Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Robust acoustic scene classification using a multi-spectrogram encoder-decoder framework

Lam Pham [a,*], Huy Phan [b], Truc Nguyen [c], Ramaswamy Palaniappan [a], Alfred Mertins [d], Ian McLoughlin [e]

[a] University of Kent, School of Computing, Medway, Kent, UK
[b] School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[c] Signal Processing and Speech Communication Lab, Graz University of Technology, Austria
[d] Institute for Signal Processing, University of Lübeck, Germany
[e] Singapore Institute of Technology, Singapore

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Acoustic scene classification
Encoder-decoder network
Low-level features
High-level features
Multi-spectrogram

ABSTRACT

This article proposes an encoder-decoder network model for Acoustic Scene Classification (ASC), the task of identifying the scene of an audio recording from its acoustic signature. We make use of multiple low-level spectrogram features at the front-end, transformed into higher level features through a well-trained *CNN-DNN* front-end encoder. The high-level features and their combination (via a trained feature combiner) are then fed into different decoder models comprising random forest regression, DNNs and a mixture of experts, for back-end classification. We conduct extensive experiments to evaluate the performance of this framework on various ASC datasets, including LITIS Rouen and IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) 2016 Task 1, 2017 Task 1, 2018 Tasks 1A & 1B and 2019 Tasks 1A & 1B. The experimental results highlight two main contributions; the first is an effective method for high-level feature extraction from multi-spectrogram input via the novel *CNN-DNN* architecture encoder network, and the second is the proposed decoder which enables the framework to achieve competitive results on various datasets. The fact that a single framework is highly competitive for several different challenges is an indicator of its robustness for performing general ASC tasks.

1. Introduction

Considering a general recording of an acoustic environment, this can be said to contain both a background sound field as well as various foreground events. If we regard the background as noise and the foreground as signal, we would find that the signal-to-noise ratio exhibits high variability due to the diverse range of environments and recording conditions. To complicate matters further, a lengthy sound event could be considered background in certain contexts and foreground in others. For instance, a *pedestrian street* recording may have a generally quiet background, but with short *engine* foreground events, as traffic passes. However, a lengthy *engine* sound in an *on bus* recording would be considered a background sound. Furthermore, both background and foreground contain true noise – continuous, periodic or aperiodic acoustic signals that interfere with the understanding of the scene. These variabilities and difficulties make acoustic scene classification (ASC) particularly challenging.

To deal with such challenges, recent ASC papers have tended to focus on two main machine hearing areas. The first aims to solve the lack of discriminative information by exploiting various methods of trained low-level feature extraction. In particular, researchers transform input audio data into one-dimentional frame-based [1] or two-dimensional spectrogram representations [2] to be fed into a back-end classifier.

Frame-based representations often utilise Mel Frequency Cepstral Coefficients (MFCC) [3], providing powerful feature extraction capabilities, which is borrowed from the automatic speech recognition (ASR) community. MFCCs are often combined with other low-level features such as intensity, zero-crossing rate, etc. [1], or modified features such as perceptual linear prediction (PLP), power nomalised cepstral coefficients (PNCC), robust compressive gamma-chirp filter-bank cepstral coefficients (RCGCC) or subspace projection cepstral coefficients (SPCC) [4]. Some systems first transform audio into spectrograms, then attempt to learn different aspects of those spectrograms to extract frame-based features. For

* Corresponding author.
E-mail address: ldp7@kent.ac.uk (L. Pham).

https://doi.org/10.1016/j.dsp.2020.102943
1051-2004/

instance, Alain et al. [5] applied non-negative matrix factorisation (NMF) techniques over a log-mel spectrogram. Meanwhile Song et al. [6] applied the auditory statistics of a cochlear filter model to extract discriminative features directly from audio signals. Conventionally, frame-based approaches are combined with machine learning methods, such as Gaussian mixture models (GMM) [4,7], support vector machines (SVM) [1,6] and so on, for the role of back-end classification. Spectrogram-based approaches use linear, log-mel or other short-time fast Fourier transform (STFT) spectra, stacked into a two-dimensional image. Recent papers have utilised diverse combinations of log-mel and different types of spectrograms such as mel-based nearest neighbour filter (NNF) spectrogram [2], constant-Q transform (CQT) [8], or gammatonegram [9]. To test a wavelet-transform derived spectrogram representation, Ren et al. [10] compared results from STFT spectrograms and both *Bump* and *Morse* scalograms. By exploiting channel information, Sakashita and Aono [11] generated multi-spectrogram inputs from two channels, their average and side channels, and even explored separated harmonic and percussive spectrograms from mono channels to achieve good results. Some papers proposed combining spectrogram and vector features such as log-mel spectrogram and x-vector in [8]. Comparing between frame-based and spectrogram representations, the latter provides richer low-level feature input detail and appears to enable better performance [9,12,13].

Systems using spectrograms [14,15] as low-level features tend to be associated with more complex deep learning classifiers. In general, input spectrograms are first transformed to high-level features containing condensed information before feeding into a final classifier [16]. In some systems, both high-level feature extraction and classification are integrated into one learning process as an end-to-end model. If the high-level feature transformer is well designed and effective at obtaining discriminative features, it is reasonable to assume that final classifier performance will benefit. From this inspiration, the second research trend focuses on constructing and training powerful learning models to transform spectrograms into discriminative higher level features. For instance, Lidy and Schindler [17] proposed two parallel CNN-based models with different kernel sizes to learn from a CQT spectrogram input, capturing both temporal and frequency information. Similarly, Bae et al. [18] applied a parallel recurrent neural network (RNN) to capture sequential correlation and a CNN to capture local spectro-temporal information over an STFT spectrogram. Focusing on pooling layers where high-level features are condensed, Zhao et al. [19,20] proposed an attention pooling layer that showed effective improvement compared to conventional max or mean pooling layers. With the inspiration that different frequency bands in a spectrogram contain distinct features, Phaye et al. [21] proposed a *SubSpectralNet* network which was able to extract discriminative information from 30 sub-spectrograms. Recently, Song et al. [22] proposed a new way to handle distinct features in a sound scene recording, where a deep learning model extracts a bag of features from log-mel spectrograms, including similar and dissimilar ones, then a back-end network exploits this to enhance accuracy.

Looking at the recent approaches surveyed above, we see three main factors explored by all authors: low-level feature input, high-level feature extraction, and output classification. All of these affect final system accuracy. All are chosen in a task-specific way, and no consensus has emerged regarding an optimum choice for any of the three factors. In this paper, we address all three factors in the following way:

1. Firstly, we believe that low-level features often contain valuable and complementary information, hence we develop a method to effectively combine three different spectrogram input features, namely log-mel, gammatone filter (Gamma) and CQT spectrograms.

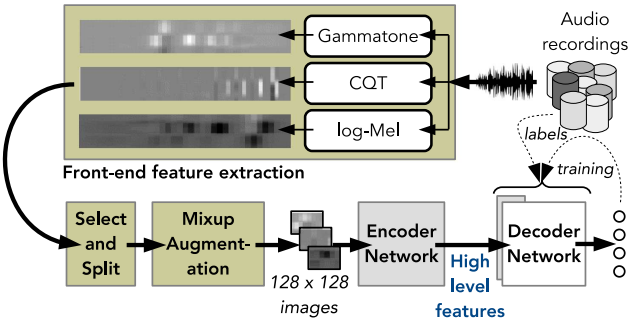


Fig. 1. System view of feature extraction process.

2. To extract high-level features from a multi-spectrogram input, we propose a novel encoder-decoder architecture comprising three parallel *CNN-DNN* paths. Each *CNN* block learns to map one spectrogram into high-level features, and we also combine these high-level features from the middle layers of the networks to form a combined feature.
3. In terms of decoder as final classifier, we evaluate three different models – random regression forest, a deep neural network, and a mixture of experts. We compare the performance of each against state-of-the-art approaches.

Rather than selecting a single task, we evaluate over a wide set: LITIS Rouen, DCASE 2016 Task 1, DCASE 2017 Task 1, DCASE 2018 Tasks 1A & 1B and DCASE 2019 Tasks 1A & 1B. We will see that the performance of our proposed system is competitive with (and for two tasks, outperforms) the state-of-the-art systems. The remainder of this paper is organised as follows. Our motivation for a combined multi-spectrogram approach with a retraining (two-pass) model architecture is described in Section 2. Section 3 describes the evaluation process. Results are discussed in Section 4, and we conclude in Section 5.

2. The proposed system

The overall proposed system is outlined in Fig. 1. Firstly, audio from channel 1 of a recording (some datasets evaluated in this paper have two channels) is represented by a spectrogram. Having tested numerous spectrogram types in our research, we have found – and will demonstrate below – that different spectrograms perform better for different types of scene or task. We therefore design an architecture that is able to effectively combine the benefits of log-mel, gammatonegram (Gamma) and CQT spectrograms. The window size, hop size and number of filters is set empirically to 43 ms, 6 ms and 128 for each spectrogram. Spectrograms are then split into matching non-overlapping patches of size 128×128. We apply mixup data augmentation [23,24], over the patches to increase variation, forcing the learning model to enlarge Fisher's criterion (i.e. the ratio of the between-class distance to the within-class variance in the feature space). After mixup, patches are input to the encoder network.

2.1. Low-level feature with multiple spectrograms

As mentioned in Section 1 and depicted in Fig. 1, we employ three different types of spectrograms as low-level features:

a) Log-mel spectrogram (log-mel) is popular for ASC tasks, appearing in many recent papers. It begins with a set of short-time Fourier transform (STFT) spectra, computed from

$$X[k,m] = \sum_{n=0}^{N-1} x[n+m]w[n]e^{-i2\pi nk/N} \quad (1)$$

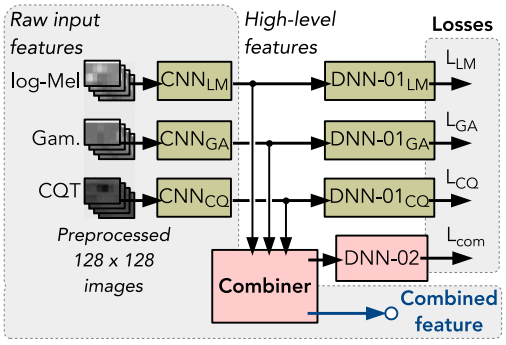


Fig. 2. High-level feature extraction from the encoder network.

where $x[n+m]$ is the discrete audio signal input, $w[n]$ is a window function, typically Hamming, and N is length of audio input signal. A Mel filter bank, which simulates the overall frequency selectivity of the human auditory system using the frequency warping $f_{mel} = 2595 log_{10}(1+f/700)$ [25], is then applied to generate a Mel spectrogram. Logarithmic scaling is applied to obtain the log-mel spectrogram. We use the Librosa toolbox [26] in our experiments.

b) Gammatonegram (Gamma): Gammatone filters are designed to model the frequency-selective cochlea activation response of the human inner ear, as given by

$$g[k] = k^{P-1} T^{P-1} e^{-2b\pi kT} cos(2\pi fkT + \theta) \quad (2)$$

where P is the filter order, θ is the phase of the carrier, b is filter bandwidth, and f is central frequency, and T is sampling period. As with the log-mel spectrogram, the audio signal is first transformed into STFT spectra before applying a gammatone weighting to obtain the gammatone spectrogram, as in [27].

c) Constant Q Transform (CQT): the CQT is designed to model the geometric relationship of pitch, which makes it likely to be effective when undertaking a comparison between natural and artificial sounds, as well as being suitable for frequencies that span several octaves. As with the log-mel spectrogram, we also use Librosa [26] to generate the CQT.

Since these spectrograms are derived from different auditory models, it is plausible that they can each contribute distinct features for classification. This provides an inspiration to explore the three spectrograms. In particular, to design a novel architecture able to extract high-level features from a combination of the three, as described in the following section.

2.2. Encoder network to extract high-level feature

High-level features are extracted by the parallel *CNN-DNN* front-end paths as shown in Fig. 2, referred to as the encoder network. Image patches of size 128×128 pixels, after mixup, are fed into the three parallel networks each of which comprises a *CNN* and a *DNN-01* block, like the VGG-7 architecture [28]. The three parallel networks (each containing *CNN* and *DNN-01*) will learn to extract high-level features from one type of spectrogram each. While the structure of these three CNNs is identical and the structure of the three *DNN-01* blocks is identical, they will contain very different weights after training. In Fig. 2, the three paths are denoted by subscripts LM, GA, and CQ, respectively, referring to the kind of spectrogram they process. The architecture of the *CNN* and *DNN-01* blocks is described in the upper and middle sections of Table 1. While each *CNN* comprises six sub-blocks employing layers of batch normalization (Bn), convolution (Cv [kernel size]), rectified linear units (Relu), average pooling (Ap [kernel size]), global average pooling (Gp), and dropout (Dr(%)) layers, *DNN-01* block comprises of a fully-connected (Fl) followed by a Softmax layer with dimensions given in the table. "C" is the number of classes found within the given dataset, which depends on the particular evaluation task.

Table 1
Encoder network structures of the *CNN* (top), *DNN-01* (middle) and *DNN-02* (bottom).

Network architecture	Output
CNN	
Input layer (image patch)	128×128×1
Bn - Cv [9×9] - Relu - Bn - Ap [2×2] - Dr (10%)	64×64×32
Bn - Cv [7×7] - Relu - Bn - Ap [2×2] - Dr (15%)	32×32×64
Bn - Cv [5×5] - Relu - Bn - Dr (20%)	32×32×128
Bn - Cv [5×5] - Relu - Bn - Ap [2×2] - Dr (20%)	16×16×128
Bn - Cv [3×3] - Relu - Bn - Dr (25%)	16×16×256
Bn - Cv [3×3] - Relu - Bn - Gp - Dr (25%)	256
DNN-01	
Input layer (vector)	256
Fl - Softmax	C
DNN-02	
Input layer (vector)	256
Fl - Dr (30%)	512
Fl - Dr (30%)	1024
Fl - Softmax	C

The output of each of the *CNN* blocks shown in the upper part of Table 1 is a 256-dimensional vector. We refer to the vector extracted from each individual spectrogram, as a high-level feature, and we will explore the relationship between these later. A size of 256 was selected after evaluation of power-of-two dimensions from 64 to 1024 on DCASE 2018 Task 1B.

The *Combiner* block in Fig. 2 has the role of combining the three high-level feature vectors into one composite feature vector. We will evaluate three methods of combining the high-level features. Consider vectors $\mathbf{x_{LM/GA/CQ}}\ [x_1, x_2, ..., x_{256}]$ as the high-level feature outputs of the *CNN* blocks. The first combination method we evaluate, called *sum-comb*, is the unweighted sum of the three vectors. i.e. the individual vectors contribute equally to the combined high-level feature,

$$\mathbf{x_{sum-comb}} = \mathbf{x_{LM}} + \mathbf{x_{GA}} + \mathbf{x_{CQ}} \quad (3)$$

The second method, which is called *max-comb*, obtains $\mathbf{x_{max-comb}}[x_1, x_2, ...$ by selecting the element-wise maximum of the three vectors across the dimensions as in eqn. (4). The motivation is to pick the most important (highest magnitude) feature from among the three high-level feature vectors,

$$\mathbf{x_{max-comb}}[x_i] = max(\mathbf{x_{LM}}[x_i], \mathbf{x_{GA}}[x_i], \mathbf{x_{CQ}}[x_i]) \quad (4)$$

For the final method, we assume elements of three vectors to have a linear relationship across dimensions. We then derive a simple data-driven combination method called *lin-comb* by employing a fully-connected layer trained to weight and combine the three high-level features, as in

$$\mathbf{x_{lin-comb}} = ReLU\{\mathbf{x_{LM}}\mathbf{w_{LM}} + ... \\ \mathbf{x_{GA}}\mathbf{w_{GA}} + \mathbf{x_{CQ}}\mathbf{w_{CQ}} + \mathbf{w_{bias}}\} \quad (5)$$

where $\mathbf{w_{LM/GA/CQ/bias}}[w_1, w_2, ..., w_{256}]$ are the trained parameters. The combined high-level feature vector from the output of the *Combiner* block is then fed into *DNN-02*, with the structure shown in the lower part of Table 1. Note that the combined high-level feature vectors, like the individual high-level vectors, have a dimension of 256 – meaning that the higher layer classifier of the decoder can be set for evaluation with either individual or combined feature input, without changing its structure or complexity.

Regarding training loss, we define four loss functions to train the encoder network; three to optimize individual spectrograms, and the final one for their combination. Eventually, the overall loss function is computed as

$$Loss_{EN} = \alpha(L_{LM} + L_{GA} + L_{CQ}) + \beta L_{com} \quad (6)$$

and L_{LM}, L_{GA}, L_{CQ} and L_{com} are individual losses from the log-mel, Gamma and CQT spectrograms, and their combinations. These are depicted in Fig. 2 and will be defined in Section 3.3. The balancing parameters α and β focus on learning particular features or combinations and are set to 1/3 and 1.0 here, making the contributions from each spectrogram equal.

2.3. Decoders for back-end classification

Our previous work [29], which introduced a *CNN-DNN* structure for individual spectrograms, found mixup to be beneficial for training feature extractors. The new architecture introduces a feature combiner, so we maintain the previous mixup to help train the low-level features, but introduce a second mixup stage, for high-level features when training the decoder. Furthermore, we will evaluate three types of decoder: A random forest regressor (RFR) with classifier, a DNN, and a mixture of experts (MoE), described below

a) Random Forest Regression (RFR-decoder): A regression forest [30] is a type of ensemble model, comprising multiple regression trees. The role of each tree is to map the complex input space defined by the high-level features from the encoder network, into a continuous class-dimension output space. Its nonlinear mapping is achieved by dividing the large original input space into smaller sub-distributions. Individual trees are trained using a subset randomly drawn from the original training set. By using many trees (e.g. 100), the structure is effective at tackling overfitting issues that can occur with single trees. We also believe the regressor structure benefits from the continuous mixed-class training labels provided by employing mixup. Eventually, the decoded output spaces are classified as in our previous work [31] by mean pooling the output over all trees.

b) Deep Neural Network (DNN-decoder): In this paper, we propose a DNN architecture, denoted *DNN-03* for output classification in the decoder. The network comprises four fully connected dense blocks with same dropout (30%), having node sizes of $512 - 1024 - 1024 - C$, where "C" is the number of classes in the task being evaluated. Note that this is similar to the *DNN-02* architecture in Fig. 1, but incorporates one additional fully-connected and one dropout layer, which is useful in practice to refine the accuracy for similar classes.

c) Mixture of Experts (MoE-decoder): An MoE is a machine learning technique that divides the problem spaces into homogeneous regions by using an array of different trained (but in this case identical structure) models, referred to as experts [32]. A conventional MoE architecture comprises many experts and incorporates a gate network to decide which expert is applied in which input region. In this paper, we apply the MoE technique to the combined high-level features, as shown in Fig. 3. Specifically, the 256-dimensional input vector goes through three dense layers with dropout (30%), having 512, 1024 and 1024 hidden nodes, respectively. The output enters the MoE layer, which is expanded in Fig. 3. The combined result from all experts is gated before passing through a softmax to determine the final C class scores. Each MoE expert comprises a dense block with a ReLU activation function. Its input dimension is 1024 and its output size is C. The gate network

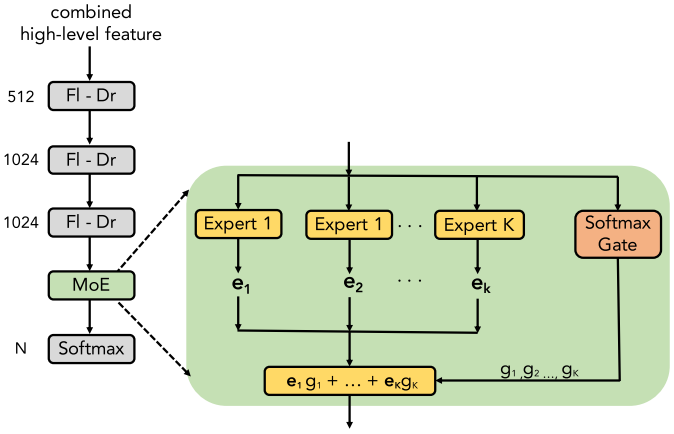


Fig. 3. Proposed mixture of experts within its deep back-end decoder network.

is implemented as a *Softmax Gate* – an additional fully-connected layer with softmax activation and a gating dimension equal to the number of experts.

Let $\mathbf{e_1}, \mathbf{e_2}, \ldots \mathbf{e_K} \in \mathbb{R}^{\mathbb{C}}$ be the output vectors of the K experts, and $g_1, g_2, \ldots, g_K$ be the outputs of the gate network where $g_k \in [0, 1]$, $\sum_{k=1}^{K} g_k = 1$ The predicted output is then found as,

$$\hat{\mathbf{y}} = softmax\left\{\sum_{k=1}^{K} \mathbf{e_k} g_k\right\} \quad (7)$$

3. Evaluation methodology

To clearly demonstrate the general performance of the proposed systems we will evaluate using five different ASC tasks. While it is relatively easy to perform well in one challenge, it is considerably more difficult to do so for all – this helps us to explore one of the hypothesised strengths of our combined-spectrogram approach, that it can be more generic. Four of the datasets we use are derived from annual DCASE challenges, whereas the fifth is the extensive LITIS Rouen dataset. Each is described briefly below.

3.1. DCASE datasets

We adopt datasets from the DCASE 2016 [7], 2017 [33], 2018 [34] and 2019 [35] challenges. For DCASE 2016, both development set (1170 segments) and evaluation set (390 segments) were published and recordings were sampled at 44100 Hz, with 30 s duration per segment over the 15-class challenge. DCASE 2017 reused all DCASE 2016 dataset and added more recording data, with 4680 and 1620 segments for development and evaluation, respectively (segments are of 10 s duration). For DCASE 2016 and 2017, we use the development set (dev. set) for training, and report the classification accuracy over the evaluation set (eva. set). DCASE 2018 and 2019 Task 1A datasets contain 10 s segments, recorded at 48000 Hz and spanning 10 classes. Unlike DCASE 2016 and 2017, these recent challenges only released the development set publicly, providing 8640 and 13370 segments for DCASE 2018 and 2019, respectively. Moreover, DCASE 2018 and 2019 also proposed a different ASC challenge type that involves mismatched recording devices. This is known as Task 1B. Specifically, all recorded segments from device A (Soundman OKM II Klassik/studio A3 electret microphone and a Zoom F8 audio recorder) for the conventional ASC task (the 1A dataset) were reused in Task 1B. Then additional segments were recorded using two different devices B & C (e.g. recorded from a variety of smart phones and cameras) and added, but with unbalanced recording times of 4 and 6 hours respectively.

This is much less than the approximately 24 and 37 hours of device A recordings included in DCASE 2018 and 2019, respectively. In this paper, we follow the setting of the DCASE 2018 challenge; subdividing the development dataset into two subsets; a training set (train set) and a testing set (test set), respectively. For the most recent DCASE 2019 dataset, we report the results over the eva. set (noting that this set has not been released publicly yet), as used for the Kaggle competition associated with the DCASE 2019 challenge.[1]

3.2. LITIS rouen dataset

This extensive dataset [5] comprises 19 urban scene classes with 3026 segments, divided into 20 training/testing splits. The audio was recorded at a sample rate of 22050 Hz, with each segment duration being 30 s. We follow the mandated settings for 20 times cross validation, obtaining final classification accuracy by averaging over the 20 testing folds.

3.3. Experimental setup

All of our proposed networks are built on the Tensorflow framework using cross-entropy loss,

$$Loss(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \mathbf{y_i} log\left\{\hat{\mathbf{y}}_\mathbf{i}(\theta)\right\} + \frac{\lambda}{2}||\theta||_2^2 \quad (8)$$

where $Loss(\theta)$ is the loss function over all parameters θ, constant λ is set to 0.0001, $\mathbf{y_i}$ and $\hat{\mathbf{y}}_\mathbf{i}$ are ground-truth and network output, respectively. Experiments use the Adam optimiser to adjust learning rate, with a batch size of 50. Results were obtained after 200 epochs (in practice we lose only a small degree of performance by not continuing beyond this). As aforementioned, we also performed mixup data augmentation [23,24]. For the pre-training process on the extractor, each of the raw 128×128 dimensional features was repeated twice by including same-dimension beta and Gaussian distribution mixup images of the same dimensions. When training the decoder, we applied mixup to the high-level feature vectors prior to the final classifier. In this case it doubles the number of 256 element feature vectors by including same-length beta distribution mixup vectors. In each case, we incorporated both original and generated mixup data into the training processes to improve performance, at the cost of increasing the training time. The experiment were performed on a Nvidia GPU-V100 with 16Gb RAM for both training and inference jobs. The training time depended on the particular dataset used, but on average each of the sound scene datasets required around four days to train each *Encoder-Decoder* framework entirely. The inference computation for the datasets was much quicker, requiring less than one second for each 10-s audio segment.

4. Experimental results and comparison

In this section we will analyse the performance of the encoder network to specifically understand the contribution made by different spectrogram types, as well as their combinations. We will then analyse the performance of the decoder to assess different back-end classifiers, then compare overall performance to a range of state-of-the art methods.

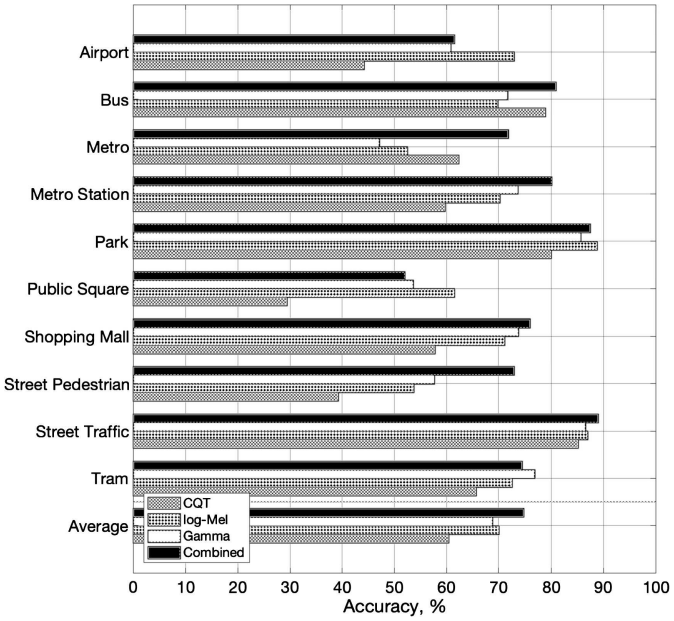


Fig. 4. Performance comparison of different spectrograms types, and their combination, for the DCASE 2018 Task 1B dev. set.

4.1. The performance of each spectrogram by class

We first evaluated the baseline architecture to determine how different spectrogram types contributed to the performance of different classes. To do this, we began by training three *CNN-DNN* encoder networks, comprising *CNN* and *DNN-02* blocks, each encoder for an individual spectrogram input. We trained another *CNN-DNN* encoder network, the entire network as in Fig. 2, for spectrogram combination. These four trained systems were subsequently used as high-level feature extractors to train the decoder and then to test the overall system. We trained four different decoders using the *DNN-03* architecture from Section 2.3 to assess individual spectrogram performance, as well as the performance of the combined high-level features – using the *lin-comb* method from Section 2.2. These experiments were conducted using the DCASE 2018 Task 1B dev. set.

To compare performance, class-wise accuracies for the three spectrograms and their combinations are shown in Fig. 4, with overall average performance shown at the bottom. Clearly, the combined features performed best overall, with the log-mel and gammatonegram performing similarly, and both being better than CQT. However a glance at the per-class accuracy shows some interesting variation. For example, the CQT spectrogram was particularly good at discriminating the *Bus* and *Metro* classes, compared to the other spectrograms. Also, while log-mel and Gamma performances were similar, the former excelled on *Airport* and *Public Square* classes, whereas the latter tended to be slightly better for classes containing vehicular sounds (with the exception of the *Metro* class).

We conclude from this that the three spectrograms represent sounds in ways that have affinity for certain types of sounds (mirroring a conclusion in [36], albeit on very different types of sound data). It is therefore unsurprising that intelligently combining the three spectrograms into a high-level feature vector can achieve significant performance gain over single spectrograms.

[1] 1A: https://www.kaggle.com/c/dcase2019-task1a-leaderboard/overview, 1B: https://www.kaggle.com/c/dcase2019-task1b-leaderboard/leaderboard.

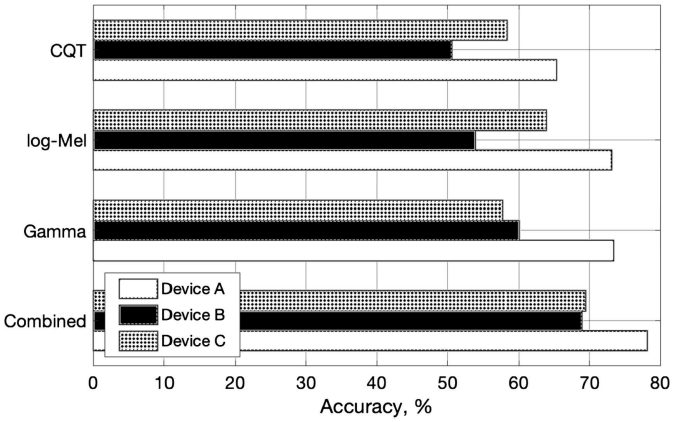


Fig. 5. Performance comparison for different recording devices within the DCASE 2018 Task 1B dev. set.

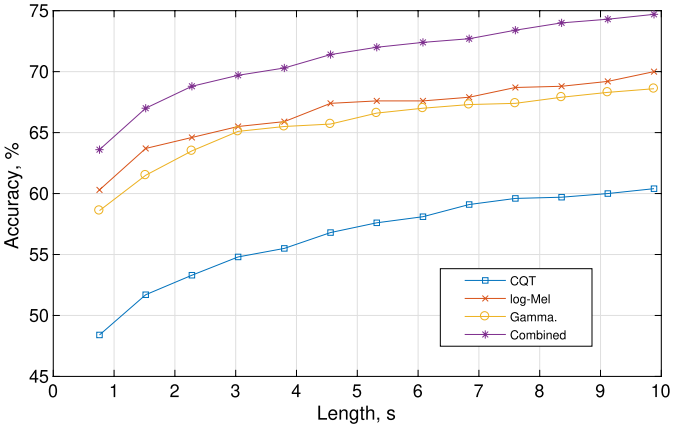


Fig. 6. Classification performance as a function of the length of the test signal over DCASE 2018 Task 1B dev. set - all devices.

4.2. Spectrogram performance for each device

DCASE 2018 task 1B includes highly unbalanced data recordings from three different devices as described in Section 3. We analysed the performance of different spectrograms for those three devices, with results plotted in Fig. 5. The device with the largest amount of training data (Device A) obviously scored best, achieving around 9% better accuracy than devices B and C. Again, the Gamma and log-mel results were similar, but each 'preferred' a different minority device. Although there were not enough devices included in the dataset for the evidence to be conclusive, this variability suggests that spectrograms differ in their affinity for different devices (or device locations, or channels). Again, the combined features effectively leveraged the advantages of each spectrogram type.

4.3. Spectrogram performance by segment length

Inspired by a number of previous works that considered the ability of systems to recognise a sound class early [37,38], we also evaluated this ability for the different spectrogram types. Figs. 6 and 7 plot early classification accuracy for DCASE 2018 Task 1B for all devices and for devices B+C, respectively. Early classification means that class assignment is performed only on the first part of the audio recording, rather than the entire duration (i.e. on cropped audio). Performance is plotted for a number of cropped segment lengths between 1 s and the full 10 s.

From both plots, immediate observations are that the combined high-level features performed much better than the individual spectrogram types. The CQT performed worst while the other

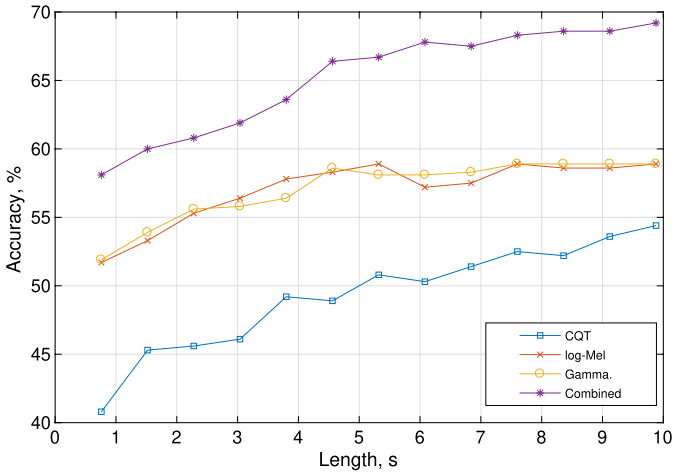


Fig. 7. Classification performance as a function of the length of the test signal over DCASE 2018 Task 1B dev. set - devices B&C.

Table 2
Performance of re-trained models (encoder/decoder Acc. %) over DCASE 2018 Task 1B dev. set.

Device A	RFR-decoder	DNN-decoder	MoE-decoder
sum-comb	71.5/75.6	71.5/72.2	71.5/71.9
max-comb	74.1/75.3	74.1/74.7	74.1/75.5
lin-comb	73.7/75.2	73.7/75.5	73.7/**75.9**
Devices B & C:	**RFR-decoder**	**DNN-decoder**	**MoE-decoder**
sum-comb	63.9/64.4	63.9/65.6	63.9/63.9
max-comb	61.4/65.3	61.4/63.9	61.4/63.9
lin-comb	64.2/68.9	64.2/69.2	64.2/**70.6**

two spectrograms had similar performance (as in the experiments above). Looking closer at Fig. 6 (accuracy for all devices), the score for all features continued to climb as duration progressed towards the full 10 s. This provides a strong indication that the system was data-constrained and is likely to perform better with longer duration recordings.

By contrast, Fig. 7 contains indications that the performance of the log-mel and Gamma spectrograms began to plateau as duration exceeded 5 s, indicating that performance might not substantially increase if longer duration recordings were available. However the continued improvement of the CQT representation as length increased gave the combined features an ability to gain higher accuracy from longer recordings: The strength of CQT may lie in the analysis of longer recordings.

However, in these experiments, CQT performance lagged the combined features by around 15% absolute, with the other spectrograms lagging by only around 5% absolute – apart from the area in Fig. 7 where they plateaued. Most remarkable, though, is that with just 2 s of input data from a recording, our proposed combined high-level feature was able to match or outperform any of the individual spectrograms operating with the full 10 s of input data. This clearly demonstrates a major advantage of the proposed system. It effectively captures the advantages of the individual spectrogram features, which vary in their affinity for different classes and devices, and yields extremely good performance even when a restricted amount of data is available for classification.

4.4. Performance of classifiers in the decoder

Three methods were proposed in Section 2.2 to incorporate the three high-level spectrogram features into a combined high-level feature in the encoder network. These were namely *sum-comb, max-comb* and *lin-comb*. To make use of the combined features, we then introduced three back-end classifier methods for the decoder

Table 3
Comparison to DCASE 2018 baselines for Task 1B dev. set (using *lin-comb* for the pre-training process).

	Device A				Devices B & C			
Classes	D.2018	RFR-decoder	DNN-decoder	MoE-decoder	D.2018	RFR-decoder	DNN-decoder	MoE-decoder
Airport	73.4	67.5	60.4	66.8	72.5	55.6	69.4	75.0
Bus	56.7	78.5	80.2	80.2	78.3	88.9	86.1	88.9
Metro	46.6	67.0	72.8	69.3	20.6	75.0	63.9	66.7
Metro Stn.	52.9	84.6	82.6	80.3	32.8	50.0	61.1	50.0
Park	80.8	89.7	86.8	88.4	59.2	91.7	91.7	94.4
Pub. Sq.	37.9	47.7	52.8	50.9	24.7	52.8	47.2	47.2
Shop. Mall	46.4	74.6	75.3	73.8	61.1	80.6	80.6	80.6
Str. Ped.	55.5	65.6	72.5	71.3	20.8	66.7	75.0	77.8
Str. Traffic	82.5	91.1	90.7	92.3	66.4	75.0	77.8	77.8
Tram	56.5	83.1	79.3	83.1	19.7	52.8	38.9	47.2
Average	**58.9**	**75.2**	**75.5**	**75.9**	**45.6**	**68.9**	**69.2**	**70.6**

Table 4
Comparison of the proposed system (*lin-comb+MoE-decoder*) to state-of-the-art results, with best performance in **bold** (Upper part: Dataset; Middle part: top-ten DCASE challenges; Lower part: State-of-the-art papers).

D.2016 (eva. set)	Acc.	D.2017 (eva. set)	Acc.	D.2018-1A (dev. set)	Acc.	D.2018-1B (dev. set)	Acc.	D.2019-1A (eva. set)	Acc.	D.2019-1B (eva. set)	Acc.	LITIS (20-fold ave.)	Acc
Wei [39]	84.1	Zhao [40]	70.0	Li [41]	72.9	Baseline [34]	45.6	Mingle [42]	79.9	Baseline [35]	61.6	Bisot [43]	93.4
Bae [18]	84.1	Jung [44]	70.6	Jung [45]	73.5	Li [46]	51.7	Wu [47]	80.1	Kong [48]	61.6	Ye [49]	96.0
Kim [50]	85.4	Karol [51]	70.6	Hao [52]	73.6	Tchorz [53]	53.9	Gao [54]	80.5	Waldekar [55]	62.1	Huy [31]	96.4
Takahasi [56]	85.6	Ivan [57]	71.7	Christian [58]	74.7	Kong [59]	57.5	Wang [60]	80.6	Wang [61]	70.3	Yin [62]	96.4
Elizalde [63]	85.9	Park [64]	72.6	Zhang [65]	75.3	Wang [66]	57.5	Jung [67]	81.2	Jiang [68]	70.3	Huy [9]	96.6
Valenti [69]	86.2	Lehner [70]	73.8	Li [46]	76.6	Waldekar [71]	57.8	Huang [72]	81.3	Song [73]	72.2	Ye [74]	97.1
Marchi [1]	86.4	Hyder [75]	74.1	Dang [76]	76.7	Zhao [19]	58.3	Haocong [77]	81.6	Primus [78]	74.2	Huy [79]	97.8
Park [4]	87.2	Zhengh [80]	77.7	Octave [81]	78.4	Truc [2]	63.6	Hyeji [82]	82.5	Hamid [83]	74.5	Zhang [84]	97.9
Bisot [85]	87.7	Han [86]	80.4	Yang [87]	79.8			Hamid [83]	83.8	Gao [54]	74.9	Zhang [88]	98.1
Hamid [89]	89.7	Mun [90]	**83.3**	Golubkov [91]	**80.1**			Chen [92]	**85.2**	Kosmider [93]	**75.3**	Huy [12]	98.7
Mun [94]	86.3	Zhao [10]	64.0	Bai [95]	66.1	Zhao [20]	63.3						
Li [96]	88.1	Yang [97]	69.3	Gao [98]	69.6	Truc [99]	64.7						
Hyder [100]	88.5	Waldekar [101]	69.9	Zhao [20]	72.6	Truc [102]	66.1						
Song [6]	89.5	Wu [103]	75.4	Phaye [21]	74.1								
Yin [62]	**91.0**	Chen [104]	77.1	Heo [105]	77.4								
Our system	88.2	Our system	72.6	Our system	77.5	Our system	**70.6**	Our system	76.8	Our system	72.8	Our system	**98.9**

block, namely *RFR-decoder*, *DNN-decoder* and *MoE-decoder* in Section 2.3. In total, the three classifiers and three combiners yield 9 models to evaluate. In this section, we compare performance among these 9 models on the DCASE 2018 Task 1B dev. dataset. We separately note the accuracy of the encoder network (i.e. the feature extractor, alone), as well as the full system accuracy (i.e. incorporating the decoder).

Results are presented in Table 2, again split into Device A and Device B & C performance. Best performance for both device sets, highlighted in bold, was achieved by the *MoE-decoder* classifier with the *lin-comb* combiner. However some interesting trends were evident. Firstly, *DNN-decoder* was only very slightly inferior to *MoE-decoder* for all combiners and device types. Secondly, looking at the encoder network results for the Device A evaluation, the *max-comb* combiner actually outperformed *lin-comb*, although the latter performed best for most of the full systems. This means that the optimal high-level feature combiner for the full system was not the best combiner for loss computation when training the encoder network. However the situation reverses when looking at Devices B & C – an indication that the performance gain of *lin-comb* may have been due to better generalisation.

4.5. Per-class performance of decoders

Given that the results presented so far indicate that the *lin-comb* combiner performed best, we now feed those high-level combined features into the three alternative decoders to explore class-by-class performance. Table 3 presents results for DCASE 2018 Task 1B (dev. dataset). Device A and Device B & C results are again shown separately, and the "D.2018" column is the DCASE 2018 baseline. Results show that the three classifiers all outperformed the baseline – with the mixture of experts system improving accuracy by 17.0% and 25.0% absolute, for Device A and Devices B & C, respectively.

4.6. Performance comparison to state-of-the-art systems

While performance against the baseline score of DCASE 2018 is good, we now evaluate the same model configuration (i.e. *lin-comb* combiner and *MoE-decoder* back-end classifier) on various datasets and competitions, to compare performance against the state of the art at time of writing. The results, listed in Table 4, show that the system proposed in this paper achieves the highest accuracy for two datasets – achieving 70.6% and 98.9% for DCASE 2018 Task 1B dev. and LITIS Rouen, respectively. For DCASE 2016, an accuracy of 88.2% was achieved, taking second position on the challenge table, and ranked top-four among state-of-the-art systems. DCASE 2017 performance is a little less competitive at 72.6% (note that the system used for that was slightly modified in that it normalized the input data). Our DCASE 2018 Task 1A performance was 77.5%, taking third place on the challenge table. We also entered the system to the recent DCASE 2019 challenge, achieving 76.8% and 72.8% for DCASE 2019 Task 1A and 1B, respectively. The accuracies reported in Table 4 are collected from the latest cited papers and technical reports.

5. Conclusion

This paper has presented a robust framework for acoustic scene classification. Using a feature approach based upon three kinds of time-frequency transformation (namely log-mel, gammatone filterbank and constant Q transform), we presented a two-step training method to first train a front-end encoder network, and then train a decoder to perform back-end classification. To deal with the many challenges implicit in the ASC task, we investigated how the different time-frequency spectrogram types can be combined effectively to improve classification accuracy. In terms of results, the classification accuracy obtained from the proposed system, comprising a trained feature combiner and utilising an MoE-based decoder performs particularly well. As experimental results on DCASE and LITIS Rouen datasets, the proposed method achieves highly competitive results compared to state-of-the-art systems for all tasks, in particular achieving the highest LITIS Rouen and DCASE 2018 Task 1B accuracy at the time of writing.

In future we envisage further exploration related to the *Decoder CNN*. A framework such as VGG will be evaluated with different architectures such as Resnet or Inception. We believe that there is still potential to extract better combined features from the *CNN* part. Secondly, the final pooling layer of the *Decoder*, where high-level features are extracted and condensed, should be investigated. An attention layer may be good approach to replace the final pooling layer for extracting better individual features as well as combined features. There is also potential for multi-scale processing as the duration of the auditory components of scenes and events is inherently non-uniform.

CRediT authorship contribution statement

Lam Pham conceptualized the study, implemented and conducted experimental analysis with engineering support from Truc Nguyen and Huy Phan. Ramaswamy Palaniappan, Alfred Mertins and Ian McLoughlin provided computational resources, support, and guidance. Lam Pham wrote the manuscript, with comments from all other authors.

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.

References

[1] Erik Marchi, Dario Tonelli, Xinzhou Xu, Fabien Ringeval, Jun Deng, Stefano Squartini, Björn Schuller, Pairwise decomposition with deep neural networks and multiscale kernel subspace learning for acoustic scene classification, in: Proc. DCASE, 2016, pp. 65–69.
[2] Truc Nguyen, Franz Pernkopf, Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters, in: Proc. DCASE, 2018, pp. 34–38.
[3] Steven B. Davis, Paul Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. Audio Speech Signal Process. ASSP-28 (4) (1980) 357–366.
[4] Sangwook Park, Seongkyu Mun, Younglo Lee, Hanseok Ko, Score fusion of classification systems for acoustic scene classification, Tech. Rep., DCASE2016 Challenge, 2016.
[5] Alain Rakotomamonjy, Alain Rakotomamonjy, Supervised representation learning for audio scene classification, IEEE/ACM Trans. Audio Speech Lang. Process. 25 (6) (2017) 1253–1265.
[6] Hongwei Song, Jiqing Han, Deng Shiwen, A compact and discriminative feature based on auditory summary statistics for acoustic scene classification, in: Proc. INTERSPEECH, 2018, pp. 3294–3298.
[7] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, TUT database for acoustic scene classification and sound event detection, in: Proc. EUSIPCO, 2016, pp. 1128–1132.
[8] Hossein Zeinali, Lukas Burget, Jan Cernocky, Convolutional neural networks and X-vector embedding for DCASE2018 acoustic scene classification challenge, in: Proc. DCASE, 2018, pp. 202–206.
[9] Huy Phan, Lars Hertel, Marco Maass, Philipp Koch, Radoslaw Mazur, Alfred Mertins, Improved audio scene classification based on label-tree embeddings and convolutional neural networks, IEEE Trans. Audio Speech Lang. 25 (6) (2017) 1278–1290.
[10] Zhao Ren, Kun Qian, Yebin Wang, Zixing Zhang, Vedhas Pandit, Alice Baird, Bjorn Schuller, Deep scalogram representations for acoustic scene classification, IEEE/CAA J. Autom. Sin. 5 (3) (2018) 662–669.
[11] Yuma Sakashita, Masaki Aono, Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions, Tech. Rep., DCASE, 2018.
[12] Huy Phan, Oliver Chan, Lam Pham, Philipp Koch, Maarten de Vos, Ian Mcloughlin, Alfred Mertins, Spatio-temporal attention pooling for audio scene classification, in: Proc. INTERSPEECH, 2019, pp. 3845–3849.
[13] Huy Phan, Oliver Y. Chan, Philipp Koch, Lam Pham, Ian McLoughlin, Alfred Mertins, Maarten De Vos, Beyond equal-length snippets: how long is sufficient to recognize an audio scene?, in: Proc. AES, Jun 2019.
[14] Ian McLoughlin, H.-M. Zhang, Z.-P. Xie, Y. Song, W. Xiao, Robust sound event classification using deep neural networks, IEEE Trans. Audio Speech Lang. 23 (March 2015) 540–552.
[15] Haomin Zhang, Ian McLoughlin, Yan Song, Robust sound event recognition using convolutional neural networks, in: Proc. ICASSP, IEEE, Apr 2015, pp. 559–563.
[16] Ian McLoughlin, Haomin Zhang, Zhipeng Xie, Yan Song, Wei Xiao, Huy Phan, Continuous robust sound event classification using time-frequency features and deep learning, PLoS ONE 12 (9) (2017) e0182309.
[17] Thomas Lidy, Alexander Schindler, CQT-based convolutional neural networks for audio scene classification, in: Proc. DCASE, 2016, pp. 1032–1048.
[18] Soo Hyun Bae, Inkyu Choi, Nam Soo Kim, Acoustic scene classification using parallel combination of LSTM and CNN, in: Proc. DCASE, 2016, pp. 11–15.
[19] Ren Zhao, Kong Qiuqiang, Qian Kun, Mark D. Plumbley, W. Bjorn Schuller, Attention-based convolutional neural networks for acoustic scene classification, in: Proc. DCASE, 2018, pp. 39–43.
[20] Z. Ren, Q. Kong, J. Han, M.D. Plumbley, B.W. Schuller, Attention-based atrous convolutional neural networks: visualisation and understanding perspectives of acoustic scenes, in: Proc. ICASSP, 2019, pp. 56–60.
[21] Sai Phaye, Emmanouil Benetos, Ye Wang, SubSpectralNet using subspectrogram based convolutional neural networks for acoustic scene classification, in: Proc. ICASSP, 2019, pp. 825–829.
[22] Hongwei Song, Jiqing Han, Shiwen Deng, Zhihao Du, Acoustic scene classification by implicitly identifying distinct sound events, in: Proc. INTERSPEECH, 2019, pp. 3860–3864.
[23] Kele Xu, Dawei Feng, Haibo Mi, Boqing Zhu, Dezhi Wang, Lilun Zhang, Hengxing Cai, Shuwen Liu, Mixup-based acoustic scene classification using multi-channel convolutional neural network, in: Pacific Rim Conference on Multimedia, 2018, pp. 14–23.
[24] Yuji Tokozume, Yoshitaka Ushiku, Tatsuya Harada, Learning from between-class examples for deep sound recognition, arXiv preprint, arXiv:1711.10282, 2017.
[25] Ian Vince McLoughlin, Speech and Audio Processing: A MATLAB-Based Approach, Cambridge University Press, 2016.
[26] Brian McFee, Raffel Colin, Liang Dawen, Daniel P.W. Ellis, McVicar Matt, Battenberg Eric, Nieto Oriol, Librosa: audio and music signal analysis in python, in: Proceedings of the 14th Python in Science Conference, 2015, pp. 18–25.
[27] D.P.W. Ellis, Gammatone-Like Spectrogram, 2009.
[28] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint, arXiv:1409.1556, 2014.
[29] Lam Pham, Ian McLoughlin, Huy Phan, Ramaswamy Palaniappan, Yue Lang, Bag-of-features models based on C-DNN network for acoustic scene classification, in: Proc. AES, 2019.
[30] Leo Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[31] Huy Phan, Lars Hertel, Marco Maass, Philipp Koch, Alfred Mertins, Label tree embeddings for acoustic scene classification, in: Proc. ACM, 2016, pp. 486–490.
[32] Ekaterina Garmash, Christof Monz, Ensemble learning for multi-source neural machine translation, in: The 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1409–1418.
[33] Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, Tuomas Virtanen, DCASE 2017 challenge setup: tasks, datasets and baseline system, in: Proc. DCASE, 2017, pp. 85–92.
[34] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, A multi-device dataset for urban acoustic scene classification, in: Proc. DCASE, 2018, pp. 9–13.
[35] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, Acoustic scene classification in DCASE 2019 challenge: closed and open set classification and data mismatch setups, in: Proc. DCASE, 2019.
[36] Ian McLoughlin, Zhipeng Xie, Yan Song, Huy Phan, Ramaswamy Palaniappan, Time-frequency feature fusion for noise robust audio event classification, Circuits Syst. Signal Process. (2019).

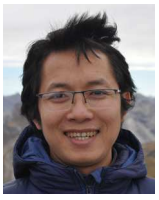
[37] Huy Phan, Phillip Koch, Ian McLoughlin, Alfred Mertins, Enabling early audio event detection with neural networks, in: Proc. ICASSP, 2018.
[38] Ian McLoughlin, Yan Song, Lam Dam Pham, Huy Pham, Palaniappan Ramaswamy, Lang Yue, Early detection of continuous and partial audio events using CNN, in: Proc. INTERSPEECH, 2018.
[39] Wei Dai, Juncheng Li, Phuong Pham, Samarjit Das, Shuhui Qu, Acoustic scene recognition with deep neural networks (DCASE challenge 2016), Tech. Rep., DCASE2016 Challenge, September 2016.
[40] Shengkui Zhao, Thi Ngoc Tho Nguyen, Woon-Seng Gan, Douglas L. Jones, ADSC submission for DCASE 2017: Acoustic scene classification using deep residual convolutional neural networks, Tech. Rep., DCASE2017 Challenge, September 2017.
[41] YangXiong Li, Xianku Li, Yuhan Zhang, The SEIE-SCUT systems for challenge on DCASE 2018: Deep learning techniques for audio representation and classification, Tech. Rep., DCASE2018 Challenge, September 2018.
[42] Mingle Liu, Wucheng Wang, Yanxiong Li, The system for acoustic scene classification using resnet, Tech. Rep., DCASE2019 Challenge, June 2019.
[43] Victor Bisot, Slim Essid, Gaël Richard, HOG and subband power distribution image features for acoustic scene classification, in: Proc. EUSIPCO, IEEE, 2015, pp. 719–723.
[44] Jung Jee-Weon, Heo Hee-Soo, Yang IL-Ho, Yoon Sung-Hyun, Shim Hye-Jin, Yu Ha-Jin, DNN-based audio scene classification for DCASE 2017: Dual inputfeatures, balancing cost, and stochastic data duplication, Tech. Rep., DCASE2017 Challenge, September 2017.
[45] Jee-weon Jung, Hee-soo Heo, Hye-jin Shim, Hajin Yu, DNN based multi-level features ensemble for acoustic scene classification, Tech. Rep., DCASE2018 Challenge, 2018.
[46] Zhitong Li, Liqiang Zhang, Shixuan Du, Wei Liu, Acoustic scene classification based on binaural deep scattering spectra with CNN and LSTM, Tech. Rep., DCASE2018 Challenge, September 2018.
[47] Yuzhong Wu, Tan Lee, Stratified time-frequency features for CNN-based acoustic scene classification, Tech. Rep., DCASE2019 Challenge, June 2019.
[48] Qiuqiang Kong, Yin Cao, Turab Iqbal, Wenwu Wang, Mark D. Plumbley, Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems, Tech. Rep., DCASE2019 Challenge, June 2019.
[49] Jiaxing Ye, Takumi Kobayashi, Masahiro Murakawa, Tetsuya Higuchi, Acoustic scene classification based on sound textures and events, in: Proc. ACM, 2015, pp. 1291–1294.
[50] Jaehun Kim, Kyogu Lee, Empirical study on ensemble method of deep neural networks for acoustic scene classification, Tech. Rep., DCASE2016 Challenge, September 2016.
[51] Karol Piczak, The details that matter: Frequency resolution of spectrograms in acoustic scene classification, Tech. Rep., DCASE2017 Challenge, September 2017.
[52] WenJie Hao, Lasheng Zhao, Qiang Zhang, HanYu Zhao, JiaHua Wang, DCASE 2018 task 1a: Acoustic scene classification by bi-LSTM-CNN-net multichannel fusion, Tech. Rep., DCASE2018 Challenge, September 2018.
[53] Juergen Tchorz, Combination of amplitude modulation spectrogram features and MFCCs for acoustic scene classification, Tech. Rep., DCASE2018 Challenge, September 2018.
[54] Wei Gao, Mark McDonnell, Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths, Tech. Rep., DCASE2019 Challenge, June 2019.
[55] Shefali Waldekar, Goutam Saha, Wavelet based mel-scaled features for DCASE 2019 task 1a and task 1b, Tech. Rep., DCASE2019 Challenge, June 2019.
[56] Gen Takahashi, Takeshi Yamada, Shoji Makino, Nobutaka Ono, Acoustic scene classification using deep neural network and frame-concatenated acoustic feature, Tech. Rep., DCASE2016 Challenge, September 2016.
[57] Ivan Kukanov, Ville Hautamski, Kong Aik Lee, Recurrent neural network and maximal figure of merit for acoustic event detection, Tech. Rep., DCASE2017 Challenge, September 2017.
[58] Christian Roletscheck, Tobias Watzka, Using an evolutionary approach to explore convolutional neural networks for acoustic scene classification, Tech. Rep., DCASE2018 Challenge, September 2018.
[59] Qiuqiang Kong, Iqbal Turab, Xu Yong, Wenwu Wang, Mark D. Plumbley, DCASE 2018 challenge surrey cross-task convolutional neural network baseline, Tech. Rep., DCASE2018 Challenge, September 2018.
[60] Mou Wang, Rui Wang, Ciaic-ASC system for DCASE 2019 challenge task1, Tech. Rep., DCASE2019 Challenge, June 2019.
[61] Zhuhe Wang, Jingkai Ma, Chunyang Li, Acoustic scene classification based on CNN system, Tech. Rep., DCASE2019 Challenge, June 2019.
[62] Yifang Yin, Rajiv Ratn Shah, Roger Zimmermann, Learning and fusing multimodal deep features for acoustic scene categorization, in: Proc. ACM, 2018, pp. 1892–1900.
[63] Benjamin Elizalde, Anurag Kumar, Ankit Shah, Rohan Badlani, Emmanuel Vincent, Bhiksha Raj, Ian Lane, Experiments on the DCASE challenge 2016: Acoustic scene classification and sound event detection in real life recording, Tech. Rep., DCASE2016 Challenge, September 2016.
[64] Sangwook Park, Seongkyu Mun, Younglo Lee, Hanseok Ko, Acoustic scene classification based on convolutional neural network using double image features, Tech. Rep., DCASE2017 Challenge, September 2017.
[65] Liwen Zhang, Jiqing Han, Acoustic scene classification using multi-layered temporal pooling based on deep convolutional neural network, Tech. Rep., DCASE2018 Challenge, September 2018.
[66] Wang Jun, Li Shengchen, Self-attention mechanism based system for DCASE 2018 challenge task1 and task4, Tech. Rep., DCASE2018 Challenge, September 2018.
[67] Jee-weon Jung, Hee-Soo Heo, Hye-jin Shim, Ha-Jin Yu, Knowledge distillation with specialist models in acoustic scene classification, Tech. Rep., DCASE2019 Challenge, June 2019.
[68] Shengwang Jiang, Chuang Shi, Acoustic scene classification using ensembles of convolutional neural networks and spectrogram decompositions, Tech. Rep., DCASE2019 Challenge, June 2019.
[69] Michele Valenti, Aleksandr Diment, Giambattista Parascandolo, Stefano Squartini, Tuomas Virtanen, DCASE 2016 acoustic scene classification using convolutional neural networks, Tech. Rep., DCASE2016 Challenge, September 2016.
[70] Bernhard Lehner, Hamid Eghbal-Zadeh, Matthias Dorfer, Filip Korzeniowski, Khaled Koutini, Gerhard Widmer, Classifying short acoustic scenes with I-vectors and CNNs: Challenges and optimisations for the 2017 DCASE ASC task, Tech. Rep., DCASE2017 Challenge, September 2017.
[71] Shefali Waldekar, Goutam Saha, Wavelet-based audio features for acoustic scene classification, Tech. Rep., DCASE 2018 Challenge, 2018.
[72] Jonathan Huang, Paulo Lopez Meyer, Hong Lu, Hector Cordourier Maruri, Juan Del Hoyo, Acoustic scene classification using deep learning-based ensemble averaging, Tech. Rep., DCASE2019 Challenge, June 2019.
[73] Hongwei Song, Hao Yang, Feature enhancement for robust acoustic scene classification with device mismatch, Tech. Rep., DCASE2019 Challenge, June 2019.
[74] Jiaxing Ye, Takumi Kobayashi, Nobuyuki Toyama, Hiroshi Tsuda, Masahiro Murakawa, Acoustic scene classification using efficient summary statistics and multiple spectro-temporal descriptor fusion, Appl. Sci. 8 (8) (2018) 1363.
[75] Rakib Hyder, Shabnam Ghaffarzadegan, Zhe Feng, Taufiq Hasan, BUET bosch consortium (B2C) acoustic scene classification systems for DCASE 2017, Tech. Rep., DCASE2017 Challenge, September 2017.
[76] An Dang, Toan Vu, Jia-Ching Wang, Acoustic scene classification using ensemble of convnets, Tech. Rep., DCASE2018 Challenge, September 2018.
[77] Yang Haocong, Shi Chuang, Li Huiyong, Acoustic scene classification using CNN ensembles and primary ambient extraction, Tech. Rep., DCASE2019 Challenge, June 2019.
[78] Paul Primus, David Eitelsebner, Acoustic scene classification with mismatched recording devices, Tech. Rep., DCASE2019 Challenge, June 2019.
[79] Huy Phan, Philipp Koch, Fabrice Katzberg, Marco Maass, Radoslaw Mazur, Alfred Mertins, Audio scene classification with deep recurrent neural networks, in: Proc. INTERSPEECH, 2017, pp. 3845–3849.
[80] Zheng Weiping, Yi Jiantao, Xing Xiaotao, Liu Xiangtao, Peng Shaohu, Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion, Tech. Rep., DCASE2017 Challenge, September 2017.
[81] Octave Mariotti, Matthieu Cord, Olivier Schwander, Exploring deep vision models for acoustic scene classification, in: Proc. DCASE, 2018, pp. 103–107.
[82] Seo Hyeji, Park Jihwan, Acoustic scene classification using various preprocessed features and convolutional neural networks, Tech. Rep., DCASE2019 Challenge, June 2019.
[83] Hamid Eghbal-zadeh, Khaled Koutini, Gerhard Widmer, Acoustic scene classification and audio tagging with receptive-field-regularized CNNs, Tech. Rep., DCASE2019 Challenge, June 2019.
[84] Teng Zhang, Kailai Zhang, Ji Wu, Data independent sequence augmentation method for acoustic scene classification, in: Proc. INTERSPEECH, 2018, pp. 3289–3293.
[85] Victor Bisot, Romain Serizel, Slim Essid, Gaël Richard, Supervised nonnegative matrix factorization for acoustic scene classification, Tech. Rep., DCASE2016 Challenge, September 2016.
[86] Yoonchang Han, Jeongsoo Park, Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification, Tech. Rep., DCASE2017 Challenge, September 2017.
[87] Liping Yang, Xinxing Chen, Lianjie Tao, Acoustic scene classification using multi-scale features, in: Proc. DCASE, 2018, pp. 29–33.
[88] Teng Zhang, Kailai Zhang, Ji Wu, Temporal transformer networks for acoustic scene classification, in: Proc. INTERSPEECH, 2018, pp. 1349–1353.
[89] Hamid Eghbal-Zadeh, Bernhard Lehner, Matthias Dorfer, Gerhard Widmer, CP-JKU submissions for DCASE-2016: a hybrid approach using binaural I-vectors and deep convolutional neural networks, Tech. Rep., DCASE2016 Challenge, September 2016.
[90] Seongkyu Mun, Sangwook Park, David Han, Hanseok Ko, Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane, Tech. Rep., DCASE2017 Challenge, September 2017.
[91] Alexander Golubkov, Alexander Lavrentyev, Acoustic scene classification using convolutional neural networks and different channels representations and its fusion, Tech. Rep., DCASE2018 Challenge, September 2018.
[92] Hangting Chen, Zuozhen Liu, Zongming Liu, Pengyuan Zhang, Yonghong Yan, Integrating the data augmentation scheme with various classifiers for acoustic scene modeling, Tech. Rep., DCASE2019 Challenge, June 2019.
[93] Michał Kośmider, Calibrating neural networks for secondary recording devices, Tech. Rep., DCASE2019 Challenge, June 2019.

[94] Seongkyu Mun, Suwon Shon, Wooil Kim, David K. Han, Hanseok Ko, Deep neural network based learning and transferring mid-level audio features for acoustic scene classification, in: Proc. ICASSP, 2017, pp. 796–800.
[95] Xue Bai, Jun Du, Zi-Rui Wang, Chin-Hui Lee, A hybrid approach to acoustic scene classification based on universal acoustic models, in: Proc. INTERSPEECH, 2019, pp. 3619–3623.
[96] Juncheng Li, Wei Dai, Florian Metze, Shuhui Qu, Samarjit Das, A comparison of deep learning methods for environmental sound detection, in: Proc. ICASSP, 2017, pp. 126–130.
[97] Yuhong Yang, Huiyu Zhang, Weiping Tu, Haojun Ai, Linjun Cai, Ruimin Hu, Fei Xiang, Kullback–Leibler divergence frequency warping scale for acoustic scene classification using convolutional neural network, in: Proc. ICASSP, 2019, pp. 840–844.
[98] Liang Gao, Haibo Mi, Boqing Zhu, Dawei Feng, Yicong Li, Yuxing Peng, An adversarial feature distillation method for audio classification, IEEE Access 7 (2019) 105319–105330.
[99] Truc Nguyen, Franz Pernkopf, Acoustic scene classification with mismatched devices using cliquenets and mixup data augmentation, in: Proc. INTERSPEECH, 2019, pp. 2330–2334.
[100] Rakib Hyder, Shabnam Ghaffarzadegan, Zhe Feng, John H.L. Hansen, Taufiq Hasan, Acoustic scene classification using a CNN-supervector system trained with auditory and spectrogram image features, in: Proc. INTERSPEECH, 2017, pp. 3073–3077.
[101] Shefali Waldekar, Goutam Saha, Wavelet transform based mel-scaled features for acoustic scene classification, in: Proc. INTERSPEECH, 2018, pp. 3323–3327.
[102] Truc Nguyen, Franz Pernkop, Acoustic scene classification with mismatched recording devices using mixture of experts layer, in: Proc. ICME, 2019, pp. 1666–1671.
[103] Yuzhong Wu, Tan Lee, Enhancing sound texture in cnn-based acoustic scene classification, in: Proc. ICASSP, 2019, pp. 815–819.
[104] Hangting Chen, Pengyuan Zhang, Yonghong Yan, An audio scene classification framework with embedded filters and a dct-based temporal module, in: Proc. ICASSP, 2019, pp. 835–839.
[105] Hee-Soo Heo, Jee-weon Jung, Hye-jin Shim, Ha-Jin Yu, Acoustic scene classification using teacher-student learning with soft-labels, arXiv preprint, arXiv: 1904.10135, 2019.

Lam Pham received the Bachelor of Engineering, and Master of Science degree in Electronics-Telecommunication Engineering from Ho Chi Minh City University of Technology in 2009 and 2012, respectively. Currently he is being PhD in University of Kent, UK and also working as research assistant in University of Surrey, UK. His research interests include machine learning and signal processing with a research focus on Machine Hearing.

Professor Alfred Mertins received his Dipl.-Ing. degree from the University of Paderborn, Germany, in 1984, the Dr.-Ing. degree in Electrical Engineering and the Dr.-Ing. habil. degree in Telecommunications from the Hamburg University of Technology, Germany, in 1991 and 1994, respectively. From 1986 to 1991 he was a Research Assistant at the Hamburg University of Technology, Germany, and from 1991 to 1995 he was a Senior Scientist at the Microelectronics Applications Center Hamburg, Germany. From 1996 to 1997 he was with the University of Kiel, Germany, and from 1997 to 1998 with the University of Western Australia. In 1998, he joined the University of Wollongong, where he was at last an Associate Professor of Electrical Engineering. From 2003 to 2006, he was a Professor in the Faculty of Mathematics and Science at the University of Oldenburg, Germany. In November 2006, he joined the University of Lübeck, Germany, where he is a Professor and Director of the Institute for Signal Processing. His research interests include speech, audio, and image processing, wavelets and filter banks, pattern recognition, and medical imaging.

Truc, Thi Kim Nguyen received her M.Sc. degree in electrical engineering from University of Ulsan, South Korea in 2013. She has been a Ph.D. candidate in the department of Electrical and Information Technology at Graz University of Technology since 2016. Her research interests include machine learning for applications of image and sound signal processing such as video-based fire and smoke detection, acoustic scene classification and lung sound classification.



Huy Phan received the M.Eng. degree from Nanyang Technological University, Singapore, in 2012, and the Dr.-Ing. degree in computer science from the University of Lübeck, Lübeck, Germany, in 2017. From 2017 to 2018, he was a Postdoctoral Research Assistant with the University of Oxford, Oxford, United Kingdom. From 2018 to 2020, he was a Lecturer at the University of Kent, Kent, United Kingdom. In April 2020, he joined Queen Mary University of London, London, United Kingdom, where he is a Lecturer in Artificial Intelligence in the School of Electronic Engineering and Computer Science. His research interests include machine learning and signal processing with a special focus on audio and biosignal analysis. In 2018, he was awarded the Bernd Fischer award by the University of Lübeck for the best PhD thesis.

Ramaswamy Palaniappan is currently a Reader in the School of Computing, University of Kent and heads the Data Science Research Group. His current research interests include signal processing and machine learning for electrophysiological applications. To date, he has written two text books in engineering and published over 200 peer-reviewed articles (with over 3000 citations). He serves in editorial boards for several international journals. He also serves in the prestigious Peer Review College for UK Research Councils and many other international grant funding bodies. He has supervised more than half a dozen postgraduate students to completion and has more than two decades of multi-disciplinary teaching experience in computer science and engineering (electrical and biomedical) disciplines. His pioneering work on revolutionary new areas of brain-computer interfaces and emerging biometrics has not only received international awards and recognition by the scientific community but also from the media and public.

Professor Ian McLoughlin completed his PhD in Electronic and Electrical Engineering at the University of Birmingham, UK in 1997. He has worked for over 10 years in the R&D industry and 19 years in academia, on three continents. He is a Fellow of the IET, a Chartered Engineer and has been a full Professor since 2012. He has written many papers and hand holds several patents in this domain, and is the author of the Cambridge University Press reference text on speech and audio processing.