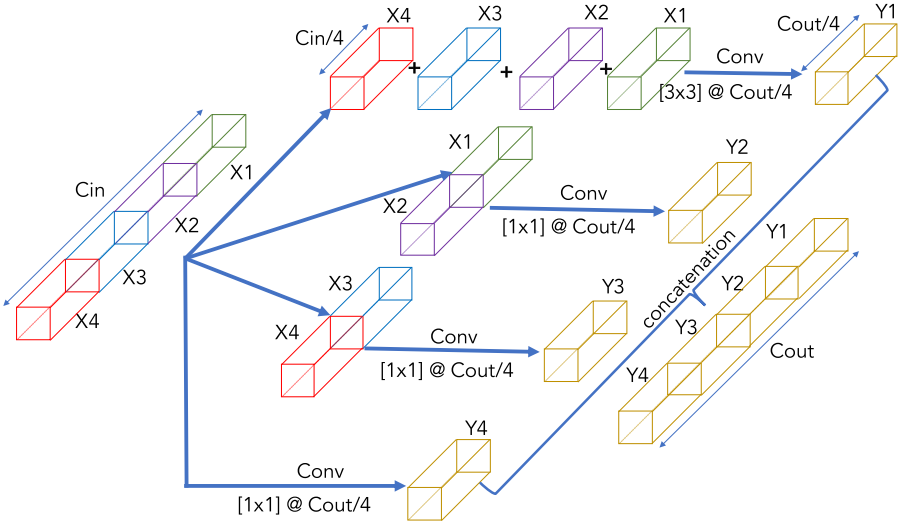

Fig. 4. Channel deconvolution (CD) for reducing trainable parameters.

Table 6
Channel numbers and model complexities after applying channel reduction (CR) and channel deconvolution (CD).

Sub blocks	SPECs-NRI	SPECs-NRI -RD128	SPECs-NRI -RD64	SPECs-NRI -RD32	SPECs-NRI −120 KB
Dob-Inc	*Ch*=128	*Ch*=128	*Ch*=64	*Ch*=32	*Ch*=16
Inc-Res 01	*Ch*=128	*Ch*=128	*Ch*=64	*Ch*=32	*Ch*=32
Inc-Res 02	*Ch*=256	*Ch*=128	*Ch*=64	*Ch*=32	*Ch*=32
Inc-Res 03	*Ch*=256	*Ch*=128	*Ch*=64	*Ch*=32	*Ch*=32
Fully connected 01	*Ch*=1024	*Ch*=1024	*Ch*=1024	*Ch*=1024	-
Fully connected 02	*Ch*=10	*Ch*=10	*Ch*=10	*Ch*=10	*Ch*=10
Parameters (M)	12.9	2.62	0.86	0.36	0.12
Memory (MB)	49.8	9.9	3.3	1.4	0.48
FLOPs (B)	43.68	25.86	6.72	1.8	0.82

Further apply the quantization technique on SPECs-NRI-120 KB, we achieve a very low-complexity model of 120 KB and 0.82 BFLOPs, but still perform an accuracy of 71.0%.

6. Explore acoustic event detection to improve the ASC system

6.1. Adapt an AED pre-trained model to the ASC task

To further improve ASC performance by leveraging sound event information, we first define the task of acoustic event detection (AED) as the up-stream task where sound events in a sound recording are detected. The available model used for AED task is called as the up-stream pre-trained model. We then feed spectrograms of sound scene recordings into the pre-trained model to extract feature maps, referred to as the audio-event-based embeddings. The embeddings are finally classified by a MLP based network into target sound scene classes. In other words, classifying the audio-event-based embeddings using MLP based network is considered as the down-stream ASC task. To the best our knowledge, there are thee papers proposed various up-stream pre-trained models which was trained on the AudioSet, the largest Audio dataset of sound events. The first paper published by Google introduced Trill

Table 7
Performance comparison among SPECs-NRI and four variants of SPECs-NRI-RD128, SPECs-NRI-RD64, SPECs-NRI-RD32, and SPECs-NRI-120 KB w/ quantization on DC-20-1A dataset.

	SPECs-NRI	SPECs-NRI -RD128	SPECs-NRI -RD64	SPECs-NRI -RD32	SPECs-NRI −120 KB
A(%)	80.6	77.9	76.4	75.5	75.7
B(%)	78.7	75.4	74.5	72.3	72.0
C(%)	73.9	78.1	73.9	74.8	77.2
S1(%)	74.6	71.5	73.3	73.0	69.9
S2(%)	74.2	77.3	74.5	70.9	69.9
S3(%)	76.4	76.7	73.6	75.1	74.8
unseen-S4(%)	67.3	67.3	68.5	70.9	69.6
unseen-S5(%)	71.8	69.7	71.8	70.6	71.2
unseen-S6(%)	65.2	62.4	62.1	58.8	61.5
Aver. (%)↑	**73.6**	72.9	72.0	71.3	71.0
Parameters (M)↓	12.9	2.62	0.86	0.36	**0.12**
Memory (MB)↓	49.8	9.9	3.3	1.4	**0.12**
FLOPs (B)↓	43.68	25.86	6.72	1.8	**0.82**

model [73] and Frill model [74] which reused the MobileNetV3 and ResNet50 architectures, respectively. These two pre-trained models present the trainable parameters of 98.1 M and 38.5 M, respectively. The second paper introduced a VGGish network architecture, referred to as openL3 model [75,76], which presents 5.3 M trainable parameters. Meanwhile, the third paper [77] presented a wide range of up-stream pre-trained networks using VGGish, ResNet, MobileNet, DaiNet, LeeNet, Res1dNet, and Wavegram based architectures. As we aim to achieve a low complexity model less than 5 M of trainable parameters or maximum occupying 20 MB memory on devices or mobiles in this paper, we therefore reuse the pre-trained MobinetV2 network from [77] which presents the smallest memory footprint of 4.1 M trainable parameters (occupying 16.0 MB memory on devices). Notably, as all available up-stream pre-trained models recently mentioned are larger than 15 MB, we do not aim to achieve a low complexity model with 128 KB memory occupation in this section.

Given the up-stream pre-trained MobileNetV2 model in [77], we feed Mel spectrograms of sound scene recordings into this model to extract sound-event-based embeddings. The extracted embeddings are the feature maps at the global pooling layer of the up-stream pre-trained MobileNetV2 model. We then use the MLP-based classification as shown in the lower part of Table 2 to classify these sound-event-based embeddings into *C* target sound scene categories. This down-stream ASC task is referred to as DS-ASC-MobV2. The predicted probabilities from the down-stream ASC task (DS-ASC-MobV2) is finally fused with the probabilities obtained from the novel residual-inception based network of SPECs-NRI-RD64 (i.e. The PROD fusion method as mentioned in Section 4.2 is used to fuse the probability results). As the pre-trained MobileNetV2 and SPECs-NRI-RD64 networks present 4.1 M and 0.86 M of trainable parameters respectively, the ensemble of these two models presents 4.96 M of trainable parameter which satisfies our target of low complexity ASC model less than 5 M of trainable parameters or occupying 20 MB memory.

6.2. Performance of ASC models with or without leveraging an AED pre-trained model

To evaluate the role of sound event information to improve ASC performance, we continue using DC-20-1A dataset. All settings and implementation are reused from Section 3.3.

As experimental results are shown in Table 8, the down-stream model of DS-ASC-MobV2 achieves an overall accuracy of 58.9%. The performance is nearly equal to CQT-NRI and significant lower than MEL-NRI and GAM-NIR (CQT-NRI, MEL-NRI, and GAM-NIR performance are shown in Table 5). This indicates that direct training on spectrogram input is better than the approach of using up-stream pre-trained models with the large-scale Audio dataset of sound event.

When we combine DS-ASC-MobV2 with SPECs-NRI-RD64, we can achieve a low complexity model with 4.96 M of trainable parameters (19.4 MB memory occupation). The combination of DS-ASC-MobV2 and SPECs-NRI-RD64, medium-size model (MM), outperforms the large-size model of SPECs-NRI (73.9% compared to 73.6%) and presents a lower model complexity (19.3 MB compared to 49.8 MB and 7.12 BFLOPs compared to 43.78 BFLOPs).

7. Propose a visualization method for well presenting a sound scene context

7.1. Motivation for the developing a visualization method to present a sound scene context

The motivation to develop a visualization method to comprehensively present a sound scene context is driven from two main reasons. First, as the literature review of the ASC task in Section 1 presents, current ASC tasks are specific, defined on certain datasets, and considered as an individual task. Indeed, given an ASC model, we can only acknowledge that how an input audio recording is close to a certain scene context basing on the predicted probabilities. Therefore, when the ASC result is used as the input for other tasks in a complex system, it is hard to give a decision if the predicted probabilities are not significantly different. Additionally, currently proposed ASC models have presented limited performances (i.e. The best models proposed for ASC tasks in DCASE challenges cannot achieve more than 90% classification accuracy) due to various challenges of mismatch recording devices, constrains of low-complexity model, or very similar sound scene contexts (e.g. *'Pedestrian street'* and *'traffic street'* in DCASE challenges or *'firework event'* and *'riot context'* in Crowded Scenes dataset). As a result, applying the ASC task as a main component or as a sub function in real-life applications is limited or shows ineffective if only predicted probabilities of sound scenes are provided from ASC models.

Second, it is fact that sound events and sound scene in a recording present a high correlation. For an instance, *'gun'* sound can be only detected in a *'riot context'* or a group of sound events such as *'wind, grass'*, and *'bird song'* is usually detected *'in a park'*. Therefore, it is potential to use sound event detection (SED) results to enhance an ASC system. Indeed, this has already been proven in the previous Section 6.2 or in some recent published papers [37,38]. However, these works only focus on how to enhance the accuracy performance instead of leveraging detected sound events to present the sound context more comprehensively.

Table 8
Performance comparison among MEL-NIR, SPECs-NRI-RD64, Down-stream ASC task (DS-ASC-MobV2), and ensemble of SPECs-NRI-RD64 and DS-ASC-MobV2 on DC-20-1A dataset.

	SPECs-NRI	SPECs-NRI-RD64	DS-ASC-MobV2	DS-ASC-MobV2, SPECs-NRI-RD64
A(%)	80.6	76.4	65.8	78.8
B(%)	78.7	74.5	58.7	74.5
C(%)	73.9	73.9	67.5	79.6
S1(%)	74.6	73.3	54.9	74.5
S2(%)	74.2	74.5	52.7	76.9
S3(%)	76.4	73.6	57.0	77.0
unseen-S4(%)	67.3	68.5	56.1	68.8
unseen-S5(%)	71.8	71.8	58.2	70.3
unseen-S6(%)	65.2	62.1	59.1	64.8
Aver. (%)↑	73.6	72.0	58.9	**73.9**
Parameters(M)↓	12.9	**0.86**	4.1	4.96
Memory (MB)↓	49.8	**3.3**	16.0	19.3
FLOPs (B)↓	43.68	6.72	**0.40**	7.12

These two reasons recently mentioned inspires us to propose a visualization method (A demo is available[2]), which not only reports predicted probabilities of sound scene contexts but also visually presents a sound scene context more comprehensively by leveraging sound event information.

7.2. Dataset and the use case definition

To evaluate the proposed visualization method, we first define a dataset and propose a case study. Regarding the evaluating dataset, we combine DC-20-1A [42] and Crowded Scenes [39] to form a new dataset of 15 sound scene contexts: *'Airport'*, *'Bus'*, *'Metro'*, *'Metro-Station'*, *'Park'*, *'Public-Square'*, *'Shopping-Mall'*, *'Street-Pedestrian'*, *'Street-Traffic'*, *'Tram'*, *'Music-Event'*, *'Sport-Event'*, *'Firework'*, *'Noise-Street'*, and *'Riot'*. These 15 sound scenes are then grouped into 8 main categories: *'Daily Indoor'* (*'Airport'*, *'Shopping-Mall'*, *'Metro-Station'*), *'Daily Outdoor'* (*'Park'*, *'Public-Square'*, *'Street-Pedestrian'*, *'Street-Traffic'*), *'Daily Transportation'* (*'Bus'*, *'Metro 'Tram'*), *'Music-Event'*, *'Sport-Event'*, *'Firework'*, *'Noise-Street'*, *'Riot'*, which is referred to as 8-sound-scene-context dataset. Given the 8-sound-scene-context dataset, we define a specific task (the case study) which satisfies three requirements of: (1) detect a riot context from the 8-sound-scene-context dataset recently defined (i.e. In the other words, the requirement (1) is a task of sound scene classification on 8-sound-scene-context dataset), (2) low-complexity classification model with less than 5 M trainable parameters (approximately 20 MB memory occupation using 32-bit floating point to present 1 model trainable parameter) which is potential to integrate into a wide range of edge devices and mobiles, and (3) a visualization method for comprehensively presenting statistic information of sound events which show high-relevant to the riot context detected. Experimental settings for evaluating the proposed visualization method are reused from Section 3.3.

7.3. Propose an audio based system for detecting and presenting a riot context

To meet the requirements of (1) and (2), we apply the results from Section 6. In particular, we use two models of SPECs-NRI-RD64 and DS-ASC-MobV2 as shown in Table 8, presenting 4.96 M of trainable parameters (19.4 MB memory occupation), to train and evaluate on the 8-sound-scene-context dataset. While SPECs-NRI-RD64 is only for the ASC task, DS-ASC-MobV2 is used not only for improving the ASC task as experimental results in Section 6.2 but also for detecting sound events occurring in the sound context (i.e. The up-stream task). Given the result of the sound scene classification and the sound events detection, we meet the requirement (3) by generating figures to describe the relationship between sound events and sound scene.

To better describe sound events which present high-relevant to riot contexts regarding the requirement (3), we propose two methods to separate sound events into certain groups. The first method is presented in Table 9 which separates 527 types of sound events defined in AudioSet dataset into three main different alarming levels. In particular, we have three levels, namely Red Level, Yellow Level, and Green Level as shown in Table 9. The Red Level presents very dangerous and rare sound events which are only found in violent-relevant contexts. The sound events with Yellow Level cause a negative or annoying felling which are separated into four sub-categories: sound events made by individual human, sound events from a crowd, natural sound events, and sound events from machines or things. The other sound events are grouped into the Green Level, which is considered as usual events in daily life. In the second method, we separate 527 types of sound events into 7 main categories: Human, Music, Things, Acoustic, Nature, Machine or Vehicle, and Animal. While the first grouping method is driven from our statistics on sound events occurring in riot context, which is conducted on riot recordings in Crowded Scene dataset, the second method is based on the ontology of AudioSet dataset introduced by Google in [78].

Overall, we expect that a sound scene of a riot context in the proposed case study can be presented more comprehensively by exploring both sound scene information (e.g. Predicted probabilities of a sound recording) and sound event information (e.g. Statistics and visualization of sound events in a sound recording).

7.4. Experimental results

Fig. 5 presents a confusion matrix of 8 classes which is the classification result of SPECs-NRI-RD64 and DS-ASC-MobV2 on 8-sound-scene-context dataset. As Fig. 5 shows, the accuracy on each class is larger than 80% and the overall accuracy achieves 90.9%. As the proposed model (SPECs-NRI-RD64 and DS-ASC-MobV2) proves high performance and presents a low memory footprint of 19.4 MB memory occupation on the 8-sound-scene-context dataset, the model is very potential to apply on various mobiles or edge devices.

To present how results of sound scene classification and statistic information of sound events are comprehensively presented via the proposed visualization method, we set up an 80-s recording which presents different sound scene contexts: *'in metro'* from 0 s to 10 s, *'in metro station'* from 10 s to 20 s, *'in traffic street'* from 20 s to 30 s, *'in shopping mall'* from 30 s to 40 s, *'in very noise street'*

[2] https://zenodo.org/record/7366699#.Y4J2HdLMJhG

Table 9
527 sound events and alarming level definition from AudioSet dataset.

Levels	Sound events
Red Level	'Explosion', 'Gunshot, gunfire', 'Machine gun', 'Fusillade', 'Artillery fire', 'Cap gun', 'Eruption', 'Fire', 'Fireworks', 'Firecracker'
Yellow Level (individual person)	'Wail,moan', 'Shout', 'Bellow', 'Whoop', 'Yell', 'Children shouting', 'Screaming', 'Crying, sobbing', 'Baby cry, infant cry'
Yellow Level (a crowd)	'Cheering', 'Crowd', 'Run', 'Applause', 'Hubbub, speech noise, speech babble', 'Battle cry'
Yellow Level (from nature)	'Thunderstorm', 'Thunder'
Yellow Level (thing and machine sounds)	'Basketball bounce', 'Crackle', 'Machanisms', 'Detal drill', 'Buzzer', 'Hammer', 'Jackhammer', 'Power tool', 'Drill', 'Burst, pop', 'Crack', 'Skidding', 'Toot', 'Race car, auto racing', 'Tire squeal', 'Air brake', 'Traffic noise, roadway noise', 'Engine knocking', 'Engine knocking', 'Engine starting', 'Breaking', 'Bouncing', 'Scratch', 'Thump, thud', 'Bang', 'Slam', 'Knock', 'Tap'
Green Level	Other events

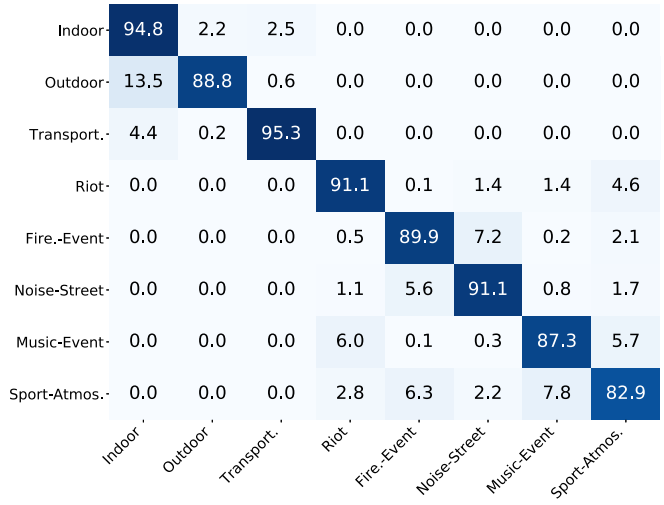

Fig. 5. Confusion matrix results for the ASC task on 8-sound-scene-context dataset.

from 40 s to 50 s, and finally *'in a riot context'* from 50 s to 80 s. Given the audio recording, we fed into the proposed the system (SPECs-NRI-RD64 and DS-ASC-MobV2), present results as shown Fig. 6–9, and Table 10.

We can see that Fig. 6 presents sound scenes detected on each 5-s segment. When the sound context changes at a certain time (e.g. Example: From *'in metro'* to *'in metro station'* at the tenth second), Fig. 6 shows both sound scenes before and after this time point. The riot context is correctly detected and marked with the red color as shown from the fiftieth second to the eightieth second in Fig. 6. As the riot context is detected from the fiftieth second to the eightieth second, we check the groups of alarming sound events recently defined in Table 9. As Fig. 7 and 8 show, we can see that both Red Level alarming sound events and Yellow Level alarming events are detected from the fiftieth second to the eightieth second. While Fig. 7 presents the number of alarming events, Fig. 8 shows the ratios of these alarming sound events on each 5-s segment. We also see that the 'Alarming' sub-group of Yellow Level events mainly appear in the riot context detected from the fiftieth second to the eightieth second. The Fig. 9 indicates that almost sound events related to machines/devices or human occur in a riot context. Finally, Table 10 presents both popular and distinct sound events occurring in each 5-s recording duration together with predicted probabilities and scores.

Our above experimental results have proven that a riot context can be indicated and comprehensively analyzed on a wide range of devices or mobiles by using models of SPECs-NRI-RD64, DS-ASC-MobV2 and the proposed visualization method. By early detect certain riot contexts, it helps to predict a possible large-scale migration or trigger immediately a warning for a certain region (e.g., a violent riot is occurring at the street/district/country X) before a mainstream media (e.g. Television channels, newspaper, etc.) reports. Given the comprehensive analysis of our case study, an application of detecting and presenting a certain sound scene context can be feasibly developed and implemented on a wide range of edge devices and mobiles.

Compare to other visualization methods provided for analyzing ASC systems [79,80], our method presents some advantages. First, we successfully align the sound scene context and the sound event statistic information (i.e. The number of sound events, the percentage of group of sound event, etc.) for each short 5-s duration that helps to indicate a matching or an anomaly between the sound scene and the sound events (i.e. For example, a *'gun'* sound in a shopping mall is considered as an anomaly). Second, not only popular sound events (i.e. *'Crowded sound'*, *'human speech'*, etc. in a *'riot context'*) but also distinct sound events in a sound scene context (i.e. *'Explosion'* sound in a *'riot context'*) are indicated in our visualization method. Finally, as the visualization method explores results obtained from the medium-size model (MM) with less than 20 MB memory footprint occupation, the visualization method is feasibly integrated into a wide range of edged devices and mobiles. These advantages help our proposed visualization method not only present a sound scene context more comprehensively but also potentially apply to a wide range of applications such as audio-based anomaly detection, audio-based observation, etc. on various target devices.

8. Compare to the state-of-the-art ASC systems

Before comparing to the state-of-the-art ASC systems, we present our main proposed systems with the trade-off between accuracy performance and trainable parameters in Fig. 10. As Fig. 10 shows, although NRI based architectures without applying model decompression such as MEL-NRI, GAM-NRI, SPECs-NRI outperform both the proposed baseline and DCASE baseline, they present large models. However, when the model decompression techniques are applied, we can achieve: (1) very low-complexity model (SPECs-NRI-120 KB); (2) a wide range of low-complexity model (SPECs-NRI-120 KB, SPECs-NRI-RD32, SPECs-NRI-RD64) which not only outperform the proposed baseline and DCASE baseline but also present lower trainable parameters; (3) a combined model (DS-ASC-MobV2 + SPECs-NRI-RD64) which can be applied for two tasks of ASC, AED and presents lower than 5 M trainable parameters. Notably, all proposed models, which apply NRI base architecture, multiple spectrogram, and model decompression, achieve performances larger than 70%.

To compare with the state-of-the-art ASC systems, we propose three models in this paper: large-size model (LM) which combines SPECs-NRI and DS-ASC-CNN14 (i.e. DS-ASC-CNN14 is the down-stream ASC task using the up-stream pre-trained CNN14 model in [77]); medium-size model (MM) which combines SPECs-NRI-RD64 and DS-ASCMobv2 presenting 4.96 M of trainable parameter and occupying 19.4 MB memory (i.e. This model was evaluated in the upper sections of 7 and 6); small-size model (SM) which used SPECs-NRI-120 KB with quantization presenting 120 K of trainable parameters, occupying 120 KB memory, and consuming 0.82 BFLOPs (i.e. This model was evaluated in the upper Section 5). These three models are evaluated on a wide range of ASC datasets mentioned in Section 2: DC-18-1A, DC-18-1B, DC-19-1A,

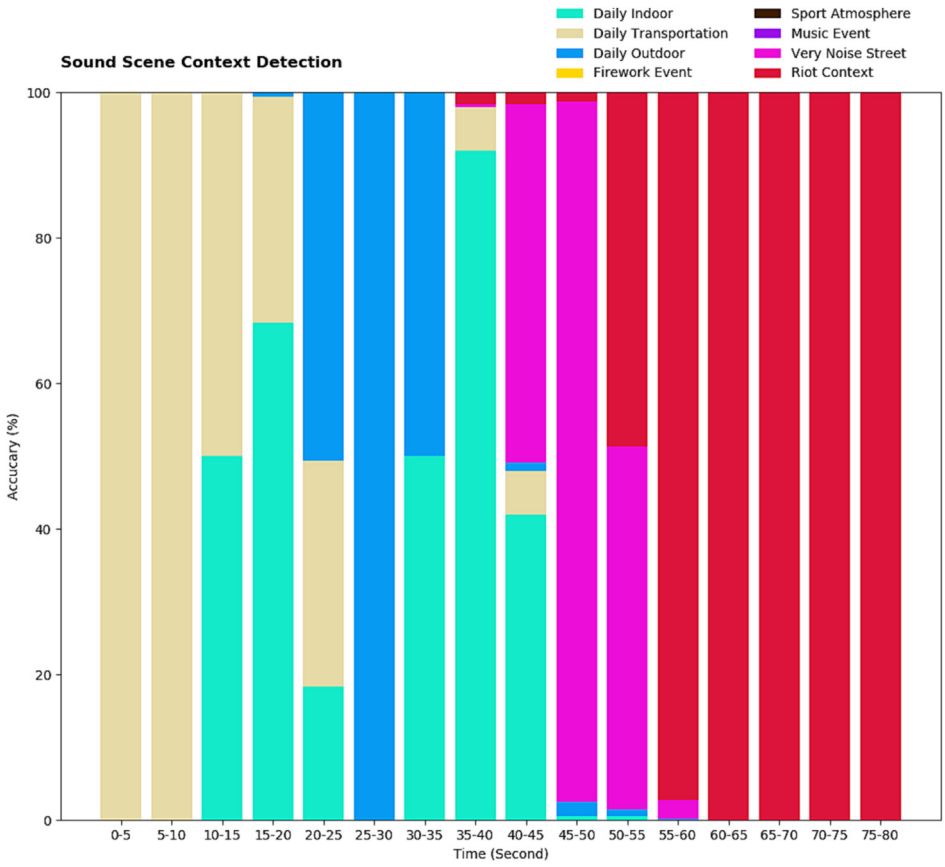

Fig. 6. Visualization method: Presenting the accuracy of detected sound scene contexts and the transferring between two sound contexts.

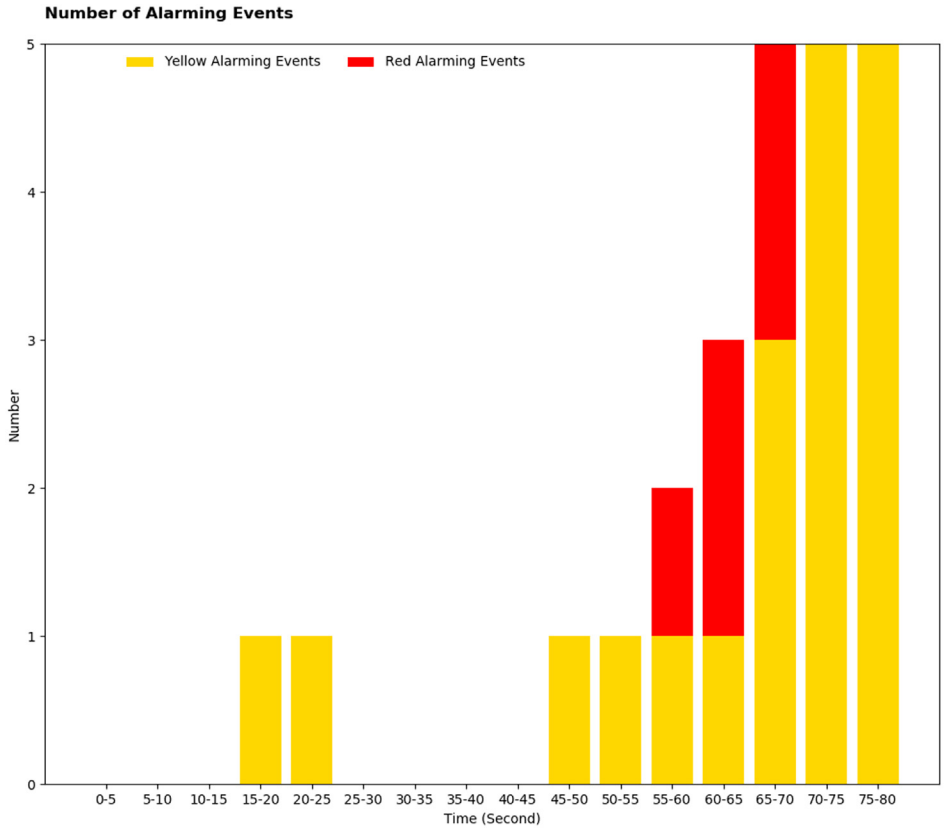

Fig. 7. Visualization method: Presenting Red and Yellow alarming sound events numbers on each 5-s segment.

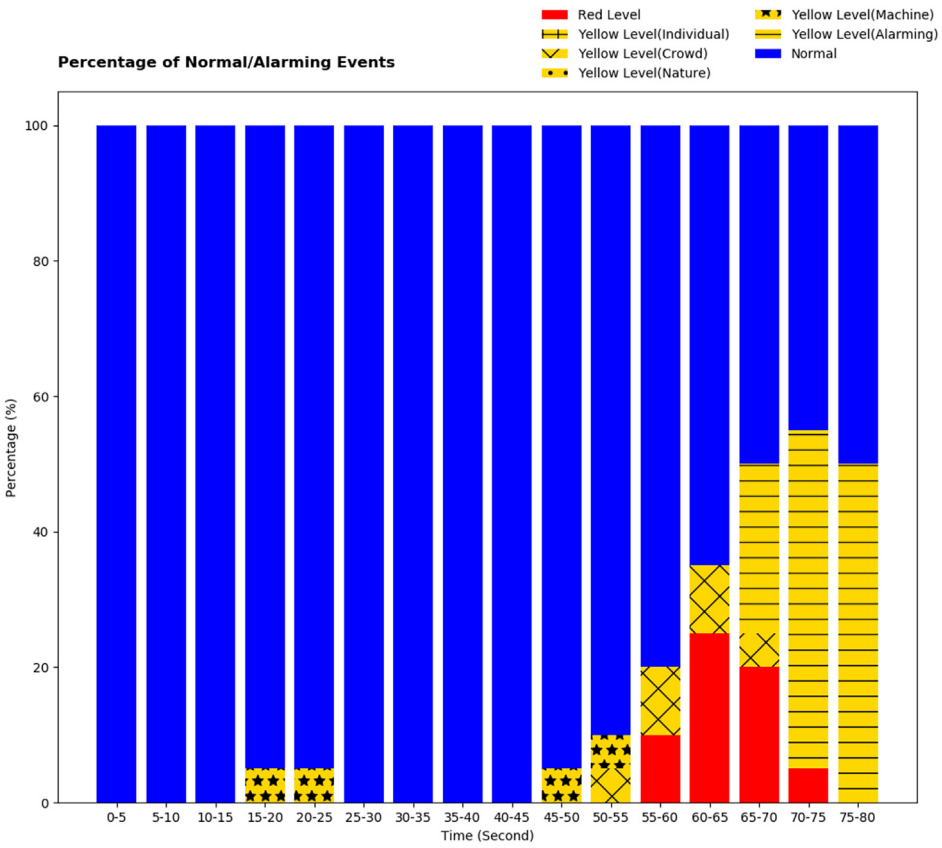

Fig. 8. Visualization method: Presenting percentage of Red, Yellow, and Green sound events numbers on each 5-s segment.

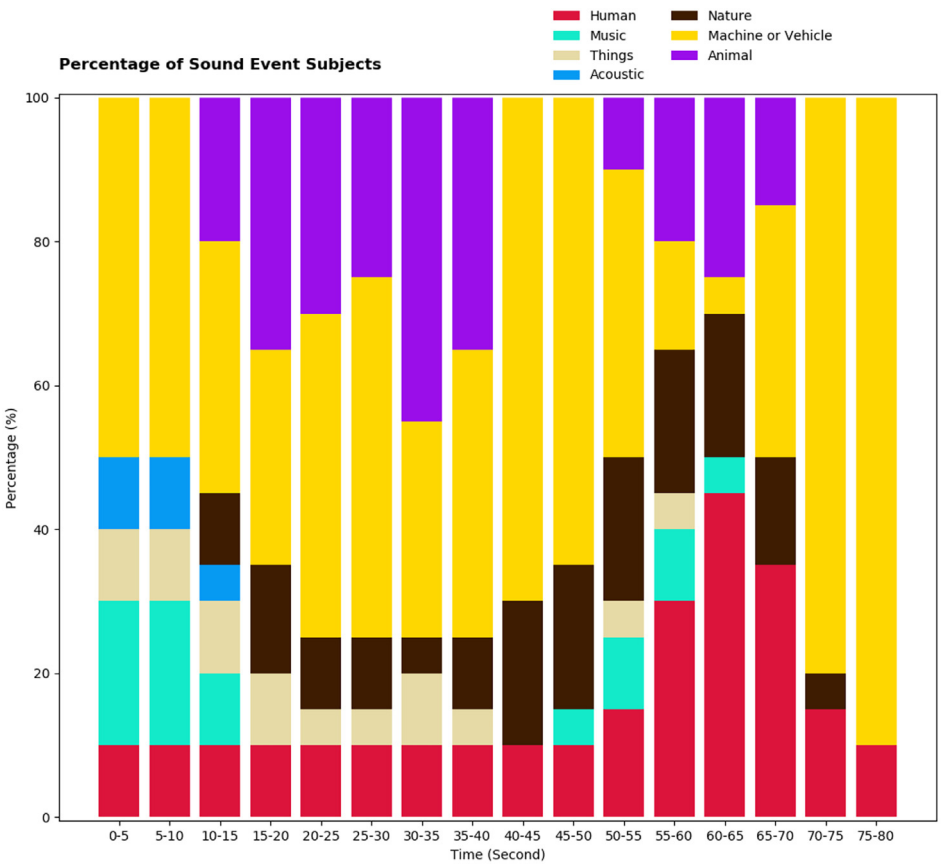

Fig. 9. Visualization method: Presenting topology group of sound events on each 5-s segment.

Table 10
Popular and distinct sound events present from 60 s to 80 s.

Duration	Sound Scenes (Probabilities)	Popular Sound Events (Scores)	Distinct Sound Events(Scores)
...	...	...	...
60s to 65s	riot (99.1%), outdoor (0.4%), firework (0.3%), ..	Speech (0.85), Outside + urban (0.2), ..	Explosion (0.1), Spray (0.03)
65s to 70s	riot (99.5%), outdoor (0.2%), firework (0.2%), ..	Speech (0.8), Outside + urban (0.15), ..	Firework (0.4), Firecracker (0.2)
70s to 75s	riot (68.5%), indoor (31.0%), sport atmosphere (1.5%), ..	Speech (0.75), Outside + urban (0.15), ..	Explosion (0.2), Firecracker (0.2)
75s to 80s	riot (65.5%), indoor (32.0%), sport atmosphere (3.0%), ..	Speech (0.7), Outside + urban (0.2), ..	Slam (0.1)

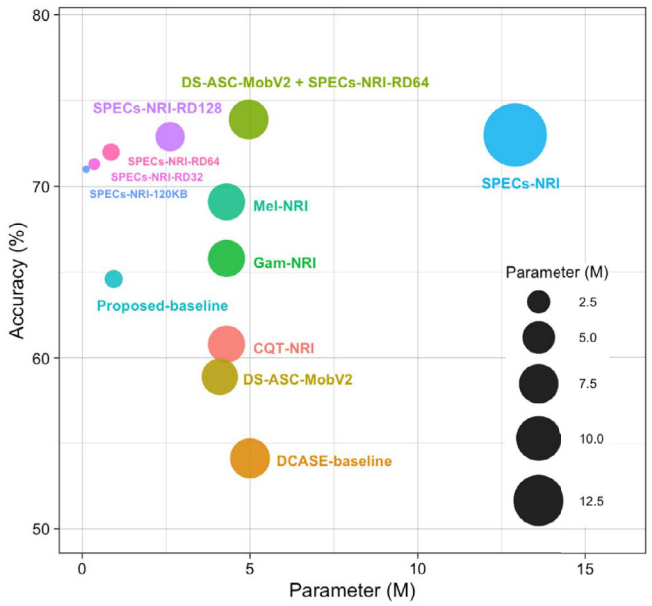

Fig. 10. Model Size (Parameters) vs. Accuracy (%). The trade-off between model size and model performance evaluating on DCASE 2020 Task 1A Development set among DCASE baseline, our proposed baseline, individual spectrograms (CQT, GAM, MEL) with our proposed novel residual-inception (NRI) architecture (CQT-NRI, GAM-NRI, MEL-NRI), downstream ASC task finetuning from the up-stream task of audio detection (DS-ASC-MobV2), and various NRI architectures using ensemble of multiple spectrograms and model decompression (SPECs-NRI-120 KB, SPECs-NRI-RD32, SPECs-NRI-RD64, SPECs-NRI-RD128, SPECs-NRI, DS-ASC-MobV2 + SPECs-NRI-RD64).

DC-19-1B, DC-20-1A. For DCASE 2021 Task 1A and DCASE 2022 Task 1 challenges, we report the results which are from our submitted models presented in [119,120], respectively. Notably, the submitted models also make use of CNN-based network architecture and model compression techniques of channel reduction (CR), channel deconvolution (CD), and Quantization (Qu.). As experimental results are shown in Table 11, we first compare our proposed systems with the state-of-the-art ASC systems without a limitation of model size. We can see that our large-size model (LM) outperforms the state-of-the-art ASC systems in DC-18-1B, DC-19-1B, achieves the top-2 in both DC-18-1A and DC-20-1A, and occupies the top-4 in DC-19-1A. Although the accuracy performance of our proposed medium-size model (MM) slightly reduces as this model is constrained by maximum 5 M of trainable parameter and occupying 20 MB memory to be able to apply on a wide range of edge devices and mobiles, the results are still very competitive to the state-of-the-art systems (top-3 in DC-18-1A, top-1 in DC-18-1B, top-6 in DC-19-1A, top-2 in DC-19-1B, and top-4 in DC-20-1A). Regarding our proposed small-size model (SM) which is constrained by maximum 128 KB memory occupation, this model is still in top-10 compared to the state-of-the-art systems on all evaluating datasets. Specially, this model achieves top-3 and top-4 in DC-19-1B and DC-20-1A. Our submitted systems for DCASE 2021 Task 1A and DCASE 2022 Task 1 challenges (These challenges requires low complexity model with the same constrain of maximum 128 KB memory occupation) also achieve top-6 and top-4 accuracy rankings, respectively. We then compare our small-size model (SM) to the state-of-the-art ASC system with constrains of less than 128 K trainable parameters and without using a pruning technique for model compression. While the first constrain of 128 K trainable parameters (or 128 KB memory occupation using 8-bit quantization) matches the requirements of the recent DCASE 2021 and DCASE 2022 challenges to be compatible for limited memory devices, the second constrain helps to make sure that proposed models can be directly deployed on target devices and matches the DCASE 2022 challenge 's requirement. As Table 12 shows, our proposed SM system outperforms the state of the art on the largest ASC dataset of DC-20-1A. Notably, our SM model as well as other proposed ASC systems in this paper report the number of floating point operations (FLOPs) which is not mentioned in recently published papers.

9. Conclusion and future work

This paper has presented a comprehensive analysis of acoustic scene classification (ASC) and achieved two main outcomes. First, by using multiple techniques: a novel inception-residual based network architecture, an ensemble of multiple spectrogram inputs, ASC down-stream task inherited from the up-stream SED task, and model compression methods, we successfully developed very competitive ASC systems compared to the state of the art on almost challenging ASC datasets. Among our proposed ASC systems, two low-complexity models of medium-size model (19.3 MB memory occupation) and small-size model (128 KB memory occupation) are compatible for real-life applications on a wide range of edge devices and mobiles. This effectively helps to create a benchmark to compare among ASC models. Second, we propose an effective visualization method to present a sound scene context by exploring both sound events and sound scene information.

For the future works, the teacher-student scheme will be investigated. By using the teacher-student scheme, a low-complexity ASC model, which is considered as the student, is potentially improved by leveraging knowledge distilled from the teacher. Additionally, techniques of frequency and time normalization applied to feature maps, which proved effective for the ASC task [28], also need to be deeply analyzed.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 11
Compare our proposed ASC systems (Large model (LM) with SPECs-NRI and DS-ASC-CNN14; Medium Model (MM) with SPECs-NRI-64 and DS-ASC-Mobv2, and Small Model (SM) with SPECs-NRI-120 KB w/ quantization; Submitted models: Models submitted to DCASE 2021 and 2022 challenges) to the state-of-the-art systems on various sound scene datasets.

DC-18-1A (dev. set)	Acc.%	DC-18-1B (dev. set)	Acc.%	DC-19-1A (dev. set)	Acc.%	DC-19-1B (dev. set)	Acc.%	DC-20-1A (dev. set)	Acc.%	DC-21-1A (test. set)	Acc.%	DC-22 (test. set)	Acc.%
Wang [81]	72.4	DCASE baseline	45.6	DCASE baseline	63.3	DCASE baseline	47.7	Jung [38]	70.4	top-1	**76.1**	top-1	**60.8**
Zhao [82]	72.6	Tchorz [83]	53.9	Sun [84]	75.9	Wang [85]	55.2	Shim [86]	71.3	top-2	73.1	top-2	59.7
Zhao [87]	72.7	Shefali [88]	56.2	Wang [81]	75.7	Jiang [89]	64.2	Kim [90]	71.6	top-3	72.1	top-3	56.3
Phaye [17]	74.1	Zhao [91]	63.3	Jung [92]	76.2	Primus [8]	65.1	Zhao [93]	72.2	top-4	71.8	top-4	55.2
Jung [94]	74.8	Truc [14]	63.6	Javier [95]	76.7	McDonnell [96]	66.3	Choi [97]	72.3	top-5	70.1	top-5	54.9
Hossein [9]	76.8	Truc [98]	64.7	Cho [99]	77.2	Zhao [93]	66.5	Liu [100]	73.1	top-6	69.6	top-6	54.7
Heo [101]	77.4	Truc [102]	66.1	Mars [103]	79.3	Song [104]	70.3	Koutini [19]	73.3	top-7	68.8	top-7	53.8
Hou [105]	77.4	Dat [106]	67.5	Choi [97]	81.1	Michal [107]	74.0	Xing [108]	73.3	top-8	68.5	top-8	52.7
Yuanbo [109]	77.4	Yang [110]	67.8	Wang [111]	82.6			Suh [16]	74.2	top-9	68.3	top-9	52.7
Koutini [112]	78.1	Wang [113]	69.0	Huang [114]	83.1			Ma [21]	75.0	top-10	68.1	top-10	51.7
Yuanbo [109]	78.1	Lam [115]	70.6	Liu [116]	83.1			Wang [15]	**81.8**				
Octave [117]	79.3			Koutini [112]	**83.7**								
Yang [118]	**79.8**												
Our LM	79.3	Our LM	**73.3**	Our LM	81.3	Our LM	**75.1**	Our LM	75.4	Our Submitted Model [119]	69.6	Our Submitted Model [120]	55.2
Our MM	77.8	Our MM	**73.0**	Our MM	78.5	Our MM	**70.5**	Our MM	73.9				
Our SM	71.6	Our SM	66.7	Our SM	73.5	Our SM	66.6	Our SM	71.0				

Table 12
Compare our proposed ASC system (SM) to the state-of-the-art ASC systems on DCASE 2021 Task 1A Development dataset with the constrains: less than 128 K parameters and without using pruning techniques

Authors	Systems	Acc. (%)↑	Parameters (K)↓
Hee [121]	ResNetSE-KD	70.5	63.6
	AMFM-KD	69.7	65.4
Liu [122]	FR_agm	68.2	106
	Onebit_agm	68.0	42.5
	Weight_qz	45.4	119
Seo [123]	Fusion	69.0	126.5
Fan [124]	Res-attention	69.7	93.3
Grzegorz [125]	GhostNet	58.8	8.3
	LSTM	60.8	95.1
Xie [25]	TC-SK baseline	58.2	20.9
	TC-SK(AM)	59.9	20.9
Kim [126]	RFN	63.7	8.1
Xing [127]	CNN	70.6	64
Our SM system	**NRI**	**71.0**	**120**

Acknowledgement

The work is also partially supported by the Vietnamese Ministry of Education and Training under the "Research and develop models to predict the safety and lifespan of coastal and island infrastructures using artificial intelligence and condition monitoring" project, number B2021-GHA-03. The work described in this paper is performed in the H2020 project STARLIGHT ("Sustainable Autonomy and Resilience for LEAs using AI against High priority Threats"). This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101021797.

References

[1] Richard F. Lyon, Human and Machine Hearing, Cambridge University Press, 2017.
[2] Brian Clarkson, Nitin Sawhney, and Alex Pentland, "Auditory context awareness via wearable computing," in Proc. of Workshop On Perceptual User Interfaces, 1998, pp. 1–6.
[3] K. El-Maleh, A. Samouelian, and P. Kabal, "Frame level noise classification in mobile environments," in Proc. ICASSP, 1999, pp. 237–240.
[4] Abeber Jakob, Ioannis Mimilakis Stylianos, Grafe Robert, and Lukashevich Hana, "Acoustic scene classification by combining autoencoder-based dimensionality reduction and convolutional neural networks," in Proc. DCASE, 2017, pp. 7–11.
[5] Heittola Toni, Mesaros Annamaria, Eronen Antti, Virtanen Tuomas. Context-dependent sound event detection. Eurasip Journal On Audio, Speech, And Music Processing 2013;1:1–13.
[6] Dcase Community, "DCASE 2018 Task 1B Description," URL:https://dcase.community/challenge2018/task-acoustic-scene-classification#subtask-b.
[7] Dase Community, "DCASE 2020 Task 1A Description," URL:https://dcase.community/challenge2020/task-acoustic-scene-classification-results-a.
[8] Paul Primus, Hamid Eghbal-zadeh, David Eitelsebner, Khaled Koutini, Andreas Arzt, and Gerhard Widmer, "Exploiting parallel audio recordings to enforce device invariance in cnn-based acoustic scene classification," in Proc. DCASE, 2019, pp. 204–208.
[9] Hossein Zeinali, Lukas Burget, and Jan Cernocky, "Convolutional neural networks and X-vector embedding for DCASE2018 acoustic scene classification challenge," in Proc. DCASE, 2018, pp. 202–206.
[10] Phan Huy, Hertel Lars, Maass Marco, Koch Philipp, Mazur Radoslaw, Mertins Alfred. Improved audio scene classification based on label-tree embeddings and convolutional neural networks. IEEE Trans. Audio, Speech and Language 2017;25(6):1278–90.
[11] Dennis Fedorishin, Nishant Sankaran, Deen Dayal Mohan, Justas Birgiolas, Philip Schneider, Srirangaraj Setlur, and Venu Govindaraju, "Waveforms and spectrograms: Enhancing acoustic scene classification using multimodal feature fusion," in Proc. DCASE, 2021, pp. 216–220.
[12] Ren Zhao, Qian Kun, Wang Yebin, Zhang Zixing, Pandit Vedhas, Baird Alice, Schuller Bjorn. Deep scalogram representations for acoustic scene classification. IEEE/CAA Journal of Automatica Sinica 2018;5(3):662–9.
[13] Yuma Sakashita and Masaki Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," Tech. Rep., DCASE Challenge, 2018.
[14] Truc Nguyen and Franz Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in Proc. DCASE, 2018, pp. 34–38.
[15] Helin Wang, Yuexian Zou, and Dading Chong, "Acoustic scene classification with spectrogram processing strategies," in Proc. DCASE, 2020, pp. 210–214.
[16] Sangwon Suh, Sooyoung Park, Youngho Jeong, and Taejin Lee, "Designing acoustic scene classification models with cnn variants," Tech. Rep., DCASE Challenge, 2020.

[17] Sai Phaye, Emmanouil Benetos, and Ye Wang, "SubSpectralNet using sub-spectrogram based convolutional neural networks for acoustic scene classification," in Proc. ICASSP, 2019, pp. 825–829.
[18] Kenneth Ooi, Santi Peksi, and Woon-Seng Gan, "Ensemble of pruned low-complexity models for acoustic scene classification," in Proc. DCASE, 2020, pp. 130–134.
[19] Khaled Koutini, Florian Henkel, Hamid Eghbal-zadeh, and Gerhard Widmer, "Cp-jku submissions to dcase'20: Low-complexity cross-device acoustic scene classification with rf-regularized cnns," Tech. Rep., DCASE Challenge, 2020.
[20] Hu Hu, Chao-Han Huck Yang, Xianjun Xia, Xue Bai, Xin Tang, Yajian Wang, Shutong Niu, Li Chai, Juanjuan Li, Hongning Zhu, et al., "A two-stage approach to device-robust acoustic scene classification," in Proc. ICASSP, 2021, pp. 845–849.
[21] Xinxin Ma, Yunfei Shao, Yong Ma, and Wei-Qiang Zhang, "Three submission for dcase 2020 challenge task1a," Tech. Rep., DCASE Challenge, 2020.
[22] Huy Phan, Oliver Y Chén, Philipp Koch, Lam Pham, Ian McLoughlin, Alfred Mertins, and Maarten De Vos, "Beyond equal-length snippets: How long is sufficient to recognize an audio scene?," in Proc. AES, 2019, p. 16.
[23] Dase Community, "DCASE 2021 Task 1A Description," URL:https://dcase.community/challenge2021/task-acoustic-scene-classification#subtask-a.
[24] Dase Community, "DCASE 2022 Task 1 Description," URL:https://dcase.community/challenge2022/task-low-complexity-acoustic-scene-classification.
[25] Luyuan Xie, Yan Zhong, Lin Yang, Zhaoyu Yan, Zhonghai Wu, and Junjie Wang, "Tc-sknet with gridmask for low-complexity classification of acoustic scene," in Proc. APSIPA ASC, 2022, pp. 1091–1095.
[26] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang, "Selective kernel networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 510–519.
[27] Yifei Xin, Yuexian Zou, Fan Cui, and Yujun Wang, "Low-complexity acoustic scene classification with mismatch-devices using separable convolutions and coordinate attention," Tech. Rep., DCASE Challenge, 2022.
[28] Byeonggeun Kim, Seunghan Yang, Jangho Kim, and Simyung Chang, "Qti submission to dcase 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," arXiv preprint arXiv:2206.13909, 2022.
[29] Nicolas Pajusco, Richard Huang, and Nicolas Farrugia, "Lightweight convolutional neural networks on binaural waveforms for low complexity acoustic scene classification," in Proc. DCASE, 2020, pp. 135–139.
[30] Arshdeep Singh and Mark D. Plumbley, "Low-complexity cnns for acoustic scene classification," in Proc. DCASE, 2022, pp. 191–195.
[31] Google, "Post-training integer quantization," URL:https://www.tensorflow.org/lite/performance/post_training_integer_quant.
[32] Joo-Hyun Lee, Jeong-Hwan Choi, Pil Moo Byun, and Joon-Hyuk Chang, "Hyu submission for the dcase 2022: fine-tuning method using device-aware data-random-drop for device-imbalanced acoustic scene classification," Tech. Rep., DCASE Challenge, 2022.
[33] Florian Schmid, Shahed Masoudian, Khaled Koutini, and Gerhard Widmer, "Cpjku submission to dcase22: Distilling knowledge for lowcomplexity convolutional neural networks from a patchout audio transformer," Tech. Rep., DCASE Challenge, 2022.
[34] Arshdeep Singh and Mark D Plumbley, "Efficient similarity-based passive filter pruning for compressing cnns," arXiv preprint arXiv:2210.17416, 2022.
[35] Zhichuang Sun, Ruimin Sun, Long Lu, and Alan Mislove, "Mind your weight (s): A large-scale study on insufficient machine learning model protection in mobile apps," in 30th USENIX Security Symposium, 2021, pp. 1955–1972.
[36] Taiwo Samuel Ajani, Agbotiname Lucky Imoize, and Aderemi A Atayero, "An overview of machine learning within embedded and mobile devices–optimizations and applications," Sensors, vol. 21, no. 13, pp. 4412, 2021.
[37] Hongwei Song, Jiqing Han, Shiwen Deng, and Zhihao Du, "Acoustic scene classification by implicitly identifying distinct sound events," in Proc. INTERSPEECH, 2019, pp. 3860–3864.
[38] Jee-weon Jung, Hye-jin Shim, Ju-ho Kim, and Ha-Jin Yu, "Dcasenet: An integrated pretrained deep neural network for detecting and classifying acoustic scenes and events," in Proc. ICASSP, 2021, pp. 621–625.
[39] Lam Pham, Dat Ngo, Tho Nguyen, Phu Nguyen, Truong Hoang, and Alexander Schindler, "An audio-visual dataset and deep learning frameworks for crowded scene classification," in Proc. CBMI, 2022, p. 23–28.
[40] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "A multi-device dataset for urban acoustic scene classification," in Proc. DCASE, 2018, pp. 9–13.
[41] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," in Proc. DCASE, 2019, pp. 164–168.
[42] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in Proc. DCASE, 2020, pp. 56–60.
[43] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent, "The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings," The Journal of the Acoustical Society of America, vol. 133, pp. 3591, 05 2013.
[44] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello, "A dataset and taxonomy for urban sound research," in Proceedings of the 22nd ACM International Conference on Multimedia, 2014, p. 1041–1044.
[45] Dan Stowell and Mark D Plumbley, "An open dataset for research on audio field recording archives: freefield1010," arXiv preprint arXiv:1309.5275, 2013.
[46] Karol J. Piczak, "ESC: Dataset for Environmental Sound Classification," in Proceedings of the 23rd Annual ACM Conference on Multimedia, 2015, pp. 1015–1018.
[47] Peter Foster, Siddharth Sigtia, Sacha Krstulovic, Jon Barker, and Mark D. Plumbley, "Chime-home: A dataset for sound source recognition in a domestic environment," in Proc. WASPAA, 2015, pp. 1–5.
[48] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," arXiv preprint arXiv:1609.08675, 2016.
[49] Making Sense of Sounds Data Challenge, "Msos dataset," URL:https://cvssp.org/projects/making_sense_of_sounds/site/challenge/.
[50] F.G. Jort et al., "Audio set: An ontology and human-labeled dataset for audio events," in Proc. ICASSP, 2017, pp. 776–780.
[51] Stowell D, Giannoulis D, Benetos E, Lagrange M, Plumbley MD. Detection and classification of acoustic scenes and events. IEEE Transactions on Multimedia 2015;17(10):1733–46.
[52] Rakotomamonjy Alain, Gasso Gilles. Histogram of gradients of time–frequency representations for audio scene classification. IEEE Trans. Audio, Speech and Language 2015;23(1):142–53.
[53] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "TUT database for acoustic scene classification and sound event detection," in Proc. EUSIPCO, 2016, pp. 1128–1132.
[54] McFee, Brian, Raffel Colin, Liang Dawen, Daniel. PW.Ellis, McVicar Matt, Battenberg Eric, and Nieto Oriol, "librosa: Audio and music signal analysis in python," in Proceedings of 14th Python in Science Conference, 2015, pp. 18–25.
[55] Takahashi Ryo, Matsubara Takashi, Uehara Kuniaki. Data augmentation using random image cropping and patching for deep cnns. IEEE Transactions on Circuits and Systems for Video Technology 2019;30(9):2917–31.
[56] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in Proc. INTERSPEECH, 2019, pp. 2613–2617.
[57] Kele Xu, Dawei Feng, Haibo Mi, Boqing Zhu, Dezhi Wang, Lilun Zhang, Hengxing Cai, and Shuwen Liu, "Mixup-based acoustic scene classification using multi-channel convolutional neural network," in Pacific Rim Conference on Multimedia, 2018, pp. 14–23.
[58] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada, "Learning from between-class examples for deep sound recognition," in ICLR, 2018.
[59] Sergey I. and Christian S., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. ICML, 2015, pp. 448–456.
[60] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," in ICML, 2010.
[61] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 2014;15(1):1929–58.
[62] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
[63] F. Chollet et al., "Keras library," URL:https://keras.io, 2015.
[64] Dcase Community, "DCASE Challenges," URL:https://dcase.community.
[65] Kullback Solomon, Leibler Richard A. On information and sufficiency. The annals of mathematical statistics 1951;22(1):79–86.
[66] P.K. Diederik and B. Jimmy, "Adam: A method for stochastic optimization," CoRR, vol. abs/1412.6980, 2015.
[67] Lam Pham, Ian McLoughlin, Huy Phan, Ramaswamy Palaniappan, and Yue Lang, "Bag-of-features models based on C-DNN network for acoustic scene classification," in Proc. AES, 2019.
[68] Lam Pham, Ian Mcloughlin, Huy Phan, and Ramaswamy Palaniappan, "A robust framework for acoustic scene classification," in Proc. INTERSPEECH, 09 2019, pp. 3634–3638.
[69] Huy Phan, Huy Le Nguyen, Oliver Y. Chén, Lam Pham, Philipp Koch, Ian McLoughlin, and Alfred Mertins, "Multi-view audio and music classification," in Proc. ICASSP, 2021, pp. 611–615.
[70] D.P.W. Ellis, "Gammatone-like spectrogram," URL:http://www.ee.columbia.edu/dpwe/resources/matlab/ gammatonegram.
[71] Khaled Koutini, Florian Henkel, Hamid Eghbal-zadeh, and Gerhard Widmer, "Low-complexity models for acoustic scene classification based on receptive field regularization and frequency damping," in Proc. DCASE, 2020, pp. 86–90.
[72] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," in ICLR, 2016.
[73] J. Shor, J. Aren, M.Ronnie, L.Oran, T.Omry, Q.Felix, T.Marco, I.Shavitt, D. Emanuel, and Y.Haviv, "Towards learning a universal non-semantic representation of speech," in Proc. INTERSPEECH, 2020, pp. 140–144.
[74] Google, "Frill: On-device speech representations using tensorflow-lite," URL: https://ai.googleblog.com/2021/06/frill-on-device-speech-representations.html.
[75] Jason Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in Proc. ICASSP, 2019, pp. 3852–3856.

[76] Relja Arandjelovic and Andrew Zisserman, "Look, listen and learn," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 609–617.
[77] Kong Q, Cao Y, Iqbal T, Wang Y, Wang W, Plumbley MD. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing 2020;28:2880–94.
[78] Google, "Audioset ontology," URL:https://research.google.com/audioset/ontology/index.html.
[79] Zhang Tao, Liang Jinhua, Feng Guoqing. Adaptive time-frequency feature resolution network for acoustic scene classification. Applied Acoustics 2022;195:108819.
[80] Rahil Parikh, Harshavardhan Sundar, Ming Sun, Chao Wang, and Spyros Matsoukas, "Impact of acoustic event tagging on scene classification in a multi-task learning framework," arXiv preprint arXiv:2206.13476, 2022.
[81] You Wang, Chuyao Feng, and David V Anderson, "A multi-channel temporal attention convolutional neural network model for environmental sound classification," in Proc. ICASSP, 2021, pp. 930–934.
[82] Z. Ren, Q. Kong, J. Han, M.D. Plumbley, and B.W. Schuller, "Attention-based atrous convolutional neural networks: Visualisation and understanding perspectives of acoustic scenes," in Proc. ICASSP, 2019, pp. 56–60.
[83] Juergen Tchorz and Mönkhofer Weg, "Combination of amplitude modulation spectrogram features and mfccs for acoustic scene classification," Tech. Rep., DCASE Challenge, 2018.
[84] Jianyuan Sun, Xubo Liu, Xinhao Mei, Jinzheng Zhao, Mark D Plumbley, Volkan Kílíç, and Wenwu Wang, "Deep neural decision forest for acoustic scene classification," in Proc. EUSIPCO, 2022, pp. 772–776.
[85] Zhuhe Wang, Jingkai Ma, and Chunyang Li, "Acoustic scene classification based on cnn system," Tech. Rep., DCASE Challenge, 2019.
[86] Hye-jin Shim, Jee-weon Jung, Ju-ho Kim, and Ha-Jin Yu, "Attentive max feature map and joint training for acoustic scene classification," in Proc. ICASSP, 2022, pp. 1036–1040.
[87] Ren Zhao, Kong Qiuqiang, Qian Kun, D.Plumbley Mark, and W.Schuller1 Bjorn, "Attention-based convolutional neural networks for acoustic scene classification," in Proc. DCASE, 2018, pp. 39–43.
[88] Shefali Waldekar and Goutam Saha, "Wavelet-based audio features for acoustic scene classification," Tech. Rep., DCASE Challenge, 2018.
[89] Shengwang Jiang, Chuang Shi, and Huiyong Li, "Acoustic scene classification using ensembles of convolutional neural networks and spectrogram decompositions," Tech. Rep., DCASE Challenge, 2019.
[90] Gwantae Kim, David K Han, and Hanseok Ko, "Specmix: A mixed sample data augmentation method for training withtime-frequency domain features," arXiv preprint arXiv:2108.03020, 2021.
[91] Zhao Ren, Qiuqiang Kong, Jing Han, Mark D Plumbley, and Björn W Schuller, "Attention-based atrous convolutional neural networks: Visualisation and understanding perspectives of acoustic scenes," in Proc. ICASSP, 2019, pp. 56–60.
[92] Jee-weon Jung, Hee-Soo Heo, Hye-jin Shim, and Ha-Jin Yu, "Distillation the knowledge of specialist deep neural networks in acoustic scene classification," in Proc. DCASE, 2019, pp. 114–118.
[93] Zhao Jingqiao, Kong Qiuqiang, Song Xiaoning, Feng Zhenhua, Xiaojun Wu. Feature alignment for robust acoustic scene classification across devices. IEEE signal processing letters 2022;29:578–82.
[94] Jee-weon, Hee-soo Jung, Hye-jin Heo, Ha-jin Shim, and Yu, "DNN based multi-level feature ensemble for acoustic scene classification," in Proc. DCASE, 2018, pp. 118–122.
[95] Naranjo-Alcazar Javier, Perez-Castanos Sergi, Zuccarello Pedro, Cobos Maximo. Acoustic scene classification with squeeze-excitation residual networks. IEEE Access 2020;8:112287–96.
[96] Mark D McDonnell and Wei Gao, "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," in Proc. ICASSP, 2020, pp. 141–145.
[97] Won-Gook Choi, Joon-Hyuk Chang, Jae-Mo Yang, and Han-Gil Moon, "Instance-level loss based multiple-instance learning for acoustic scene classification," arXiv preprint arXiv:2203.08439, 2022.
[98] Truc Nguyen and Franz Pernkopf, "Acoustic scene classification with mismatched devices using cliquenets and mixup data augmentation," Proc. INTERSPEECH, pp. 2330–2334, 2019.
[99] Janghoon Cho, Sungrack Yun, Hyoungwoo Park, Jungyun Eum, and Kyuwoong Hwang, "Acoustic scene classification based on a large-margin factorized cnn," in Proc. DCASE, 2019, pp. 45–49.
[100] Yue Liu, Xinyuan Zhou, and Yanhua Long, "Acoustic scene classification with various deep classifiers," in Proc. DCASE, 2020, pp. 2–4.
[101] Hee-Soo Heo, Jee-Weon Jung, Hye-Jin Shim, and Ha-Jin Yu, "Acoustic scene classification using teacher-student learning with soft-labels," in Proc. INTERSPEECH, 2019, pp. 614–618.
[102] Truc Nguyen and Franz Pernkop, "Acoustic scene classification with mismatched recording devices using mixture of experts layer," in Proc. ICME, 2019, pp. 1666–1671.
[103] Rohith Mars, Pranay Pratik, Srikanth Nagisetty, and Chongsoon Lim, "Acoustic scene classification from binaural signals using convolutional neural networks," in Proc. DCASE, 2019, pp. 149–153.
[104] Hongwei Song and Hao Yang, "Feature enhancement for robust acoustic scene classification with device mismatch," Tech. Rep., DCASE Challenge, 2019.
[105] Yuanbo Hou, Bo Kang, Wout Van Hauwermeiren, and Dick Botteldooren, "Relation-guided acoustic scene classification aided with event embeddings," in Proc. IJCNN, 2022, pp. 1–8.
[106] Ngo Dat, Pham Lam, Nguyen Anh, Ly Tien, Pham Khoa, Ngo Thanh. Sound context classification based on joint learning model and multi-spectrogram features. International Journal of Computing 2022;21(2):258–70.
[107] Michal Kosmider, "Calibrating neural networks for secondary recording devices," in Proc. DCASE, 2019, pp. 25–26.
[108] Xing Yong Kek, Cheng Siong Chin, and Ye Li, "Multi-timescale wavelet scattering with genetic algorithm feature selection for acoustic scene classification," IEEE Access, vol. 10, pp. 25987–26001, 2022.
[109] Yuanbo Hou, Siyang Song, Chuang Yu, Yuxin Song, Wenwu Wang, and Dick Botteldooren, "Multi-dimensional edge-based audio event relational graph representation learning for acoustic scene classification," arXiv preprint arXiv:2210.15366, 2022.
[110] Yang Liping, Tao Lianjie, Chen Xinxing, Xiaohua Gu. Multi-scale semantic feature fusion and data augmentation for acoustic scene classification. Applied Acoustics 2020;163:107238.
[111] Helin Wang, Yuexian Zou, and Wenwu Wang, "Specaugment++: A hidden space data augmentation method for acoustic scene classification," in Proc. INTERSPEECH, 2021, pp. 551–555.
[112] Koutini Khaled, Eghbal-zadeh Hamid, Widmer Gerhard. Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks. IEEE/ACM Transactions on Audio, Speech, and Language Processing 2021;29:1987–2000.
[113] Wang Jun and Li Shengchen, "Self-attention mechanism based system for dcase 2018 challenge task 1 and task 4," Tech. Rep., DCASE Challenge, 2018.
[114] Jonathan Huang, Paulo Lopez Meyer, Hong Lu, Hector Cordourier Maruri, and Juan Del Hoyo, "Acoustic scene classification using deep learning-based ensemble averaging," in Proc. DCASE, 2019, pp. 94–98.
[115] Pham Lam, Phan Huy, Nguyen Truc, Palaniappan Ramaswamy, Mertins Alfred, McLoughlin Ian. Robust acoustic scene classification using a multi-spectrogram encoder-decoder framework. Digital Signal Processing 2021;110:102943.
[116] Yang Liu, Alexandras Neophytou, Sunando Sengupta, and Eric Sommerlade, "Cross-modal spectrum transformation network for acoustic scene classification," in Proc. ICASSP, 2021, pp. 830–834.
[117] Octave Mariotti, Matthieu Cord, and Olivier Schwander, "Exploring deep vision models for acoustic scene classification," in Proc. DCASE, 2018, pp. 103–107.
[118] Liping Yang, Xinxing Chen, and Lianjie Tao, "Acoustic scene classification using multi-scale features," in Proc. DCASE, 2018, pp. 29–33.
[119] Lam Pham, Alexander Schindler, Anahid Jalali, Hieu Tang, Hoang Truong, "DCASE 2021 Task 1A: Technique Report," URL:https://dcase.community/documents/challenge2021/technical_reports/DCASE2021_Pham_5_t1.pdf.
[120] Lam Pham, Hieu Tang, Anahid Jalali, Alexander Schindler, Ross King, and Ian McLoughlin, "A low-complexity deep learning framework for acoustic scene classification," Tech. Rep., DCASE Challenge, 2022.
[121] Hee-Soo Heo, Jee-weon Jung, Hye-jin Shim, and Bong-Jin Lee, "Clova submission for the dcase 2021 challenge: Acoustic scene classification using light architectures and device augmentation," Tech. Rep., DCASE Challenge, 2021.
[122] Yingzi Liu, LiangLuojun Zhao Jiangnan, Jia Liu, Weiyu Liu, Kexin Zhao, Long Zhang, Tanyue Xu, and Chuang Shi, "Dcase 2021 task 1a: Low-complexity acoustic scene classification," Tech. Rep., DCASE Challenge, 2021.
[123] Soonshin Seo and J Kim, "Mobilenet using coordinate attention and fusions for low-complexity acoustic scene classification with multiple devices," Tech. Rep., DCASE Challenge, 2021.
[124] Mengfan Cui, Fan Kui, and Liyong Guo, "Consistency learning based acoustic scene classification with res-attention," Tech. Rep., DCASE Challenge, 2021.
[125] Grzegorz Stefański, Krzysztof Arendt, Paweł Daniluk, Bartłomiej Jasik, and Artur Szumaczuk, "Short-term memory convolutions," arXiv preprint arXiv:2302.04331, 2023.
[126] Byeonggeun Kim, Seunghan Yang, Jangho Kim, Hyunsin Park, Juntae Lee, and Simyung Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," arXiv preprint arXiv:2206.12513, 2022.
[127] Xing Yong Kek, Cheng Siong Chin, and Ye Li, "An intelligent low-complexity computing interleaving wavelet scattering based mobile shuffling network for acoustic scene classification," IEEE Access, vol. 10, pp. 82185–82201, 2022.