Shao *et al.*
EURASIP Journal on Audio, Speech, and Music Processing (2024) 2024:1
https://doi.org/10.1186/s13636-023-00323-5

EURASIP Journal on Audio, Speech, and Music Processing

METHODOLOGY

Open Access

Deep semantic learning for acoustic scene classification

Yun-Fei Shao[1,3], Xin-Xin Ma[2,4], Yong Ma[2*] and Wei-Qiang Zhang[1*]

Abstract

Acoustic scene classification (ASC) is the process of identifying the acoustic environment or scene from which an audio signal is recorded. In this work, we propose an encoder-decoder-based approach to ASC, which is borrowed from the SegNet in image semantic segmentation tasks. We also propose a novel feature normalization method named Mixup Normalization, which combines channel-wise instance normalization and the Mixup method to learn useful information for scene and discard specific information related to different devices. In addition, we propose an event extraction block, which can extract the accurate semantic segmentation region from the segmentation network, to imitate the effect of image segmentation on audio features. With four data augmentation techniques, our best single system achieved an average accuracy of 71.26% on different devices in the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 ASC Task 1A dataset. The result indicates a minimum margin of 17% against the DCASE 2020 challenge Task 1A baseline system. It has lower complexity and higher performance compared with other state-of-the-art CNN models, without using any supplementary data other than the official challenge dataset.

Keywords Acoustic scene classification, Audio semantic, Mini-SegNet, Mixup Normalization, DCASE 2020

1 Introduction

"Acoustic scene" is the concept that humans commonly used to identify a particular acoustic environment. The task of sensing and understanding the environment where a sound is detected is known as Acoustic Scene Classification [1]. It aims to categorize the detected sound into one of the predefined classes such as a park, airport, or bus. In recent years, methods using CNNs have been widely studied, where the spectrum of the acoustic scene is used as image input, such that best practice image classification methods can be applied [2–4]. However, there are still many issues to address.

Firstly, the accuracy of similar audio scenes is low [5], such as airports and shopping malls. Both are indoor places with many people and contain many similar sounds, such as conversation, broadcasting, and personnel movement. The only difference is the airport contains the roar of an aircraft engine. However, note that engine sound can also be much weakened when it comes to the airport interior; it is hard to recognize when the sound is weak all the time. Therefore, if the deep learning approach cannot learn the different features of similar scenes, it cannot recognize them correctly, because the proportion of similar parts of the scene is high.

Secondly, the generalization performance on unknown devices is poor [5]. Due to the different filtering properties of microphones in recording equipment, the recording quality of different equipment will be uneven. The network structure will learn the characteristics of the

*Correspondence:
Yong Ma
may@jsnu.edu.cn
Wei-Qiang Zhang
wqzhang@tsinghua.edu.cn
[1] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2] School of Linguistic Sciences and Arts, Jiangsu Normal University, Xuzhou 221009, China
[3] School of Mechanical Science and Engineering, Northeast Petroleum University, Daqing 163318, China
[4] TAL Education, Beijing 100085, China

equipment when there are few recording devices. This will make the model parameters overfit to the known equipment. However, there are many kinds of recording equipment in practical applications. If the impact of the equipment cannot be eliminated, it will not be widely used.

Finally, the current network structure is very complex, like the parameters of the top three networks in DCASE 2020 challenge Task 1A, the minimum number of parameters is 39 MB, and the maximum is 130 MB [5]. Although the use of larger models can achieve higher accuracy, the hardware requirements will also be higher, so that it cannot be used on lightweight hardware.

In order to solve the above problems, we aim to develop a CNN system with low complexity to improve recognition performance on unseen devices. We propose a concept of semantic segmentation for acoustic scene classification with multiple devices. Drawing on the experience of SegNet [6] networks of image semantic segmentation, we proposed Audio-SegNet networks of audio semantic segmentation, which is an extension of our previous work [7]. In order to reduce the number of parameters and simplify the Audio-SegNet as much as possible, we have deleted some layers in the original SegNet network. Compared with the original model using 26 layers of conventional convolution, our proposed network only has 6 layers of conventional convolutions. Moreover, we also change the convolution kernel size from 3×3 to 2×3 to further reduce the number of parameters, which is an extension of our previously proposed Mini-SegNet architecture [8].

We then propose a novel feature normalization method which we termed Mixup Normalization. It can learn useful information from scene and discard unnecessary device-specific information. This normalization layer is added to the first convolution layer and the last convolution layer. Compared with the BN [9], our normalization layer can greatly improve the convergence speed and ensure the independence between features [10].

In addition, we also propose a new module which we termed as event extraction block. This module is added to the last layer of the decoder to get the semantic segmentation area to improve the prediction of similar audio scenes.

Our main contributions are summarized as follows:

1) Proposed an audio semantic segmentation system with as low complexity as possible without using model compression method.
2) Proposed a new event extraction block module to improve the recognition performance of similar audio scenes.
3) Proposed a novel normalization method, termed Mixup Normalization.

The rest of the paper is organized as follows. In Section 2, we introduce the development history of ASC and describe some acoustic scene classification methods and existing problems and the main idea of the proposed system. In Section 3, we present the proposed ASC systems, including encoder-decoder architectures, event extraction block, data augmentation, and Mixup Normalization. In Section 4, we show the database description and experiments setup. Experimental results obtained with our system are explained and analyzed. Finally, a summary and conclusion are presented in Section 5.

2 Previous works

The first Detection and Classification of Acoustic Scene and Events 2013 challenge [11] was organized by the IEEE Audio and Acoustic Signal Processing (AASP) Technical Committee. It released open and established datasets and provided the scenario to evaluate and benchmark different approaches for the acoustic scene.

In the ASC task in DCASE 2013, 2016, and 2017, the audio data of acoustic scene classification comes from a kind of high-quality acquisition equipment. In order to study acoustic scene classification more widely, DCASE 2018 [12] and 2019 [13] proposed the mismatch task in different recording devices A, B, C, and D. In the DCASE2020 [14], ASC challenge Task 1A was Acoustic Scene Classification with Multiple Devices. This task includes 10 classes of sounds recorded on multiple devices. The dataset contains a fair number of examples from a high-quality device (referred to as A), as well as a limited number from the targeted low-quality devices (referred to as B and C) and simulated devices (referred to as S1–S6). A gap in the amount and quality of the recorded data causes overfitting on classification results. In particular, a part of the evaluation set is a compressed version of recorded audio data from device D and simulated devices S7–S11. This not only brings ASC closer to real-world conditions but also presents a huge challenge.

In the early period, researchers studied acoustic characteristics such as Zero Crossing Rate [15], Perceptual Linear Prediction [16], and Mel Frequency Coefficients [17] for the classification of the acoustic scenes. In recent years, mainly selected features are Constant-Q transform [18] and log-mel spectrogram [19]. However, there seems to be no general consensus on which features are best. Recently, Helin et al. [20] proposed several spectrogram processing fusion strategies to obtain more discriminative information for ASC, including log-Mel spectrogram, CQT, Gamma, and MFCC.

After that, more and more CNN-based classifiers are designed [2–5, 21, 22]. In [23], the authors presented a localized (small) kernel CNN layer. Sequence correction and local spectral time information are used for parallel networks of CNN and LSTM [24]. Phaye et al. [25] developed a sub-spectrogram based on CNN architecture. McDonnell and Gao [26] proposed a two-path residual network, explicitly dividing the high and low frequencies of the spectrum into two parallel pathways within the same network.

In real life, the environmental sound is mostly collected by different recording devices. Therefore, many data enhancement methods are used to reduce the impact caused by device characteristics between different devices, such as SpecAugment++ [27], GAN [28, 29], Mixup, Temporal crop, spectrum correction, pitch shift, speed change, adding random noise, and mixing audios [30]. Meanwhile, in order to get better recognition results, the network structure design is more complex. For example, for ResNet [31, 32] and FCNN [30], although the recognition accuracy is currently state-of-the-art, they may have several drawbacks. It should be noted that their network structure is highly complex and uses more data enhancement methods. Larger models may require better hardware for training and fine-tuning, such as working on the Graphics Processing Unit. In addition, the hardware resources are limited in many real-world applications, such as smart wearable devices, Bluetooth earphones, and smart phones. Therefore, large models may also face deployment issues on a computationally limited platform [33]. So it is hard to use complex networks widely. In the DCASE 2021 Task 1A [22], researchers need to solve not only the generalization problem that some devices only appear in the evaluation dataset but also the model complexity limit of 128 KB is set for the non-zero parameters. Therefore, some methods [34–36] to reduce the complexity of the model are used, such as pruning, quantization, and knowledge distillation. At the same time, Yang et al. [37] propose a novel neural model compression strategy, called Acoustic Lottery. Specifically, they use the Lottery Ticket Hypothesis [38] method to find a sub-network neural model associated with a small amount of non-zero model parameters in an advanced neural network. However, this method only reduces the number of non-zero parameters, and the total number of parameters does not decrease if we do not adopt the sparse representation.

Although the previous methods have greatly improved performance, there are still many basic problems worth exploring, such as confusion between similar scenes in terms of time and the difficulty of developing high-performance systems due to the presence of overlapping sound events, as well as the lack of distinguishing commonalities between different scene categories. Especially in the classification of acoustic scenes under different devices, there is still a problem of inconsistent audio quality. To address these issues, semantic segmentation networks have had good classification effect in image recognition and can effectively distinguish acoustic segments in different scenes [39]. Examples of such networks include Fully Convolutional Networks (FCN) [40], SegNet [6], U-net [41], and DeepLab [42–44]. For audio classification tasks, encoder-decoder network-based methods have been successfully applied for music source separation [45, 46]. For instance, Liu et al. [47] used the U-net network with a self-attention method to separate voice and accompaniment in music. In their self-attention subnets, the same musical patterns can be reconstructed to achieve better source separation performance. Moreover, Huang et al. [48] proposed an RNN-based Encoder-Decoder framework for pitch tracking. Then, the encoder part, as the pitch extractor, can be applied to a down-stream Mandarin tone classification task. Based on the aforementioned points, we believe that acoustic scenes are composed of some basic units (acoustic events) which contain certain semantic information. Therefore, we proposed audio semantic segmentation with event extraction block and Mixup Normalization for acoustic scene classification.

3 Network architecture

This section introduces an efficient model design for acoustic scene classification with multiple devices. It also describes the details involved in the processing flow and model architecture.

The diagram of the ASC classifiers used in our proposed SegNet approach is illustrated in Fig. 1. The motivation of our method is to extract fine-grained features from acoustic events by convolutional encoder-decoder. Our system consists of two important stages. Firstly, mono audio signals are converted to time-frequency representations, with zero mean and unit variance normalization. Secondly, the log-mel feature is fed to Mini-SegNet models for feature learning. The output layer includes a dense layer of K classes and a softmax function for classification.

3.1 Proposed Mini-SegNet system

In the realm of ASC, CNNs have become the preferred method [34, 49] for classifying log-Mel spectrograms [5, 22]. Specifically, a 2D time-frequency representation is initially extracted from a given audio clip. Subsequently, the neural network can perform feature extraction and dimensionality reduction through operations such as convolution and pooling [50], resulting in a deep representation.

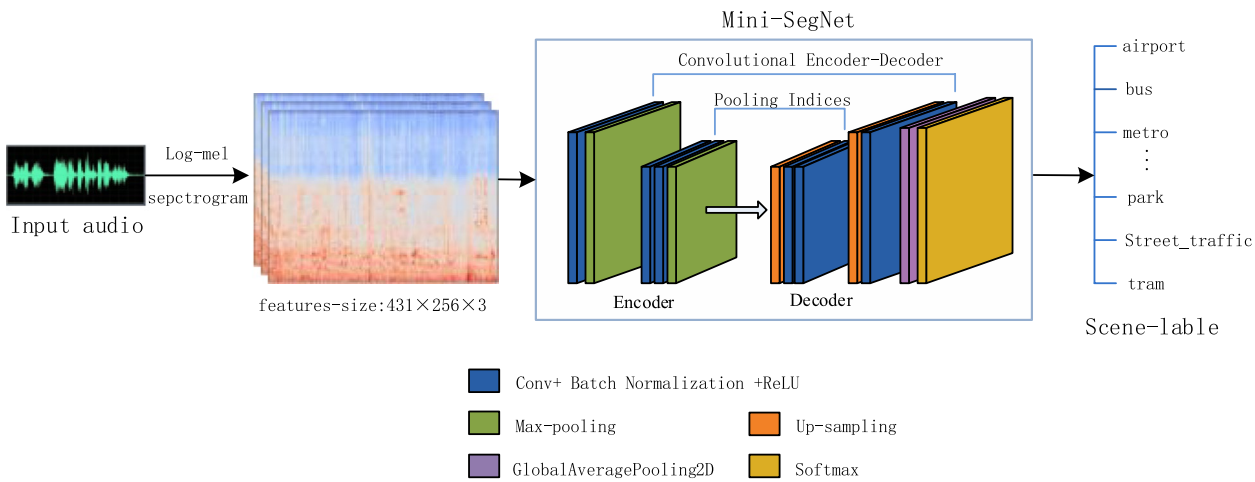


Fig. 1 Block diagram of the Mini-SegNet for the acoustic scene classification

We think that the acoustic scene is composed of some basic units (acoustic events), just as language governs the syntax of phonemes and words. As we all know, bird chirping is recorded in the park, and the sound of aircraft engines is recorded in the airport. Bird chirping and aircraft engines are what we call acoustic events. These acoustic events contain some semantic information, which has a certain internal relationship with the discrimination of acoustic scenes. Therefore, we proposed audio semantic segmentation with event extraction block for acoustic scene classification. In the field of image segmentation, the SegNet network has achieved encouraging results [6]. This is primarily because maxpooling and subsampling reduce feature map resolution, using multi-scale feature mapping to improve segmentation performance. CNN-based models have been widely utilized to encode complicated scene utterances into high-level semantic representations [50]. SegNet arises from this need to map low resolution features to input resolution for pixel-wise classification [6]. Inspired by SegNet network, the encoder-decoder network Audio-SegNet and the event extraction block are designed to capture the temporal and spatial information of an audio feature for acoustic scene classification.

The main idea of this paper is to use an encoder-decoder architecture to learn the acoustic scene for precise semantics mapping. Therefore, we verify the idea of Audio-SegNet, using pooling indices to inform the upsampling layers and extracting acoustic features from the pooling layers in the encoding process. This makes it easier for the decoder to get precise semantic segmentation in frequency. This paper conducts a more in-depth study based on our previous work on Mini-SegNet [8].

In our work, we proposed the Audio-SegNet to extract the multi-granularity abstract features, as shown in Fig. 1. In the encoder module, convolution and pooling are used to extract features and reduce dimensions. In the decoding process, the position and frequency band information are recovered by convolution of the corresponding encoding module sampled on Upsampling2D to make up for the missing pixel information. This method makes full use of the semantic information of sound events in the acoustic scene through the encoding and decoding process and uses the rules of "acoustic scene based on sound events" to provide a preliminary basis for future work.

The details of Mini-SegNet are shown in Fig. 2. In this network, we use a simpler and smaller convolution/upconvolution and maxpooling/upsampling. In image segmentation, better performance can be achieved by only using the information from the last feature map [6]. But in our case, its performance is not satisfactory. In the ASC, it is an overall classification task and does not need predicting labels for each spatial output like Image-SegNet [6]. But, we can get more refined semantic segmentation through upsampling and then get an accurate proportion of events through the event extraction block as shown in Fig. 2. Meanwhile, we add a global average pooling layer or an event extraction block after the decoder of the network. The high dimensional feature representation at the output of the final decoder is fed to a trainable softmax classifier. In Image-SegNet, this softmax classifies each pixel independently [6]. In order to realize ASC, we modify it in Audio-SegNet. The output of softmax classifiers is a K number of acoustic scene classes.

The design of Audio-SegNet represents the first technical contribution to this task. In order to analyze the performance of Audio-SegNet, we constructed several Audio-SegNet networks with different convolution network depth and convolution kernel size to classify

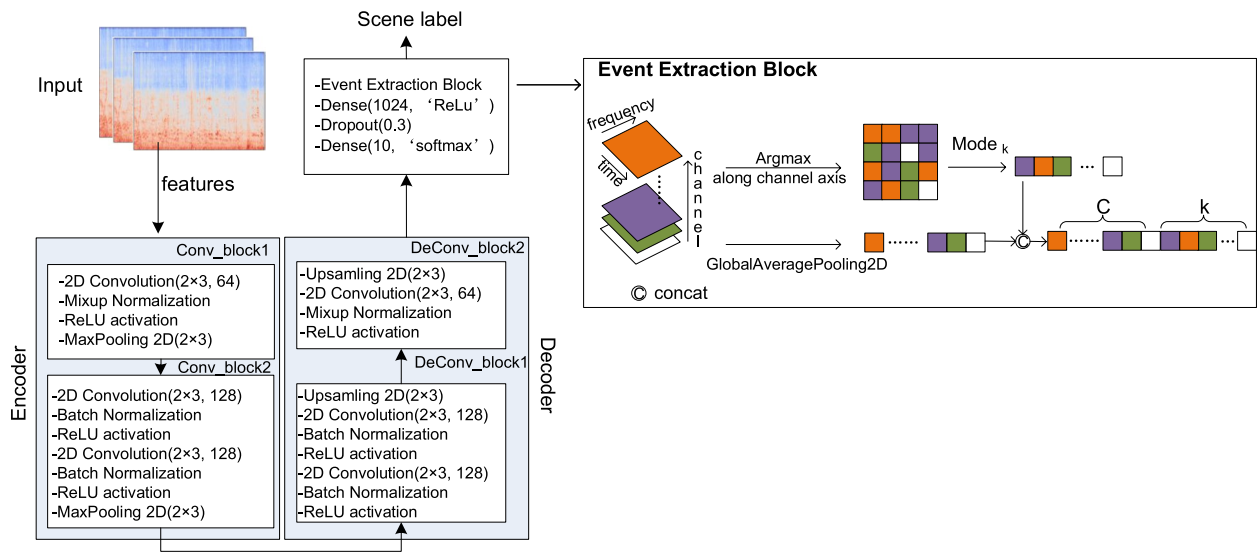


Fig. 2 Details of the Mini-SegNet model and Event Extraction Block

acoustic scenes. In our work, the Base-Mini-SegNet is the most robust and performs best.

As shown in Fig. 2, it is a simple encoder-decoder architecture. It is mainly composed of encoder and decoder modules. First, considering the amount of data, we reduce the number of network layers to maximize the ability of deep learning. Secondly, we modify the original 3×3 convolution kernel to 2×3 and get better performance in our experiment. The number maps in the encoder are 64, 128. The number of feature maps in the decoder are 128, 64. The encoder module, consists of two Conv blocks. In the first Conv_block 1, it contains a 2D Convolution layer whose kernel size is 2×3, and the number of filters is 64, then followed by a normalization, a ReLU non-linearity, and a maxpooling whose pool size is 2×3. The second is Conv_block 2, which is a 2D Convolution layer with kernel size is 2×3 and 128 filters with a batch normalization and a ReLU non-linearity. After that, the corresponding convolution, batch normalization, and activation were performed again. In maxpooling layer, the key part of the feature is retained and other weak features are discarded. For each sample, the indices of max locations computed during pooling are stored and passed to the decoder.

The output of the encoder is taken as the input of the decoder module. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map. It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a softmax classifier for classification. The decoder module is similar to the encoder and consists of two DeConv blocks. In each block, upsampling 2D with size of 2×3 is performed first, then followed by convolutional layers, normalization, and ReLU. Finally, the global average pooling layer or the event extraction block is used. Two dense layers with dropout are used to output the final prediction.

3.2 Event extraction block

In this work, we designed an event extractor and applied it to the mini SegNet network structure, as shown in Fig. 2. In the semantic segmentation network, the semantics of this point is represented by calculating the maximum value of the same position of each different channel layer [6]. It can be seen that different channel layers represent different semantic regions. We believe the different semantic segmentation regions in the whole feature graph represent different events. Suppose $x \in \mathbb{R}^{N\times F\times T\times C}$ is the input feature, where N, F, T, and C represent batch size, frequency dimension, time dimension, and the number of channel, respectively.

We first obtain the semantic segmentation tensor $S \in \mathbb{Z}^{N\times F\times T}$, which is the indices of the maximum values of input feature along the channel axis $c \in \{0, 1, \ldots, C-1\}$:

$$S = \arg\max_{c=0}^{C-1}(x) \tag{1}$$

Then, we extract top-k semantic segmentation regions:

$$Y = \mathrm{mode}_k(S) \tag{2}$$

where $\mathrm{mode}_k(\cdot)$ is top-k mode, i.e., top-k numbers that appears the most of S. $Y \in \{0, 1, \ldots, C-1\}^{N\times k}$, where N and C represent batch size and number of channels,

respectively. In our experiment, the best choice of k is 4. At the same time, we find that normalized Y performs better, so we normalize the values of Y by the number of channels so that there will be no large deviation in subsequent learning as below.

$$\tilde{Y} = Y/C \tag{3}$$

Second, the output feature of the final decoder will be fed to the global average pooling layer at the same time. Then, the output feature of the global average pooling layer along the channel axis is concatenated with $\tilde{Y}$ as the output tensor of the event extraction block. The tensor has shape $(N, C + k)$.

Finally, the event extraction block output will be fed to a trainable softmax classifier which consists of an affine transformation followed by the softmax function.

We think that the audio scene is also composed of a variety of audio events, so obtaining the audio events in the audio scene through event extraction block can be used to distinguish similar audio scenes.

3.3 Data augmentation

Data augmentation is an efficient way to avoid overfitting and enhance the model's generalization in deep neural network [51]. We use mixup, ImageDataGenerator, Specaugment, and cropping for data augmentation. We do not use any additional data and train the model from scratch. In our work, data augmentation does improve performance, and we make a detailed comparison in the next section.

Mixup [51] is performed at a mini-batch level: two data batches, along with corresponding labels, are randomly mixed in each training step. Mixup creates a new training sample by mixing a pair of two training samples. It generate a new training sample (x, y) from the data and label pair $(x_1, y_1)(x_2, y_2)$ by the following Eq. (4).

$$\begin{cases} x = \lambda x_1 + (1-\lambda)x_2 \\ y = \lambda y_1 + (1-\lambda)y_2 \end{cases} \tag{4}$$

Here, $\lambda \in [0, 1]$ is acquired by sampling from the beta distribution $\text{Beta}(\alpha, \alpha)$, and α is a hyper parameter. Besides the data x_1 and x_2, it is characteristic to mix the labels y_1 and y_2.

In addition, we tried to use ImageDataGenerator [52] in this task. It is an image generator, mainly used in image classification. At the same time, it can also enhance the data in batches, expand the size of the data set, and enhance the generalization ability of the model. In our work, it is implemented with width shift, height shift.

We additionally used crop augmentation [26] in the temporal axis: each of the two samples combined using mixup was first cropped independently and randomly. Then, we applied Specaugment [53] at a mini-batch level. For a batch of data in the training step, each feature map is randomly masked in both time and frequency axes.

3.4 Mixup Normalization

We found that instance normalization (IN) had a good performance in image style transfer [54]. Its function is equivalent to unifying different pictures into one style. In short, IN can learn domain difference from channel mean and variance in the image domain for better domain style transfer [55, 56]. The audio device ID (A-S3) of official data set differences is revealed along different dimensions of concatenation of mean and standard deviations of the output layer of the Mini-SegNet encoder as shown in 2D Fig. 3. So we use instance normalization to get audio device generalized features in channel dimension as below.

$$\text{IN}(x) = \frac{x - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \epsilon}} \tag{5}$$

where,

$$\mu_{uc} = \frac{1}{\text{FT}} \sum_{f=1}^{\text{F}} \sum_{t=1}^{\text{T}} x_{nftc} \tag{6}$$

$$\sigma_{nc}^2 = \frac{1}{\text{FT}} \sum_{f=1}^{\text{F}} \sum_{t=1}^{\text{T}} (x_{nftc} - \mu_{uf})^2 \tag{7}$$

Here, $\mu_{uc}, \sigma_{nc} \in \mathbb{R}^{N \times C}$, are mean and standard deviation of the input feature $x \in \mathbb{R}^{N \times F \times T \times C}$, where, N, F, T, and C represent batch size, frequency dimension, time dimension, and number of channel, respectively. ϵ is a small number added to σ to avoid division by zero.

As far as we know, direct use of IN can only learn the style information and lose of useful information for classification. In order to compensate the classification information and reduce the influence of excessive IN, we add a hyperparameter λ learned from the Mixup method; we can use the hyperparameter λ to balance the weights on both sides. This normalization method named Mixup Normalization (MixupNorm) is below.

$$\text{MixupNorm}(x) = \lambda x + (1-\lambda)\text{IN}(x) \tag{8}$$

We apply MixupNorm for the first encoder layer and last decoder layer in Fig. 2. There are a total of two MixupNorm modules in the Mini-SegNet network.

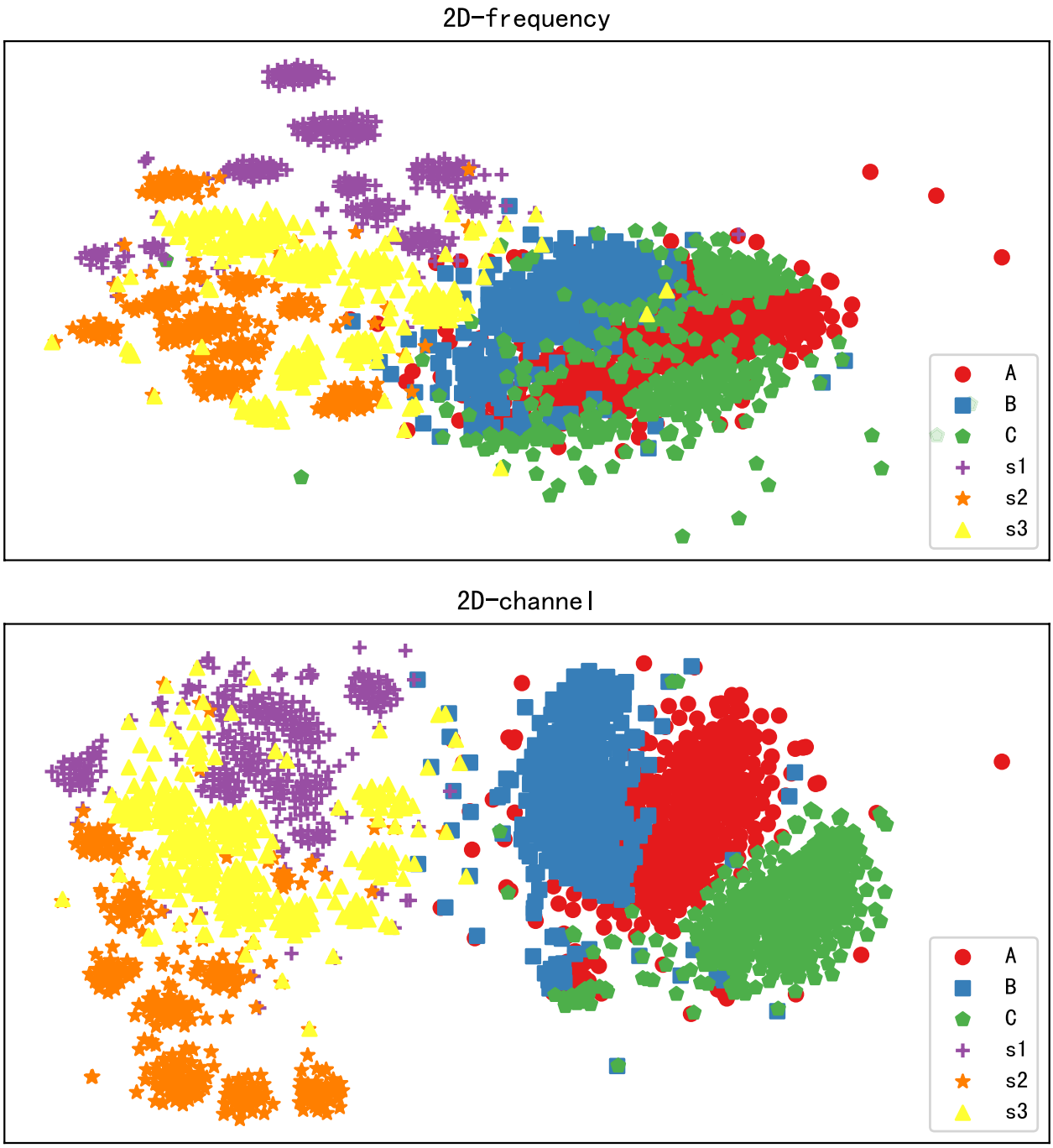


Fig. 3 2D visualization of feature maps of mean and standard deviations. *Top*: frequency-wise. *Bottom*: channel-wise

4 Experiment set up and results

4.1 Dataset

To evaluate our system, we use the Task 1A acoustic scene classification data from the official data set of the TAU Urban Acoustic Scene 2020 Mobile Development dataset [14]. The dataset consists of 10 acoustic scenes: airport, bus, metro, metro_station, park, public_square, shopping_mall, street_pedestrian, street_traffic, tram. The development set contains data from 10 cities and 9 devices, 3 real devices (A, B, C), and 6 simulated devices (S1–S6). Most of the experimental data were collected from high-quality recording device A. The other devices are commonly available customer devices: device B is a Samsung Galaxy S7, device C is iPhone SE, and device

D is a GoPro Hero5 Session. The simulated data are synthesized by processing the data of device A with various impulse responses and dynamic range compression.

The development dataset comprises 40 h of data from device A, and smaller amounts from the other devices. Audio is provided in single-channel 44.1 kHz 24-bit format and was split into 10-s segments that are provided in individual files. The organizer of the challenge provides basic meta data of train/test split consisting of 13,965 samples in the training set and 2970 samples in the test set. As shown in Table 1, some devices (S4, S5, S6) appear only in the test subset. So the device-specific information of S4–S6 cannot be learned in training.

4.2 Experiment setup

We train our models on GPU, with a batch size of 64, and with stochastic gradient descent with a momentum of 0.9 for the optimizer. At the same time, we use a warm restart learning rate schedule [57]; it gets to maximum value of 0.1 after 11, 31, 71, 151, and 311 epochs and then decays according to a cosine pattern to 1×10^{-5}. In our work, we transformed audio data into a power spectrogram by skipping every 1024 samples with 2048 length Hanning window. A spectrum of 431 frames was yielded from 10-s audio file, and each spectrum was compressed into 256 bins of mel frequency scale. Additionally, deltas and delta-deltas were calculated from the log Mel spectrogram and stacked into the channel axis. The number of frames of the input feature was cropped by the length of the delta-delta channel so that the final shape becomes $[256 \times 423 \times 3]$. And each network was trained for 310 epochs. During the training stage, the different data augmentation methods for the dataset for Mini-SegNet are used, and the parameters are set as Mixup with $\alpha = 0.3$, ImageDataGenerator with width_shift_range = 0.6 and height_shift_range = 3, and Specaugment with a temporal mask and two frequency masks with mask parameters of 80 and 30, respectively. The input data is randomly cropped into a fixed-length along the time axis. In our experiments, the input data with the size of $[\ 256 \times 423 \times 3\ ]$ was cropped into $[\ 256 \times 400 \times 3\ ]$ input feature map. We used MixupNorm with $\lambda = 0.1$ and $\lambda = 0$ for comparison on unseen devices.

Table 1 TAU Urban Acoustic Scenes 2020 Mobile Development dataset

Device	Total	Train	Test	(not used in train/test split)
A	14,400	10,215	330	3855
B, C	1080	2×750	2×330	—
S1, S2, S3	1080	3×750	3×330	—
S4, S5, S6	1080	—	3×330	750
Total	23,040	13,965	2970	—

For the train-test split, we adopt the official recommended way to split the development material. There are 13,965 train audio clips and 2970 test audio clips. The training set includes audio from devices A, B, C, and S1–S3. The test set covers data from those six devices and extra data from unseen devices S4, S5, and S6. And we applied data augmentation to increase the diversity of data distribution. The augmented data was generated from each mini-batch consisting of 64 samples during the training process in real-time. Experiments show that this method can improve the accuracy of acoustic scene classification.

For the multi-class classification tasks, cross-entropy (CE) is generally used as the loss function:

$$\mathrm{CE}(p, y) = -\sum_{j=0}^{K} y_j \log(p_j) \tag{9}$$

where p is the model's estimated probability, y is a ground-truth class label (one-hot vector), and j represents the jth class. We adopt the CE loss as the loss function for the proposed model.

4.3 Results and discussion

To illustrate the properties and performance of Audio-SegNet proposed in this paper, we adopt the official recommended way to split the train set and test set on Acoustic Scene 2020 Mobile Development dataset as shown in Table 1. We compared and analyzed various versions of Audio-SegNet, which have different convolution layers and kernel sizes. We also verify the performance of the Mixup Normalization and the event extraction block methods for unseen devices (ID S4-S6 in test) recognition.

4.3.1 Validation results of Mini-SegNet

The DCASE2020 Task 1A challenge [14] is evaluated by the average of the class-wise accuracy, also known as "macro-average accuracy." All the work in this paper is tested on the challenge dataset, because the datasets come from different devices and the train/test setup. Our experimental results are mainly shown by the average accuracy, that is, the average accuracy of scene classification under various devices.

As shown in Table 2, different Audio-SegNet networks have different performances, such as the amount of training parameters, training time, and training accuracy. In our work, according to the structure of Image-SegNet, we first constructed the Audio-SegNet for ASC. In our experiments, we term SegNet-L, which means it is a

Table 2 Results (all-accuracy: %, train/test) of various versions of Audio-SegNet network (kernel_size: 3×3, maxpooling_size: 2×2, upsampling_size: 2×2. 64 × 2 represents 2 convolution layers with 64 output mappings)

Audio-SegNet	SegNet-L	SegNet-M	SegNet-S	Mini-SegNet
Encoder	64 × 2	64 × 2	64 × 2	64 × 1
	128 × 2	128 × 2	128 × 2	128 × 2
	256 × 3	196 × 2		
	512 × 3			
	512 × 3			
Decoder	512 × 3	196 × 2	128 × 2	128 × 2
	512 × 3	128 × 2	64 × 2	64 × 1
	256 × 3	64 × 2		
	128 × 2			
	64 × 2			
Train params	31,880,650	2,051,050	707,338	670,282
Time(s)/Epoch	328	215	206	195
All-accuracy	93.86/59.06	90.84/63.44	85.32/65.35	83.45/66.46

larger Audio-SegNet with more convolution layers and larger kernel size. In Table 2, 64 × 2 represents two convolution layers with 64 output mappings. In the SegNet-L, each encoder network has a corresponding decoder layer and hence the encoder network has 13 convolutional layers. The number of parameters is 31,880,650, and the training time of each epoch is 328 s. The all-accuracy is 93.86% on the train set and 59.06% on the test set. The results show that its performance is poor, especially in the training set and test set there is a large gap. The main reason is that SegNet-L architecture has a deep network, and when our data is limited, it cannot be fully utilized. Therefore, the final classification accuracy has a great problem of overfitting.

The simplest way to prevent overfitting is to reduce the model size, that is, to reduce the number of learnable parameters in the model, which is determined by the number of layers and the number of units in each layer. Therefore, we have made many attempts to modify the depth of the network. SegNet-M, compared to SegNet-L, not only has a smaller convolution layer but also the training parameters are reduced by an order of magnitude. The training time is obviously reduced, and the accuracy of the test set is improved. But there is still overfitting phenomenon. Then, we further try to reduce the number of network layers and construct two kinds of networks, SegNet-S and Mini-SegNet. Mini-SegNet has better performance, less parameters, and shorter training time. At the same time, overfitting has also been alleviated. In our work, our acoustic scene classification is limited by the amount of data and cannot use deep networks like image segmentation. Therefore, after the analysis and test, we build the acoustic scene classification system based on the Mini-SegNet network.

Table 3 Mini-SegNet: performance with different kernel size, maxpooling size, and upsampling size on Mini-SegNet (all-accuracy: %, train/test)

Mini-SegNet			
Kernel size	(3,3)	(2,3)	(1,2)
Maxpooling size	(2,2)	(2,3)	(1,2)
Upsampling size	(2,2)	(2,3)	(1,2)
Train params	670,282	478,218	210,186
All-accuracy	83.45/66.46	80.49/71.26	50.31/47.54

A convolution kernel can be regarded as the weighted summation of a certain part; it corresponds to local perception. Its principle is that when we observe an object, we can neither observe each pixel nor observe the whole at once, but start to understand from the local, which corresponds to convolution. In the same receptive field, the smaller the convolution kernel, the smaller the parameters and computational complexity. In order to extract local features more fully, we compare the recognition performance of several different convolution kernel sizes.

Table 3 shows the accuracy of different kernel sizes, maxpooling size, and upsampling size on the basis of the Mini-SegNet network. In our work, we initially kept the original Image-SegNet kernel size configuration. However, there is still overfitting phenomenon in Mini-SegNet. Therefore, we analyze and test the different sizes of the convolution kernels. From Table 3, we can see that the problem of overfitting can be improved by reducing the kernel size to a certain extent. When the kernel size is equal to 2 × 3, the classification performance of the system is the best.

The accuracy on the training set is 80.49%, and that on the test set is 71.26%. At this time, overfitting problems can be ignored. Compared with the 3 × 3 kernel, the overall performance, such as the amount of training parameters, training time, and accuracy, has been improved. However, if we further reduce the convolution kernel size to 1 × 2, the accuracy is very poor. In Fig. 4, we analyzed the characteristic maps of frequency-wise and channel-wise of the Mini-SegNet encoder output. We found that the 2 × 3 kennel size for the channel-wise of frequency would be better between 80 and 100 than 3 × 3. Compared with the 3 × 3 kernel size, the 2 × 3 kernel size has higher spectral density at channel dimensions 80 to 100. Therefore, when the feature maps of the decoder output are pooled using the

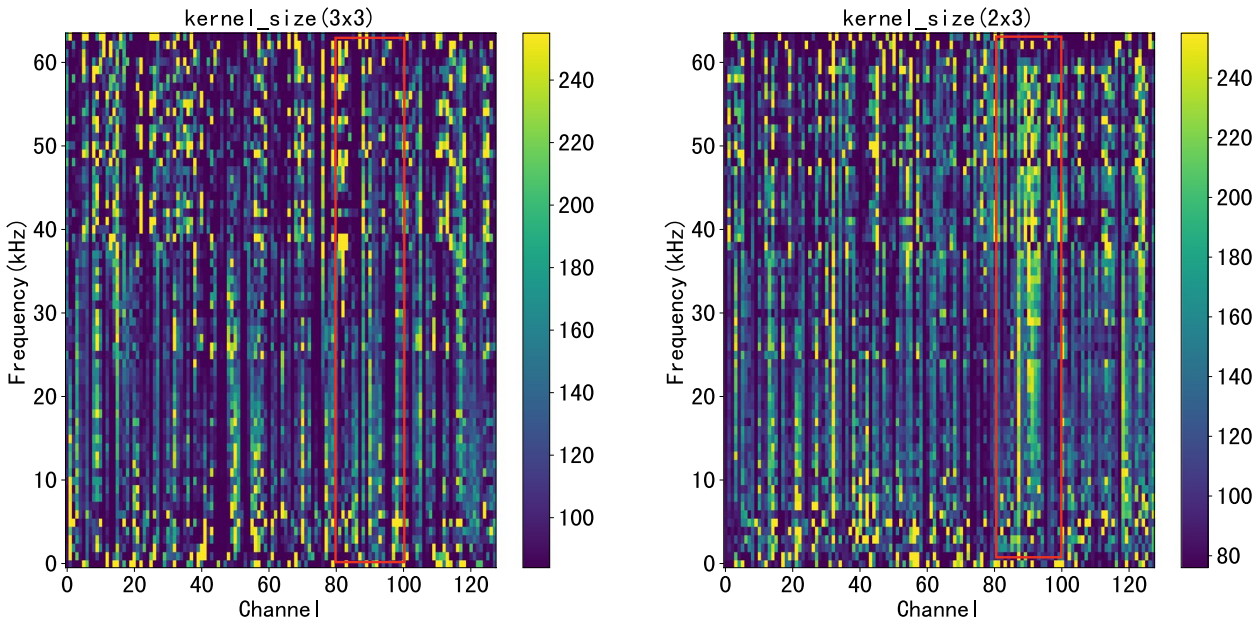


Fig. 4 Compare the performance of kernel size 2 × 3 and 3 × 3 in channel-wise 80 to 100. *Left*: the feature map with 3 × 3 kennel size of the Mini-SegNet encoder output. *Right*:the feature map with 2 × 3 kennel size of the Mini-SegNet encoder output

Table 4 All-accuracy (%) under various data enhancement methods

Mixup	Yes	Yes	Yes	Yes
ImageDtaeGenerator	Yes	No	Yes	Yes
Temporal crop	Yes	Yes	No	Yes
Specaugment	Yes	Yes	Yes	No
All-accuracy(train/test)	80.49/71.26	90.51/69.27	77.56/67.34	85.56/68.32

global average pooling layer along the channel axis, the distinction of 2 × 3 features will be higher than 3 × 3. So our acoustic scene classification system is a Mini-SegNet network with convolution kernel size of 2 × 3. Finally, the average classification accuracy is 71.26%. The total training parameters are 478,218, and the average training time is 173 s under 310 epochs.

Meanwhile, we use a variety of data enhancement methods to further improve the classification accuracy, without using additional data. Table 4 shows results for Mini-SegNet trained in various configurations using the official test-train split. Every configuration was tested on both architectures.

In [58], mixup data augmentation on acoustic scene classification has been fully verified. Therefore, we use mixup directly in our work. Then, we try and analyze the methods of temporal crop, Specaugment, and ImageDataGenerator respectively. The results in Table 4 show that temporal crop, Specaugment, and ImageDataGenerator improve performance in acoustic scene classification. It not only improves the overall classification accuracy, but also alleviates the problem of overfitting. In our parameter set, we set the width_shift_range as 0.6, which is divided by the total width. And the height_shift_range is 3, that is, the amplitude of random vertical offset of the image when the data is raised. To a certain extent, it shows that the sound signal contains more information in the frequency domain, and the experimental results also prove that. In general, ImageDataGenerator method based on image data augmentation can also be well applied to acoustic scene classification. As shown in Fig. 5, the proposed system with a warm restart learning rate schedule achieves better performance in the development set than simpler linear learning rate schedule.

4.3.2 Mixup Normalization and event extraction block

We test the Mixup Normalization and the event extraction block methods in the Mini-SegNet network structure and compare them with batch normalization (BN). The baseline is Mini-SegNet, and as shown in Fig. 2, we only used a global average pooling instead of the event extraction block when it is omitted. The results are shown in Table 5.

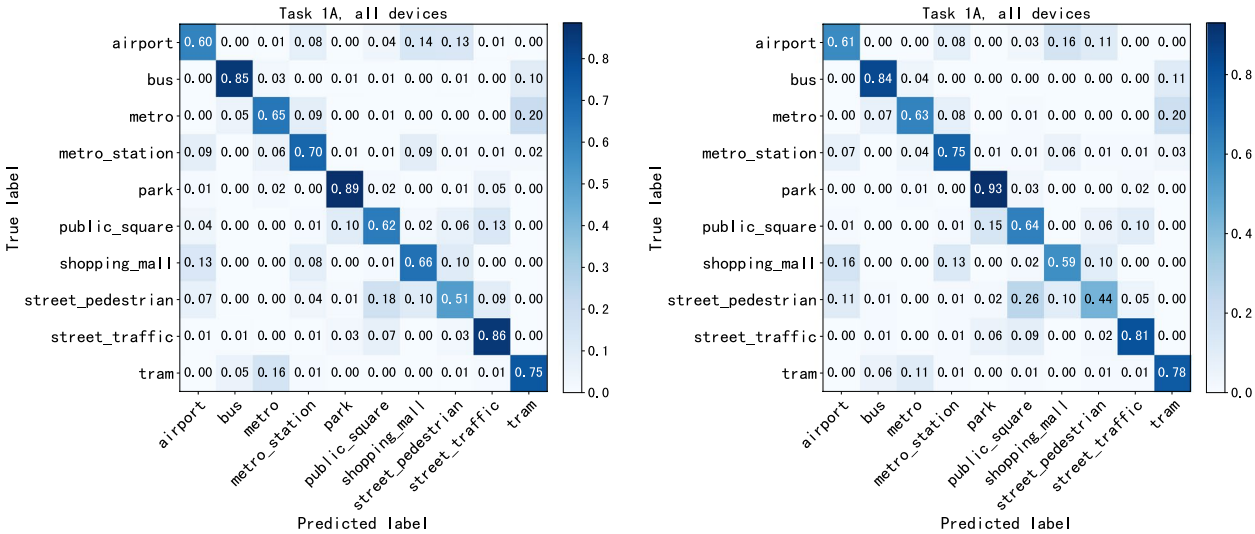
In Table 5, the average accuracy (A-S6) of mini-SegNet is 65.93% with BN, 69.82% with IN, 70.11% with MixupNorm, and 70.97% with MixupNorm and event

Fig. 5 Accuracy of proposed system (310 epochs). *Top*: with warm restart learning rate schedule. *Bottom*: without warm restart learning rate schedule

Table 5 Experimental results on Task 1A. Mixup Normalization and event extraction block are efficient on unseen devices (S4–S6) on TAU Urban AcousticScenes 2020 Mobile, Development dataset

	Devices									
Method	**A**	**B**	**C**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**Overall**
Mini-SegNet+BN	74.24	71.43	**74.77**	63.93	66.36	66.66	63.33	63.63	55.45	65.93±0.7
Mini-SegNet+IN	**78.48**	72.64	71.73	70.30	67.27	73.94	67.27	65.15	65.45	69.82±0.4
Mini-SegNet +MixupNorm	78.27	**72.82**	72.03	68.45	68.18	**74.24**	68.45	66.52	66.36	70.11±0.3
Mini-SegNet +MixupNorm +EventExtractionBlock	76.67	71.34	74.47	**70.61**	**69.39**	71.52	**69.70**	**69.70**	**67.85**	70.97±0.3

extraction block. The result of IN is 3.89% better than BN, and MixupNorm is 4.18% improvements compared to BN. For the unseen device (ID S4–S6) on the test set, "S4–S6" had an average accuracy of 67.11% using MixupNorm, which is more than 7% and 1% better than BN and IN, respectively. The MixupNorm is



(a) with Event Extraction Block (b) without Event Extraction Block

Fig. 6 The confusion matrix of average classification results (all-accuracy) under various devices in Mini-SegNet network. **a** With event extraction block. **b** Without event extraction block

Table 6 The Effects of hyperparameter λ of Mixup Normalization on TAU Urban AcousticScenes 2020 Mobile, Development dataset

Hyperparameter λ	0.05	0.1	0.2	0.5
All-accuracy	80.19/70.13	80.49/71.26	83.70/69.70	85.25/68.53

IN when the hyperparameter λ is 0 in Eq. 8. In addition, the event extraction block is effective on unseen devices.

We chose the average accuracy for various recording devices (all-accuracy) as the main performance because the task targets generalization properties of systems across a number of different devices. The confusion matrix of acoustic scene classification results under all devices is shown in Fig. 6a. From this figure, it can be seen that the generalization ability on some classes is better, with an accuracy of up to 85% in the recognition of acoustic scenes such as bus, park, and street_traffic. Comparing Fig. 6a and b, we found that the event extraction block effectively reduces the error rate of mutual recognition of similar scenes, such as airport and shopping_mall, street_pedestrian, and public_square. As shown in Table 6, we performed ablation experiments for the hyperparameter λ of the Mixup Normalization method. The performance is the best when the parameter λ is set to 0.1.

Table 7 Comparison with recent state-of-the-art systems using the performance of individual systems without a score-level ensemble. The third to seventh rows list the top five best-performing systems on the DCASE2020 Task 1A challenge. The ninth to 13th rows list the top five best-performing systems on the DCASE2021 Task 1A challenge

System	Acc(%)	#Params	Compression methods
Proposed method	**71.26**	**478K**	-
DCASE2020 Baseline [5]	54.1	5M	-
Suh et al. [31]	73.7	13M	-
Hu et al. [30]	76.9	130M	-
Gao et al. [32]	71.8	4M	-
Liu et al. [59]	72.1	3M	-
Koutini et al. [60]	71.8	225M	-
DCASE2021 Baseline [22]	46.9	46K	Quantization
Kim et al. [34]	76.3	315K	Quantization/pruning/knowledge distillation
Yang et al. [37]	78.3	3.6K	LTH/knowledge distillation
Koutini et al. [35]	69.5	631K	Sparsify/quantization
Heo et al. [36]	70.5	65K	Knowledge distillation
Liu et al. [61]	68.2	648K	Quantization

4.3.3 Comparison with recent state-of-the-art systems

Table 7 compares our proposed Mini-SegNet network with current state-of-the-art systems without applying ensemble techniques. Compared with systems in DCASE2020 Task 1A challenge, our proposed system has comparable performance and lower complexity. On

DCASE2021 Task 1A challenge, the model complexity limit of 128 KB was set for the non-zero parameters. Therefore, many model compression methods were used or proposed by researchers, such as knowledge distillation and LTH [37, 38]. Compared with the top five best-performing systems on the DCASE2021 Task 1A challenge, the proposed system does not use any compression method, so we do not need additional resources and time to train a complex network and then compress the model, such as knowledge distillation. The proposed system still has comparable performance on systems with similar parameters.

The acoustic scene classification system proposed in this paper takes log-mel spectrum as the acoustic feature and Mini-SegNet as the classifier. Our proposed system achieved 71.26% on the different devices on the development dataset.

5 Conclusions

In this paper, we proposed a new method of audio semantic segmentation for acoustic scene classification with multiple devices. Based on the paradigm of encoder-decoder, we introduced a method for extracting multi-granularity features of sound events in acoustic scenes. We explored several architectures of Audio-SegNet for the audio scene classification. The best result was achieved on Mini-SenNet with event extraction block and Mixup Normalization network. The experimental results showed encouraging findings that audio semantic segmentation with event extraction block and Mixup Normalization can be effective in extracting features for recognition of acoustic scenes. The proposed method has lower complexity and higher accuracy compared with other classic CNN models using the public DCASE 2020 Task 1A database.

Abbreviations

Image-SegNet	Image semantic segmentation
ASC	Acoustic scene classification
DCASE	Detection and Classification of Acoustic Scene and Events
AASP	Audio and Acoustic Signal Processing
GMMs	Gaussian Mixture Models
SVMs	Support Vector Machines
HMMs	Hidden Markov Models
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CNNs	Convolutional Neural Networks
ZCR	Zero Crossing Rate
PLP	Perceptual Linear Prediction
MFCC	Mel Frequency Coefficients
CQT	Constant-Q transform
HPSS	Harmonic percussive source separation
LSTM	Long Short-Term Memory units
GAN	Generative Adversarial Network
STFT	Short-time Fourier transform
CE	Cross-entropy
LTH	Lottery Ticket Hypothesis

Acknowledgements

Authors would like to thank the reviewers and editors for their effort in the improvement of this manuscript. We gratefully acknowledge the Detection and Classification of Acoustic Scene and Events challenge, for providing us a large amount of open source audio data.

Authors' contributions

YS carried out these approaches, implemented the experiments, and finished this article. XM participated in implementation of the experiments and helped to draft the manuscript. YM and WZ conceived of the study and participated in its design and helped to draft the manuscript. The authors read and approved the final manuscript.

Funding

This work was supported by the National Key R &D Program of China under Grant No. 2020AAA0104500 and the National Natural Science Foundation of China under Grant No. 62276153 and by a grant from the Guoqiang Institute, Tsinghua University.

Availability of data and materials

The datasets supporting the conclusions of this article are available on the internet, https://zenodo.org/record/3819968.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 16 April 2022 Accepted: 1 December 2023
Published online: 03 January 2024

References

1. D. Barchiesi, D. Giannoulis, D. Stowell, M. Plumbley, Acoustic scene classification: classifying environments from the sounds they produce. IEEE Signal Process. Mag. **32**(3), 16–34 (2015)
2. Y. Han, J. Park, Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification. Tech. Rep., DCASE 2017 Challenge (2017)
3. H. Zeinali, L. Burget, J. Cernocky, in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*. Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge (Zenodo, Geneve, 2018)
4. Y. Sakashita, M. Aono, Acoustic scene classifification by ensemble of spectrograms based on adaptive temporal divisions. Tech. Rep., DCASE 2018 Challenge (2018)
5. DCASE. Detection and classification of acoustic scenes and events 2020 task 1a (2020), https://dcase.community/challenge2020/task-acoustic-scene-classification-results-a
6. V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
7. X. Ma, Y. Shao, Y. Ma, W.Q. Zhang, in *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Deep semantic encoder-decoder network for acoustic scene classification with multiple devices (IEEE, Piscataway, NJ, 2020), pp. 365–370
8. X. Ma, Y. Shao, Y. Ma, W.-Q. Zhang, THUEE submission for DCASE 2020 challenge task1a. Tech. Rep., DCASE 2020 Challenge (2020)
9. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, (2015), https://arxiv.org/abs/1502.03167
10. D. Ulyanov, A. Vedaldi, V. Lempitsky, *in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. (IEEE, Piscataway, 2017), pp.4105–4113
11. DCASE. Detection and classification of acoustic scenes and events (2020), http://dcase.community/

12. DCASE. Detection and classification of acoustic scenes and events challenge 2018 (2018), http://dcase.community/challenge2018/task-acoustic-scene-classification-results-a
13. DCASE. Detection and classification of acoustic scenes and events challenge 2019 (2019), http://dcase.community/challenge2019/task-acoustic-scene-classification#subtask-a
14. DCASE. Detection and classification of acoustic scenes and events challenge 2020 (2020), http://dcase.community/challenge2020/task-acoustic-scene-classification#subtask-a
15. J.T. Geiger, B. Schuller, G. Rigoll, in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. Large-scale audio feature extraction and svm for acoustic scene classification (IEEE, Piscataway, 2013), pp. 1–4
16. S. Mun, S. Park, Y. Lee, H. Ko, Deep neural network bottleneck feature for acoustic scene classification. Tech. Rep., DCASE 2016 Challenge (2016)
17. G. Vikaskumar, S. Waldekar, D. Paul, G. Saha, Acoustic scene classification using block based mfcc features. Tech. Rep., DCASE 2016 Challenge (2016)
18. W. Zheng, J. Yi, X. Xing, X. Liu, S. Peng, in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion (Zenodo, Geneve, 2017)
19. A. Schindler, T. Lidy, A. Rauber, in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. Multi-temporal resolution convolutional neural networks for acoustic scene classification (Zenodo, Geneve, 2017)
20. H. Wang, Y. Zou, D. Chong, Acoustic scene classification with spectrogram processing strategies, (2020), https://arxiv.org/abs/2007.03781
21. J. Sun, X. Liu, X. Mei, J. Zhao, M.D. Plumbley, V. Kılıç, W. Wang, in *30th European Signal Processing Conference (EUSIPCO)*. Deep neural decision forest for acoustic scene classification (IEEE, Piscataway, 2022), pp. 772–776
22. DCASE. Low-complexity acoustic scene classification with multiple devices (2021), https://dcase.community/challenge2021/task-acoustic-scene-classification-results-a
23. J. Salamon, J.P. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process. Lett. **24**(3), 279–283 (2017)
24. S.H. Bae, I. Choi, N.S. Kim., in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. Acoustic scene classification using parallel combination of LSTM and CNN (Zenodo, Geneve, 2017)
25. S. Phaye, E. Benetos, Y. Wang, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Subspectralnet-using sub-spectrogram based convolutional neural networks for acoustic scene classification (IEEE, Piscataway, 2019), pp. 825–829
26. M.D. McDonnell, W. Gao, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths (IEEE, Piscataway, 2020), pp. 141–145
27. H. Wang, Y. Zou, W. Wang, in *Interspeech 2021*. SpecAugment++: A hidden space data augmentation method for acoustic scene classification (ISCA, Baixas, 2021), pp. 551–555
28. I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks. Adv. Neural Inf. Process Syst. **3**, 2672–2680 (2014)
29. S. Mun, S. Park, D. Han, H. Ko, Generative adversial network based acoustic scene training set augmentation and selection using svm hyper-planne. Tech. Rep., DCASE 2017 Challenge (2017)
30. H. Hu, C.H.H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S.M. Siniscalchi, Y. Wang, J. Du, C.H. Lee, Device-robust acoustic scene classification based on two-stage categorization and data augmentation. Tech. Rep., DCASE 2020 Challenge (2020)
31. S. Suh, S. Park, Y. Jeong, T. Lee, Designing acoustic scene classification models with cnn variants. Tech. Rep., DCASE 2020 Challenge (2020)
32. W. Gao, M.D. McDonnell, Coustic scene classification using deep residual networks with focal loss and mild domain adaptation. Tech. Rep., DCASE 2020 Challenge (2020)
33. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, (2017), https://arxiv.org/abs/1704.04861
34. B. Kim, S. Yang, J. Kim, S. Chang, QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design. Tech. Rep., DCASE 2021 Challenge (2021)
35. K. Koutini, S. Jan, G. Widmer, Cpjku submission to DCASE21: Cross-device audio scene classification with wide sparse frequency-damped CNNs. Tech. Rep., DCASE 2021 Challenge (2021)
36. H.S. Heo, J.w. Jung, H.j. Shim, B.J. Lee, Clova submission for the DCASE 2021 challenge: Acoustic scene classification using light architectures and device augmentation. Tech. Rep., DCASE 2021 Challenge (2021)
37. H. Yen, C.H.H. Yang, H. Hu, S.M. Siniscalchi, Q. Wang, Y. Wang, X. Xia, Y. Zhao, Y. Wu, Y. Wang, J. Du, C.H. Lee, A lottery ticket hypothesis framework for low-complexity device-robust neural acoustic scene classification, (2021), https://arxiv.org/abs/2107.01461
38. J. Frankle, M. Carbin, The lottery ticket hypothesis: Finding sparse, trainable neural networks, (2018), https://arxiv.org/abs/1803.03635
39. T.Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Feature pyramid networks for object detection (IEEE, Piscataway, 2017), pp. 936–944
40. E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2017)
41. O. Ronneberger, P. Fischer, T. Brox, in *18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. U-Net: Convolutional networks for biomedical image segmentation (Springer, Berlin, 2015), pp. 234–241
42. L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected CRFs (2014), https://arxiv.org/abs/1412.7062
43. L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
44. L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, in *European conference on computer vision (ECCV)*. Encoder-decoder with atrous separable convolution for semantic image segmentation (Springer, Berlin, 2018), pp. 801–818
45. Q. Kong, Y. Cao, H. Liu, K. Choi, Y. Wang, Decoupling magnitude and phase estimation with deep resunet for music source separation, (2021), https://arxiv.org/abs/2109.05418
46. A. Cohen-Hadria, A. Roebel, G. Peeters, in *27th European Signal Processing Conference (EUSIPCO)*. Improving singing voice separation using deep u-net and wave-u-net with data augmentation (Springer, Berlin, 2019), pp. 1–5
47. Y. Liu, B. Thoshkahna, A. Milani, T. Kristjansson, Voice and accompaniment separation in music using self-attention convolutional neural network, (2020), https://arxiv.org/abs/2003.08954
48. H. Huang, K. Wang, Y. Hu, S. Li, in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Encoder-decoder based pitch tracking and joint model training for mandarin tone classification (IEEE, Piscataway, 2021), pp. 6943–6947
49. H. Meng, T. Yan, F. Yuan, H. Wei, Speech emotion recognition from 3D log-mel spectrograms with deep learning network. IEEE Access **7**, 125868–125881 (2019)
50. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
51. H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, (2017), https://arxiv.org/abs/1710.09412
52. K. Wilkinghoff, F. Kurth, Open-set acoustic scene classification with deep convolutional autoencoders. Tech. Rep., DCASE 2019 Challenge (2019)
53. D.S. Park, W. Chan, Y. Zhang, C.C. Chiu, B. Zoph, E.D. Cubuk, Q.V. Le, in *Interspeech*. SpecAugment: A simple data augmentation method for automatic speech recognition (ISCA, Baixas, 2019), pp. 2613–2617
54. D. Ulyanov, V. Lebedev, A. Vedaldi, V. Lempitsky, Texture networks: Feed-forward synthesis of textures and stylized images. Proc. 33rd Int. Conf. Int. Conf. Mach. Learn. **48**, 1349-1357 (2016)
55. X. Huang, S. Belongie, in *2017 IEEE International Conference on Computer Vision (ICCV)*. Arbitrary style transfer in real-time with adaptive instance normalization (IEEE, Piscataway, NJ, 2017), pp. 1510–1519
56. D. Jung, S. Yang, J. Choi, C. Kim, in *2020 IEEE International Conference on Image Processing (ICIP)*. Arbitrary style transfer using graph instance normalization. (IEEE, Piscataway, 2020), pp. 1596–1600

57. I. Loshchilov, F. Hutter, SGDR: Stochastic gradient descent with warm restarts, (2016), https://arxiv.org/abs/1608.03983
58. A. Dang, T.H. Vu, J.C. Wang, in *IEEE International Conference on Consumer Electronics (ICCE)*. Acoustic scene classification using convolutional neural networks and multi-scale multi-feature extraction (IEEE, Piscataway, 2018), pp. 1–4
59. L. Jie, Acoustic scene classification with residual networks and attention mechanism. Tech. Rep., DCASE 2020 Challenge (2020)
60. K. Koutini, F. Henkel, H. Eghbal-zadeh, G. Widmer, Cpjku submissions to DCASE20: Low-complexity cross-device acoustic scene classification with rf-regularized cnns. Tech. Rep., DCASE 2020 Challenge (2020)
61. Y. Liu, J. Liang, L. Zhao, J. Liu, K. Zhao, W. Liu, L. Zhang, T. Xu, C. Shi, DCASE 2021 task 1 subtask a: Low-complexity acoustic scene classification. Tech. Rep., DCASE 2021 Challenge (2021)

Publisher's Note