

# Deep Feature Embedding and Hierarchical Classification for Audio Scene Classification

Lam Pham	Ian McLoughlin	Huy Phan	R. Palaniappan	Alfred Mertins
<i>School of Computing</i>	<i>School of Computing</i>	<i>School of Computing</i>	<i>School of Computing</i>	<i>Institute for Signal Processing</i>
<i>University of Kent</i>	<i>University of Kent</i>	<i>University of Kent</i>	<i>University of Kent</i>	<i>University of Lübeck</i>
Kent, UK	Kent, UK	Kent, UK	Kent, UK	Lübeck, Germany
ldp7@kent.ac.uk	ivm@lintech.org	h.phan@kent.ac.uk	r.palani@kent.ac.uk	alfred.mertins@uni-luebeck.de

**Abstract**—In this work, we propose an approach that features deep feature embedding learning and hierarchical classification with triplet loss function for Acoustic Scene Classification (ASC). In the one hand, a deep convolutional neural network is firstly trained to learn a feature embedding from scene audio signals. Via the trained convolutional neural network, the learned embedding embeds an input into the embedding feature space and transforms it into a high-level feature vector for representation. In the other hand, in order to exploit the structure of the scene categories, the original scene classification problem is structured into a hierarchy where similar categories are grouped into meta-categories. Then, hierarchical classification is accomplished using deep neural network classifiers associated with triplet loss function. Our experiments show that the proposed system achieves good performance on both the DCASE 2018 Task 1A and 1B datasets, resulting in accuracy gains of 15.6% and 16.6% absolute over the DCASE 2018 baseline on Task 1A and 1B, respectively.

**Index Terms**—Acoustic scene classification, spectrogram, log-Mel, Gammatone filter, constant Q transform.

## I. INTRODUCTION

In acoustic scenes, various associated and sporadic event sounds tend to occur within a typical recording. We refer to those as foreground sounds, in contrast to background, which is the more constant sound corresponding to that scene. Acoustic scene classification (ASC) is complicated by the presence of foreground sounds and by interfering noise, and is characterised by encompassing a very wide range of spectral shapes and temporal sound patterns. To deal with these challenges, many authors who achieved competitive classification accuracy [1]–[4] on the DCASE 2018 dataset [5] proposed ensemble models that explore diverse approaches to both input features and learning models. In particular, Hossein Zeinali *et al.* [1] made use of effective combination of Constant Q Transform (CQT) and log-Mel spectrograms. Firstly, they transferred draw audio into spectrogram, extracting X-vector from these spectrograms. Then, they fed these features (both two spectrograms and X-vectors extracted) into one/two-dimensional CNN models. Eventually, obtained scores were fused to produce the final classification result. Exploring nearest neighbour filter (NNF), Truc *et al.* [2] extracted NNF spectrogram from log-Mel spectrogram. Next, the authors fed four spectrograms (coming from from side, average of audio channels and two log-Mel, NNF spectrograms) into

separated CNN-based models and fuse four obtained scores. Deeply focusing on audio channels, Octave Mariotti *et al.* [3] and Yuma *et al.* [4] experimented on a wide range of input features (left, right, side and average of channels with log-Mel spectrogram and Harmonic Percussive Source Separation). Regarding ensemble models, while Yuma *et al.* [4] proposed a single CNN model similar to VGG configuration, Octave Mariotti *et al.* [3] pursued an intensive ensemble, evaluating a variety of deep learning models (VGG8, VGG10, VGG12, Resnet 18, Resnet 34, Resnet 50).

Another approach relies upon ever more powerful learning models. For example, Yang *et al.* [6] proposed a complicated CNN-based architecture called the *xception* network. This is inspired by the fact that a deep learning network trained by a wide range of feature scales and over separated channels can result in a very powerful model. Indeed, *xception* achieves the highest score for the DCASE 2018 Task 1A. Focusing on attention mechanism, an attention-based pooling layer proposed by Zhao Ren *et al.* [7] helps to improve the quality of pooling layers compared with traditional pooling layers. Exploring different frequency bands in a spectrogram, Phaye *et al.* [8] proposed a SubSpectralNet network which is useful to extract discriminative information from 30 sub-spectrograms. More recently, Hong *et al.* [9] proposed a new method that exploits distinct features in sound scenes. They firstly applied a deep learning model to extract a bag of similar and distinct features, then leverage this to enforce higher network performance. Generally, although the second trend shows complicated network architectures, almost top performances come from ensemble of CNN-based models as mentioned in the first line of methods [1]–[4], [10].

In this paper, we adopt a different approach based on deep feature embedding learning and a hierarchical classification scheme. First, feature embeddings are learned with a deep CNN in a regular classification setting. Rather than using the trained deep CNN for direct classification, it is employed as a feature extractor to embed an audio input into a high-level feature space via the learned embedding. Afterwards, the original “flat”ASC task, i.e. classification of all categories at once, is structured into multiple hierarchical sub-tasks in a divide-and-conquer manner. In the one hand, the hierarchy is constructed bottom-up. Starting from the original scene

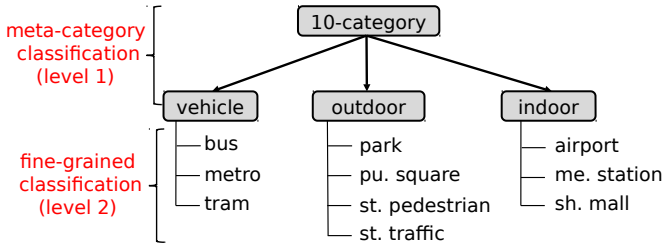


Fig. 1. The two-level hierarchy of scene categories constructed based on the categories of the DCASE 2018 datasets.

categories at the bottom, those categories, that are expected to be acoustically similar, are grouped into a meta-category as demonstrated in Figure 1. The meta-categories, therefore, constitutes the first level of the classification hierarchy. In the other hand, the classification is performed top-down, i.e. classification of the meta-categories is carried out first before classification of categories in a meta-category takes place. The classifiers in the classification hierarchy are realized by deep neural networks (DNNs). Triplet loss function, which was shown to increase Fisher’s criterion, is used to trained the DNN classifiers.

## II. THE PROPOSED SYSTEM

### A. Learning Feature Embeddings

The processing pipeline for deep feature embedding learning using a deep CNN is illustrated in Fig. 2. Each acoustic scene signal is firstly transformed into time-frequency image, such as Gammatone spectrogram with 128 Gammatone filters [11]. The time-frequency image is then decomposed into non-overlapping image patches of size  $128 \times 128$ . Let  $\mathbf{X}$  and  $\mathbf{y}$  denote an image patch and its one-hot encoding label, respectively. Mixup data augmentation [12]–[14] is then applied on the image patches to generate mixup data:

$$\mathbf{X}_{\text{mp1}} = \alpha \mathbf{X}_1 + (1 - \alpha) \mathbf{X}_2, \quad (1)$$

$$\mathbf{X}_{\text{mp2}} = (1 - \alpha) \mathbf{X}_1 + \alpha \mathbf{X}_2, \quad (2)$$

$$\mathbf{y}_{\text{mp1}} = \alpha \mathbf{y}_1 + (1 - \alpha) \mathbf{y}_2, \quad (3)$$

$$\mathbf{y}_{\text{mp2}} = (1 - \alpha) \mathbf{y}_1 + \alpha \mathbf{y}_2. \quad (4)$$

In above equations,  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are two image patches randomly selected from the set of original image patches with their labels  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , respectively.  $\mathbf{X}_{\text{mp1}}$  and  $\mathbf{X}_{\text{mp2}}$  are two mixup image patches resulted by mixing  $\mathbf{X}_1$  and  $\mathbf{X}_2$  with a random mixing coefficient  $\alpha$ .  $\alpha$  is drawn from both uniform distribution and beta distribution. Note that the labels  $\mathbf{y}_{\text{mp1}}$  and  $\mathbf{y}_{\text{mp2}}$  of the two mixup patches are no longer one-hot labels.

The resulting mixup data is used to train a network for feature embedding learning. To this end, we propose a deep CNN similar to the VGG network [15]. The network architecture and parameters are described in Table I, comprising Batch Normalization (Bn), Convolutional layers (Cv), Rectified Linear layers (Relu), Average Pooling layers (Ap), Drop-out (Dr) and Fully-Connected Layers (Fl).

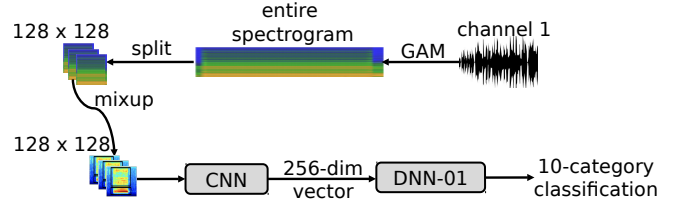


Fig. 2. Illustration of the processing pipeline to train the CNN for deep feature embedding learning.

TABLE I  
THE CNN ARCHITECTURE FOR DEEP FEATURE EMBEDDING LEARNING.

Layer	Output
Bn - Cv (9×9) - Relu - Bn - Ap (2×2) - Dr (0.1%)	64×64×32
Bn - Cv (7×7) - Relu - Bn - Ap (2×2) - Dr (0.1%)	32×32×64
Bn - Cv (5×5) - Relu - Bn - Dr (0.2%)	32×32×128
Bn - Cv (5×5) - Relu - Bn - Ap (2×2) - Dr (0.2%)	16×16×128
Bn - Cv (3×3) - Relu - Bn - Dr (0.2%)	16×16×256
Bn - Cv (3×3) - Relu - Bn - Ap (2×2) - Dr (0.2%)	8×8×256
Bn - Cv (8×8) - Relu - Bn - Dr (0.2%)	256
Fl - Dr (0.3%)	512
Fl - Dr (0.3%)	1024
Fl - Dr (0.3%)	10

For clarity, in Fig. 2 and Table I, we intentionally separate the deep CNN into two parts: the CNN part for feature learning and the DNN part for classification (denoted as **DNN-01** to distinguish it from those DNNs in Section I). Particularly, instead of using a Global Average Pooling layer at the end of the CNN as other authors do [4], [16], [17], we design an additional convolutional layer with the kernel size of  $[8 \times 8]$ , that equals to the time-frequency resolution of the output of the previous layer, to capture the interaction across the convolutional channel dimension. Since the labels of the mixup data input are no longer one-hot, we trained the network with Kullback-Leibler (KL) divergence loss rather than the standard cross-entropy loss over all  $N$  mixup training image patches:

$$E_{KL}(\Theta) = \sum_{n=1}^N \mathbf{y}_n \log\left(\frac{\mathbf{y}_n}{\hat{\mathbf{y}}_n}\right) + \frac{\lambda}{2} \|\Theta\|_2^2, \quad (5)$$

where  $\Theta$  denotes the trainable network parameters and  $\lambda$  denote the  $\ell_2$ -norm regularization coefficient.  $\mathbf{y}_c$  and  $\hat{\mathbf{y}}_c$  denote the ground-truth and the network output, respectively.

Once the network has been trained, the feature-learning CNN part of the network is used as a feature extractor and its last convolutional layer is considered as the deep feature embedding. Presented with a new input, the feature extractor will process the input starting from the first convolutional layer to the embedding layer and produce a high-level feature vector of size 256.

### B. Two-level Hierarchical Classification

Most of exiting works follow a “flat” classification scheme in which all the scenes categories at classified at once. Differently, we propose to perform the classification hierarchically. The set of scene categories are grouped to form meta-categories. Each meta-category consists of scene categories which are expected to be acoustically similar. In this sense, we

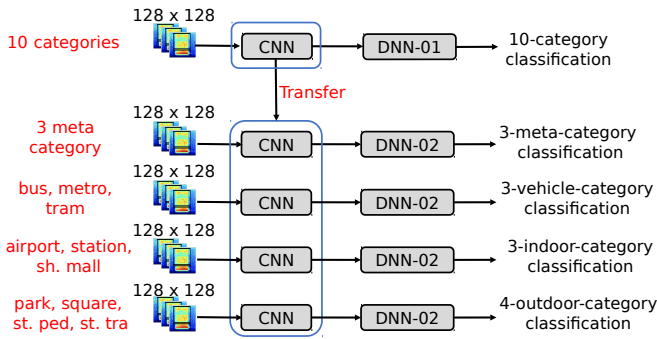


Fig. 3. Illustration of extracting high-level features from the learned feature embedding to train the DNN classifiers in the hierarchical classification scheme.

TABLE II  
DNN-02’S ARCHITECTURE.

Layer	Output Shape
Input layer	256
FI - Dr (0.3%)	512
FI - Dr (0.3%)	1024
FI - Dr (0.3%)	1024
FI - Dr (0.3%)	10

construct a two-level hierarchy based on the scene categories in the experimental DCASE 2018 datasets, as shown in Fig. 3. Three meta-categories are formed from 10 scene categories of the DCASE 2018 datasets, including “vehicle”, “indoor”, and “outdoor”. The hierarchical classification is performed in top-down fashion. The meta-categories are classified first, followed by the fine-grained classification of the scene categories in each individual meta-category. As a result, four classifiers are learned: one for meta-category classification (namely meta-category classifier) and three for classification of categories in three meta-categories (namely “vehicle” classifier, “indoor” classifier, and “outdoor” classifier, respectively). An unseen example will be then correctly classified if it is correctly classified by the classifiers at both levels. For example, a “bus” scene example is correctly classified if it is both correctly classified as “vehicle” by the meta-category classifier and as “bus” by the “vehicle” classifier. A misclassification by one of the classifiers will result in the example is wrongly classified.

The classifiers involving in the hierarchical classification are realized by DNNs, denoted as **DNN-02s**. Via the learned embedding presented in Section II, 256-dimensional high-level feature vectors are obtained for the mixup image patches and used to train the **DNN-02s**. In doing this, we effectively transfer the CNN part of the trained CNN in Section II, freeze its parameters, and use it as a feature extractor before presenting the extracted features to a **DNN-02**, as illustrated in Fig. 3. Note that the **DNN-02s** share a common architecture but are trained separately depending on the sub-tasks in the hierarchical classification. Each **DNN-02** comprises four fully-connected layers and parametrized as in Table II.

In addition to the KL-divergence loss, we additionally employ triplet loss function [18] to train the **DNN-02s** to encourage the networks to improve its discrimination power.

Triplet loss function has been shown to be efficient to learn a metric to minimize same-category distances and maximize between-category distances simultaneously, and hence, enhance the Fisher’s criterion. Supposed that we present two samples of different categories to a **DNN-02**, and denote the ground-truth of the first sample as the anchor  $\mathbf{a}$ , the prediction for the first sample as positive  $\mathbf{p}$ , and the prediction for the second sample as positive  $\mathbf{n}$ , the triplet loss is given as

$$E_{\text{triplet}} = \max(d(\mathbf{a}, \mathbf{p}) - d(\mathbf{a}, \mathbf{n}) + \text{margin}, 0), \quad (6)$$

where  $d$  is squared Euclidean distance and the *margin* is set to 0.3.

The final loss function is, therefore, a combination of the KL-divergence loss and the triplet loss:

$$E(\Theta) = \gamma E_{KL}(\Theta) + (1 - \gamma) E_{\text{triplet}}(\Theta), \quad (7)$$

where  $E_{KL}$  is the KL-divergence loss given in (5).

### C. Ensemble with Multiple Time-Frequency Inputs

Using multiple input types has been a rule of thumb in ASC [19], [20]. We, therefore, propose to use three different time-frequency inputs, including log-Mel [21], Gammatone filter (GAM) [11], and Constant Q Transform (CQT) [21], to form an ensemble of three systems. The final decision of each classification task (meta-category classification at the level 1 or fine-grained classifications at the level 2 shown in Figure 1) is obtained by aggregating the individual decisions of the three classifiers (each with one type of spectrogram) in the ensemble and the final classification label is determined via maximum posterior probability:

$$\hat{y} = \text{argmax}(\bar{\mathbf{p}}_{\text{log-Mel}} + \bar{\mathbf{p}}_{\text{GAM}} + \bar{\mathbf{p}}_{\text{CQT}}), \quad (8)$$

where  $\bar{\mathbf{p}}$  denotes the posterior probability output of a classification model and  $\hat{y}$  denotes the final label.

## III. EXPERIMENTS

### A. DCASE 2018 Datasets

Our experiments were based on the DCASE 2018 Task 1A and 1B development datasets [5]. The audio signals in Task 1A was recorded at a sample rate of 44.1 kHz by only one device

TABLE III  
THE NUMBER OF SCENE RECORDINGS CORRESPONDING TO EACH SCENE CATEGORIES IN THE TRAINING SET (TRAIN. SET) AND EVALUATION SET (EVAL. SET) OF THE DCASE 2018 TASK 1A & 1B DEVELOPMENT DATASETS [5].

Category	Task 1A	Task 1A	Task 1B	Task 1B
	Train. set	Eval. set	Train. set	Eval. set
Airport	599	265	707	301
Bus	622	242	730	278
Metro	603	261	711	297
Metro Station	605	259	713	295
Park	622	243	730	278
Public Square	648	216	756	252
Shopping Mall	585	279	693	315
Street Pedestrian	617	247	725	283
Street Traffic	618	246	726	282
Tram	603	261	711	297

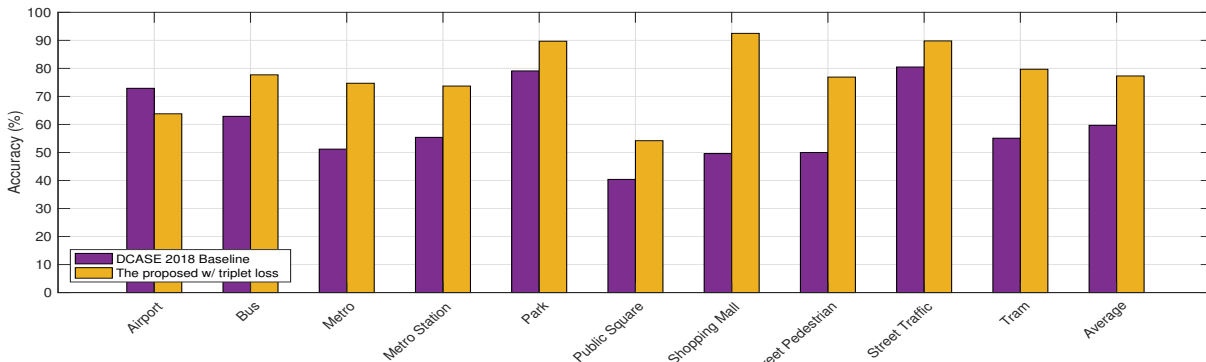


Fig. 4. Category-wise performance comparison between the proposed system with triplet loss and the DCASE 2018 baseline on **Task 1A**.

TABLE IV  
PERFORMANCE COMPARISON BETWEEN THE PROPOSED SYSTEMS, THE DCASE 2018 BASELINE, AND THE DEVELOPED BASELINE.

System	Task 1A	Task 1B
DCASE 2018 baseline [5]	59.7	45.6
The developed baseline	70.9	61.1
The proposed w/o triplet loss	73.3	<b>62.2</b>
The proposed w/ triplet loss	<b>75.3</b>	58.9

(known as device A) with 10-second long for each recording. For Task 1B, all recordings using the device A from Task 1A are reused. In addition, new recordings with two different devices (device B & device C), were added (72 recordings from each device for every category). The goal of Task 1B is to evaluate the performance on the device B and C when there are mismatched devices in real-world applications. It should be noted the imbalance of Task 1B data as there was only 4 hours of data recorded with the devices B & C compared with 24 hours of data recorded with the device A. Adhering to the setting of DCASE 2018 challenge, we divided the development dataset into a training and evaluation subsets (Train. set and Eval. set) as shown in Table III.

### B. Baselines

Besides comparison with the DCASE 2018 baseline and the results reported in previous works, we used the CNN used for deep feature embedding learning in Section II-A as the developed baseline to justify the impact of the learned deep feature embedding and the hierarchical classification scheme. When being used as a classification baseline, the CNN was trained to classify 10 categories of the datasets as in typical setting.

### C. Other parameters

The time-frequency image features, i.e. Gammatone, log-Mel, and CQT spectrogram, were obtained via a short-time window size of 43 ms and hop size of 6 ms. All of them have a common number of filter of 128.

The networks were implemented using the Tensorflow framework. The coefficient  $\lambda$  in (5) was set to  $10^{-4}$ , and  $\gamma$  in (7) was experimentally set to 0.2. The network training was

accomplished with Adam optimizer [22] with the learning rate of  $10^{-4}$ , a batch size of 100, and stop after 100 epoches.

### D. Experimental Results

Performance obtained by the proposed system, the developed baseline, and the DCASE 2018 baseline are shown in Table IV. As can be seen, the proposed system outperforms all the DCASE 2018 baseline with a large margin, 15.6% absolute (with triplet loss) on Task 1A and 16.6% absolute on Task 1B (without triplet loss). Improvements on individual categories can also be seen, as shown in Fig. 4 for a comparison between the proposed system with triplet loss and the DCASE 2018 baseline on Task 1A, with several categories enjoying a significant gain of more than 20%, such as “shopping mall”, “tram”, “metro”, “street-pedestrian”.

Compared to the developed baseline, the proposed system leads to an accuracy gain of 2.4% and 1.1% on Task 1A and Task 1B, respectively, when the triplet loss is not used. When the triplet loss is used, a significant accuracy improvement is seen on Task 1A: 2.4% absolute compared to that without triplet loss and 4.4% compared to the developed baseline thanks to the proposed hierarchical classification scheme. However, using triplet loss seems to be counter-productive on Task 1B as the accuracy is reduced by 3.3% absolute in comparison to the system without triplet loss. This is presumably due to the device mismatch or the lack of training data on the target devices (device B & C) or both. However, average over all the devices, the proposed system with triplet loss outperforms all other counterparts, as shown in Fig. 5.

We further collate the results reported in previous works (both the DCASE 2018 challenge submission systems and the recent works) and provide a comprehensive performance comparison on Task 1A and Task 1B in Tables V and VI, respectively. It should be noted that there are inconsistencies between the accuracies reported in the DCASE 2018 technical reports and those published in DCASE 2018 challenge website<sup>1</sup>. The results in Tables V and VI are collated from the technical reports which are the original sources of the reported accuracies. For clarity, we only cover top 10 DCASE

<sup>1</sup><http://dcase.community/challenge2018/>

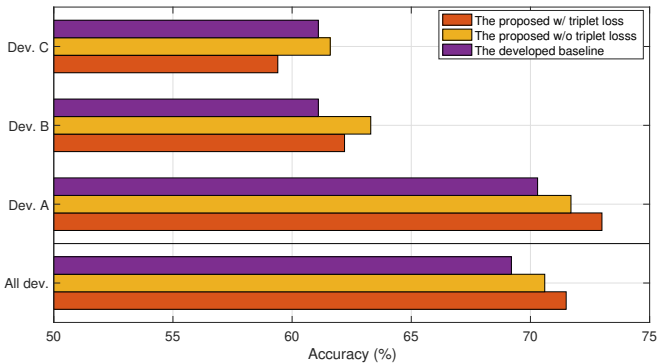


Fig. 5. Accuracy obtained by the systems developed in this work on different devices of **Task 1B**.

2018 challenge submissions in the tables. In the one hand, the proposed system outperforms the recent works (i.e. after the DCASE 2018 challenge) on Task 1A while retaining as top-3 performer in the context of the DCASE 2018 submission systems. In the other hand, our proposed system achieves state-of-the-art results on Task 1B, achieving an accuracy of 66.9% and outperforming both the DCASE 2018 submission systems and the previous works.

#### E. Discussion

To shed light on the performance of the classifiers in the proposed hierarchical classification scheme, we shown their confusion matrices in Fig. 6. Overall, the meta-categories can be discriminated very well with an average accuracy of 94% achieved by the meta-category classifier. Given the good performance of the meta-category classifier, the test examples are expected to be directed to the correct groups in the lower level. Even though the fine-grained classifiers’ performance are not as good as that of the meta-category classifier since the categories in a group tend to be similar acoustically, they are expected to perform better than the case of “flat” classification with 10 classes at once. The reason is, in one group, the classification subtask is able to avoid the confusion between its categories and those in other groups.

Overall, out of the individual time-frequency inputs (i.e. Gammatone spectrogram, log-Mel spectrogram, and CQT spectrogram), Gammatone spectrogram seems to perform best as shown in Fig. 7 while CQT spectrogram is the worst. However, aggregation the classification outputs of all three results in significant improvements over the individual ones. This is observed over all systems, the proposed system with triplet loss, the proposed system without triplet loss, and the developed baseline. It is expected as different time-frequency representations have been shown to be good for different scene categories, and their individual strength is leveraged in the ensemble to bring up performance gain.

#### IV. CONCLUSION

We have presented an approach that learns deep feature embedding to extract high-level features for audio scene signals via a deep CNN and proposed a novel hierarchical

TABLE V  
COMPARISON BETWEEN DCASE2018 BASELINE, THE TOP-10 DCASE 2018 CHALLENGE (TOP), RECENT PAPERS (MIDDLE), AND THE PROPOSED SYSTEM (BOTTOM) ON **Task 1A**.

System	Method	Acc. (%)
DCASE2018 Baseline [23]	CNN	59.7
Li [24]	DNN-biLSTM	72.9
Jung [25]	Ens. of CNN-SVM	73.5
Hao [26]	Ens. of biLSTM-CNN	73.6
Christian [27]	CNN-Voting	74.7
Zhang [28]	CNN-SVM	75.3
Li [29]	Ens. of CNN, DNN	76.6
Dang [30]	Ens. of CNNs	76.7
Yuma [4]	Ens. of CNNs	76.9
Octave [3]	Ens. of CNNs	79.3
Yang [6]	Xception CNN	<b>79.8</b>
Bai [31]	Hybrid-DNN	66.1
Zhao [32]	CNN	72.6
Phaye [8]	SubSpectralNet CNN	74.1
Zeinali [1]	Ens. of CNNs	77.5
<b>The proposed w/ triplet loss</b>	Ens. of hier. DNNs	<b>78.0</b>

TABLE VI  
COMPARISON BETWEEN THE DCASE 2018 BASELINE, THE TOP-7 DCASE 2018 CHALLENGE (TOP), THE RECENT PAPERS (MIDDLE), AND THE PROPOSED SYSTEM (BOTTOM) ON **Task 1B** (ONLY DEVICES B & C).

System	Method	Acc. (%)
DCASE2018 Baseline [23]	CNN	45.6
Li [33]	Ens. of CNN, DNN	51.7
Tchorz [34]	LSTM	53.9
Kong [35]	CNN	57.5
Wang [36]	Self-attention CNN	57.5
Waldekar [37]	Ens. of CNNs	57.8
Zhao [38]	CNN	58.3
Truc [2]	Ens. of CNNs	63.6
Zhao [32]	CNN	63.3
Truc [39]	CNN, Mix. of Experts	64.7
Yang [40]	Xception CNN	65.1
Truc [10]	Ens. of CNNs	66.1
<b>The proposed w/o triplet loss</b>	Ens. of hier. DNNs	<b>66.9</b>

classification scheme to accomplish the scene classification. In the classification hierarchy, the similar scene categories are grouped into meta-categories. Meta-category classification was carried out first, followed by the fine-grained classification in the groups. DNNs were trained with triplet loss to play the role of the classifiers in the classification hierarchy. Experiments on the DCASE 2018 Task 1A and 1B datasets demonstrated that the proposed methods significantly outperformed the DCASE 2018 baseline while achieving highly competitive results compared to state-of-the-art systems. In future work, it is worth further experimenting with deeper-level hierarchical schemes with large number of categories as well as with data-driven clustering approaches.

#### REFERENCES

- [1] Hossein Zeinali, Lukas Burget, and Jan Cernocky, “Convolutional neural networks and x-vector embedding for dcase2018 acoustic scene classification challenge,” in *Proc. DCASE*, 2018, pp. 202–206.
- [2] Truc Nguyen and Franz Pernkopf, “Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters,” in *Proc. DCASE*, 2018, pp. 34–38.
- [3] Octave Mariotti, Matthieu Cord, and Olivier Schwander, “Exploring deep vision models for acoustic scene classification,” in *Proc. DCASE*, 2018, pp. 103–107.

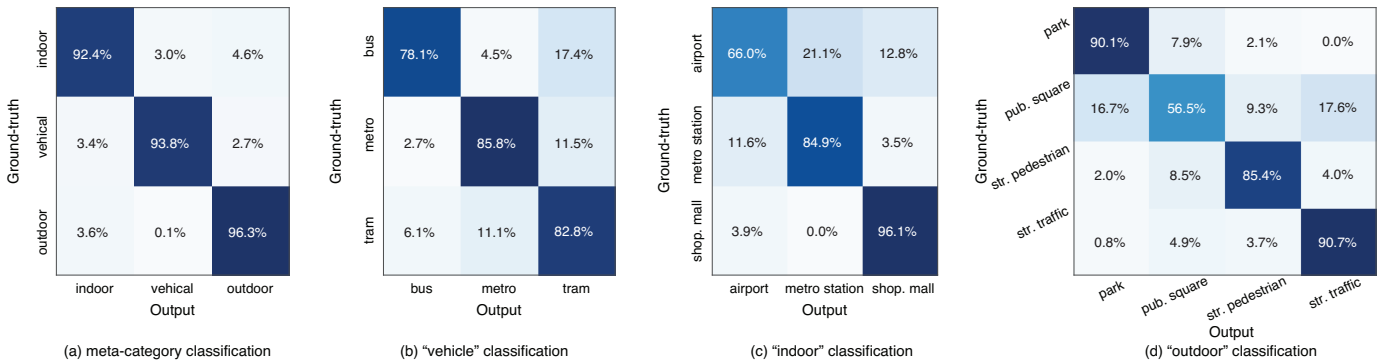


Fig. 6. Confusion matrices obtained by different classifiers in the proposed hierarchical classification scheme on **Task 1A**.

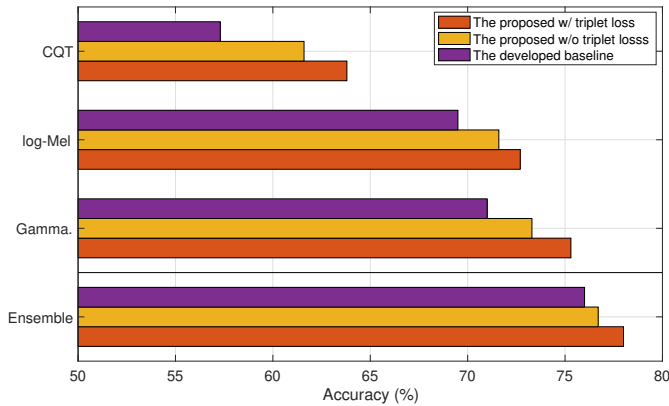


Fig. 7. Performance of individual time-frequency representations and their ensemble on **Task 1A**.

[4] Yuma Sakashita and Masaki Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," Tech. Rep., DCASE2018 Challenge, September 2018.

[5] Annamaria Mesaros, Toni Heittola, and Tumas Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. DCASE*, 2018, pp. 9–13.

[6] Liping Yang, Xinxing Chen, and Lianjie Tao, "Acoustic scene classification using multi-scale features," in *Proc. DCASE*, 2018, pp. 29–33.

[7] Ren Zhao, Kong Qiuqiang, Qian Kun, D.Plumbly Mark, and W.Schullerl Bjorn, "Attention-based convolutional neural networks for acoustic scene classification," in *Proc. DCASE*, 2018, pp. 39–43.

[8] Sai Phaye, Emmanouil Benetos, and Ye Wang, "Subspectralnet – using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *Proc. ICASSP*, 2019, pp. 825–829.

[9] Hongwei Song, Jiqing Han, Shiwen Deng, and Zhihao Du, "Acoustic scene classification by implicitly identifying distinct sound events," *Proc. INTERSPEECH*, pp. 3860–3864, 2019.

[10] Truc Nguyen and Franz Pernkop, "Acoustic scene classification with mismatched recording devices using mixture of experts layer," *Proc. ICME*, pp. 1666–1671, 2019.

[11] D P W (2009) Ellis, "Gammatone-like spectrogram," <http://www.ee.columbia.edu/~dpwe/resources/matlab/gammatonegram>.

[12] Kele Xu, Dawei Feng, Haibo Mi, Boqing Zhu, Dezhi Wang, Lilun Zhang, Hengxing Cai, and Shuwen Liu, "Mixup-based acoustic scene classification using multi-channel convolutional neural network," in *Pacific Rim Conference on Multimedia*, 2018, pp. 14–23.

[13] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada, "Learning from between-class examples for deep sound recognition," *arXiv preprint arXiv:1711.10282*, 2017.

[14] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[15] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] Truc Nguyen and Franz Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in *Proc. DCASE*, 2018, pp. 34–38.

[17] Christian Roletscheck, Tobias Watzka, Andreas Seiderer, Dominik Schiller, and Elisabeth André, "Using an evolutionary approach to explore convolutional neural networks for acoustic scene classification," in *Proc. DCASE*, 2018, pp. 158–162.

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[19] Lam Pham, Ian McLoughlin, Huy Phan, Ramaswamy Palaniappan, and Yue Lang, "Bag-of-features models based on C-DNN network for acoustic scene classification," in *Proc. AES*, 2019.

[20] Lam Pham, Ian McLoughlin, Huy Phan, and Ramaswamy Palaniappan, "A robust framework for acoustic scene classification," *Proc. INTERSPEECH*, pp. 3634–3638, 2019.

[21] McFee, Brian, Raffel Colin, Liang Dawen, Daniel. PW.Ellis, McVicar Matt, Battenberg Eric, and Nieto Oriol, "librosa: Audio and music signal analysis in python," in *Proceedings of The 14th Python in Science Conference*, 2015, pp. 18–25.

[22] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, "A multi-device dataset for urban acoustic scene classification," Tech. Rep., DCASE2018 Challenge, September 2018.

[24] YangXiong Li, Xianku Li, and Yuhuan Zhang, "The SEIE-SCUT systems for challenge on DCASE 2018: Deep learning techniques for audio representation and classification," Tech. Rep., DCASE2018 Challenge, September 2018.

[25] Jee-weon Jung, Hee-soo Heo, Hye-jin Shim, and Hajin Yu, "DNN based multi-level features ensemble for acoustic scene classification," Tech. Rep., DCASE2018 Challenge, 2018.

[26] WenJie Hao, Lasheng Zhao, Qiang Zhang, HanYu Zhao, and JiaHua Wang, "DCASE 2018 task 1a: Acoustic scene classification by bi-LSTM-CNN-net multichannel fusion," Tech. Rep., DCASE2018 Challenge, September 2018.

[27] Christian Roletscheck and Tobias Watzka, "Using an evolutionary approach to explore convolutional neural networks for acoustic scene classification," Tech. Rep., DCASE2018 Challenge, September 2018.

[28] Liwen Zhang and Jiqing Han, "Acoustic scene classification using multi-layered temporal pooling based on deep convolutional neural network," Tech. Rep., DCASE2018 Challenge, September 2018.

[29] Zhitong Li, Liqiang Zhang, Shixuan Du, and Wei Liu, "Acoustic scene classification based on binaural deep scattering spectra with CNN and LSTM," Tech. Rep., DCASE2018 Challenge, September 2018.

[30] An Dang, Toan Vu, and Jia-Ching Wang, "Acoustic scene classification using ensemble of convnets," Tech. Rep., DCASE2018 Challenge, September 2018.

[31] Xue Bai, Jun Du, Zi-Rui Wang, and Chin-Hui Lee, "A hybrid approach

- to acoustic scene classification based on universal acoustic models,” *Proc. INTERSPEECH*, pp. 3619–3623, 2019.
- [32] Z. Ren, Q. Kong, J. Han, M. D. Plumbley, and B. W. Schuller, “Attention-based atrous convolutional neural networks: Visualisation and understanding perspectives of acoustic scenes,” in *Proc. ICASSP*, May 2019, pp. 56–60.
- [33] Zhitong Li, Liqiang Zhang, Shixuan Du, and Wei Liu, “Acoustic scene classification based on binaural deep scattering spectra with CNN and LSTM,” Tech. Rep., DCASE2018 Challenge, September 2018.
- [34] Juergen Tchorz, “Combination of amplitude modulation spectrogram features and MFCCs for acoustic scene classification,” Tech. Rep., DCASE2018 Challenge, September 2018.
- [35] Qiuqiang Kong, Iqbal Turab, Xu Yong, Wenwu Wang, and Mark D. Plumbley, “DCASE 2018 challenge surrey cross-task convolutional neural network baseline,” Tech. Rep., DCASE2018 Challenge, September 2018.
- [36] Wang Jun and Li Shengchen, “Self-attention mechanism based system for dcase2018 challenge task1 and task4,” Tech. Rep., DCASE2018 Challenge, September 2018.
- [37] Shefali Waldekar and Goutam Saha, “Wavelet-based audio features for acoustic scene classification,” Tech. Rep., DCASE 2018 Challenge, 2018.
- [38] Ren Zhao, Kong Qiuqiang, Qian Kun, D.Plumbley Mark, and W.Schuller1 Bjorn, “Attention-based convolutional neural networks for acoustic scene classification,” in *Proc. DCASE*, 2018, pp. 39–43.
- [39] Truc Nguyen and Franz Pernkopf, “Acoustic scene classification with mismatched devices using cliquenets and mixup data augmentation,” *Proc. INTERSPEECH*, pp. 2330–2334, 2019.
- [40] Liping Yang, Xinxing Chen, Lianjie Tao, and Xiaohua Gu, “Multi-scale fusion and channel weighted cnn for acoustic scene classification,” in *Proceedings of the 2019 2nd International Conference on Signal Processing and Machine Learning*, 2019, pp. 41–45.