

# Acoustic Scene Classification by Implicitly Identifying Distinct Sound Events

Hongwei Song<sup>1</sup>, Jiqing Han<sup>1</sup>, Shiwen Deng<sup>2</sup>, Zhihao Du<sup>1</sup>

<sup>1</sup>School of Computer Science and Technology, Harbin Institute of Technology, China

<sup>2</sup>School of Mathematical Science, Harbin Normal University, China

{songhongwei, jqhan}@hit.edu.cn, dengswen@gmail.com, duzhihao.china@gmail.com

## Abstract

In this paper, we propose a new strategy for acoustic scene classification (ASC), namely recognizing acoustic scenes through identifying distinct sound events. This differs from existing strategies, which focus on characterizing global acoustical distributions of audio or the temporal evolution of short-term audio features, without analysis down to the level of sound events. To identify distinct sound events for each scene, we formulate ASC in a multi-instance learning (MIL) framework, where each audio recording is mapped into a bag-of-instances representation. Here, instances can be seen as high-level representations for sound events inside a scene. We also propose a MIL neural networks model, which implicitly identifies distinct instances (i.e., sound events). Furthermore, we propose two specially designed modules that model the multi-temporal scale and multi-modal natures of the sound events respectively. The experiments were conducted on the official development set of the DCASE2018 Task1 Subtask B, and our best-performing model improves over the official baseline by 9.4% (68.3% vs 58.9%) in terms of classification accuracy. This study indicates that recognizing acoustic scenes by identifying distinct sound events is effective and paves the way for future studies that combine this strategy with previous ones.

**Index Terms:** acoustic scene classification, distinct sound events identification, multi-instance learning

## 1. Introduction

Acoustic scene classification (ASC) refers to the task of categorizing real-life audio recordings into one of the *environment* classes (office, bus, etc) [1] [2]. Potential applications of ASC include context-aware devices [3] and robotic navigation.

An acoustic scene (e.g., office) consists of a stream of sound events (e.g., people-talking, pen-dropping etc), where each sound event is associated with one or more sound sources that produce it. Early works on ASC took a two-stage strategy, which recognized sound events before recognizing acoustic scenes. For instance, a hierarchical probabilistic model is presented for key audio effects (sound events) detection and auditory context (acoustic scene) inference [4]. While in [5], audio events were detected using a supervised classifier, with each audio context then represented using a histogram of sound events.

The two-stage strategy relies on a manually predefined sound events set and detailed sound event annotations (onset and offset), which requires huge human efforts and is only feasible for small scale datasets. Even more significant is that, real-life acoustic scene recordings usually involve a great number of overlapping sound events with ambiguous temporal boundaries, which makes annotating sound events almost impossible.

The aforementioned limitations have led the most research efforts to a one-stage strategy. For a one-stage strategy, only the audio recording level label is needed. Typically, acoustic

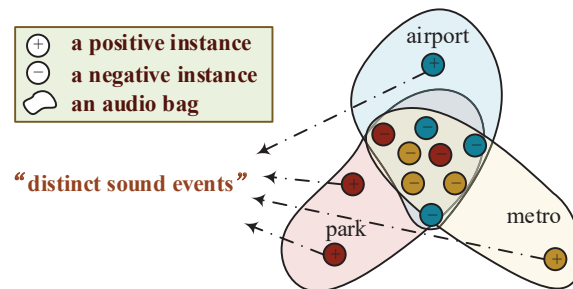


Figure 1: Illustration of the multi-instance learning (MIL) based acoustic scene classification. Each audio recording is represented as a bag-of-instances. Positive instances of a bag represent the "distinct sound events" of the scene, which are implicitly identified by the MIL framework. We assign a scene label to the audio if any distinct sound events of the scene is identified.

scenes are described by their global acoustical distributions or the temporal evolution of short-term audio features. For instance, some low-level [6] or middle-level [7] [8] [9] [10] [11] descriptors are extracted. The extraction of these descriptors is generally followed by statistical models or sequence-learning models to summarize the information and make final decisions. Some high-level feature extraction methods, which are popular in the speaker recognition field, have also been investigated and proved to be quite effective for ASC, such as the famous i-vectors [12] and x-vectors [13].

More recently, another one-stage strategy that has been widely adopted is to utilize deep learning models to map an audio recording directly to its scene label. In particular, Convolutional Neural Networks (CNN) based methods [14] [15] [16] [17] [18] have demonstrated great potential. Equally, several novel Recurrent Neural Networks (RNN) based architectures have been proposed [19] [20] to model the temporal evolution of short-term audio features. Novel data augmentation methods for the ASC have also been proposed to handle the problem of data scarcity [21] [22].

Alongside these developments, listening tests have been conducted to understand the perceptual processes of the ASC for humans. In particular, it has been observed that human recognition of soundscapes is guided by identifying prominent sound events [23]. As such it is clear that it is important to focus the recognition process on distinct sound events. This finding accords with our intuition. For instance, identifying the sound of birds singing would immediately differentiate a *park* scene from an *office* scene, which is as not easily recognized from global acoustical distributions.

Inspired by these psychological findings, in this paper we aim to integrate this strategy into designing computational algorithms for ASC. We aim to identify to what extent we can rec-

ognize acoustic scenes by identifying the distinct sound events. To avoid manually annotating sound events, we treat this problem as a weakly supervised learning problem and formulate the problem through a multi-instance learning (MIL) [24] framework. As shown in Fig. 1, the main idea of our MIL-based ASC is to map audio into a *bag-of-instances* representation and identify distinct sound events for each scene by detecting *positive* instances.

We summarize our contribution as follows: First, we propose a new strategy for ASC and formulate the problem of identifying distinct sound events for ASC in a Multi-Instance Learning (MIL) framework. Second, we propose a novel deep learning-based MIL model with specially designed modules to model the multi-temporal scale and multi-modal natures of the sound events in our daily acoustic environment.

## 2. Proposed methods

### 2.1. Multiple Instance Learning

MIL is a popular framework for solving weakly-supervised learning problems. In MIL, labels are attached to a set of *instances*, called *bag*, rather than to individual *instances* within the *bag*. MIL has been widely applied, to areas such as audio event detection (AED) [25] [26] [27] and bird sound classification [28]. An attention-based CNN model [18] has also been used for ASC and has provided interpretations in a MIL framework. However, the interpretation is under the *embedding space* [24] paradigm of the MIL, which does not have the ability to identify distinct sound events as our methods do. For more details of MIL and its applications, please see [24].

### 2.2. Formulations and notations

For each audio recording, a log mel-spectrogram is extracted and denoted as  $\mathbf{X}_i$ . The *bag-of-instances* representation of the spectrogram is noted as  $\{\mathbf{x}_{ij}\}_{j=1}^{m_i}$ , where  $\mathbf{x}_{ij} \in \mathbb{R}^d$  is a high level vector representation for sound events in a segment of the audio recording.  $i$  and  $j$  is the index for the *bag* and the *instance* respectively, and  $m_i$  denotes the number of *instances* in the *bag*. The original MIL and its *standard multi-instance (SMI) hypothesis* [24] was proposed in the context of binary classification. Here for multi-class ASC, we treat each class independently, then the SMI hypothesis can be naturally described as follows:

$$\mathbf{Y}_{il} = \begin{cases} 1, & \exists j : \mathbf{y}_{ijl} = 1 \\ 0, & \forall j : \mathbf{y}_{ijl} = 0 \end{cases} \quad (1)$$

Here,  $l \in [1..C]$  is the index for each class and  $C$  is the number of classes.  $\mathbf{Y}_i \in \{0, 1\}^C$  and  $\mathbf{y}_{ij} \in \{0, 1\}^C$  are the one-hot label for the *bag* and *instances* respectively, where 1 means *positive* and 0 means *negative*. The SMI is important since it determines the relations between the *bag* and the *instances*. By adopting the SMI hypothesis, positive instances in one scene cannot appear in other scenes. Therefore, the positive instances of one scene represent the *distinct* events of the scene.

It is worth noting that for our task, instance labels  $\mathbf{y}_{ij}$  are not available for training. Moreover, in following the convention of the MIL, we consistently utilize capital letters to represent the *bag*-level symbols and lower-case letters to represent *instance*-level symbols.

### 2.3. A general framework

The MIL based ASC model can be broken into three parts. 1) An *instance generator*  $g$ , which maps an input log mel-

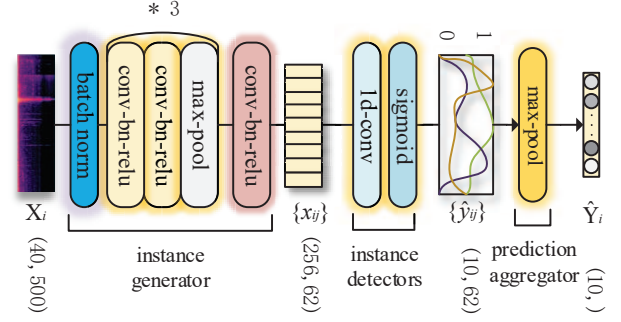


Figure 2: An overview of the CNN-MIL model.

spectrogram into instance vectors. 2) A group of *distinct instance detectors*  $\{f_l\}_{l=1}^C$ , which map each instance vector to its label prediction score  $\hat{\mathbf{y}}_{ij}$ . 3) A *prediction aggregator*  $p$ , which aggregates instance-wise predictions into a bag-level prediction  $\hat{\mathbf{Y}}_i$ . A symbolic representation of the complete MIL model is shown as follows:

$$\mathbf{X}_i \xrightarrow{g} \{\mathbf{x}_{ij}\}_{j=1}^{m_i} \xrightarrow{\{f_l\}_{l=1}^C} \{\hat{\mathbf{y}}_{ij}\}_{j=1}^{m_i} \xrightarrow{p} \hat{\mathbf{Y}}_i \quad (2)$$

### 2.4. A CNN based MIL model

Fig. 2 presents an overview of our proposed CNN-MIL model in terms of the three parts of the general framework. For the *instance generator*, a VGG-like [29] CNN module is designed to map the input log mel-spectrogram to a bag-of-instances representation. The input dimensions are ordered as (feature, time). The CNN module begins with three convolutional blocks. Each block consists of two stacked 2D convolutional layers followed by a strided (2, 2) max pooling layer, where the number of filters is doubled for each subsequent block (32, 64, 128) and all filters are of dimension (3, 3). This is followed by a single 2D convolutional layer with 256 full-height (5, 1) filters, colored pink in Fig. 2. Batch normalization layer [30] and ReLU nonlinearity is applied to the output of every convolutional layer as well as to the input of the network (i.e., after the spectrogram). Finally, the 3D tensor is reshaped into the bag-of-instances representation (256, 62). Each instance vector (with  $d = 256$ ) could be considered as a high-level representation for sound events.

For the group of *distinct instance detectors*, one independent detector  $f_l$  for each class  $l$  is applied, as shown below:

$$\hat{\mathbf{y}}_{ijl} = f_l(\mathbf{x}_{ij}) = \text{sigmoid}(\mathbf{w}_l^T \mathbf{x}_{ij} + \mathbf{b}_l), \quad (3)$$

where  $\mathbf{w}_l, \mathbf{x}_{ij} \in \mathbb{R}^d, \mathbf{b}_l \in \mathbb{R}, \forall l \in [1..C]$

Each  $f_l$  is composed of an affine transformation followed by a *sigmoid* activation function. One way to interpret the Eq. (3) is that there is one 'template'  $\mathbf{w}_l$  of distinct sound event for each class, and a large  $\hat{\mathbf{y}}_{ijl}$  would indicate that  $\mathbf{x}_{ij}$  is very likely a distinct instance (event) for the  $l^{\text{th}}$  scene. Eq. (3) could be easily implemented by a 1D convolutional layer with  $C$  (i.e., the number of classes) filters of size 1, followed by a sigmoid activation function. Since there is a single detector (SD) for each scene, we will refer to this module as the SD module.

$$\hat{\mathbf{Y}}_{il} = \max_j \{\hat{\mathbf{y}}_{ijl}\} \quad (4)$$

As shown in Eq. (4), for the *prediction aggregator*, we chose a max pooling function to aggregate instance-wise predictions into bag-level predictions, which is consistent with the

SMI assumption. Other pooling functions [31] may also be applicable as long as they are not inconsistent with the SMI assumption.

When training the MIL based models, we gathered audio samples from class  $l$  (i.e.,  $\mathbf{Y}_{il} = 1$ ) as the *positive bags* for class  $l$ , whereas audio samples from other classes (i.e.,  $\mathbf{Y}_{il} \neq 1$ ) are collected as the *negative bags* for class  $l$ . Therefore, for each class, the number of negative bags is  $C - 1$  times of the number of positive bags. In order to solve the imbalance, we apply the *weighted binary cross entropy* for each class, where the positive weight  $\alpha$  is set to  $C - 1$ . The total losses introduced by a sample is the sum of *weighted binary cross entropy* loss of all the classes:

$$L_i = - \sum_{l=1}^C (\alpha \cdot \mathbf{Y}_{il} \log \hat{\mathbf{Y}}_{il} + (1 - \mathbf{Y}_{il}) \log(1 - \hat{\mathbf{Y}}_{il})) \quad (5)$$

It is worth noting that, the bag level prediction vector  $\hat{\mathbf{Y}}_i$  is *not* a normalized posterior probability over all the classes. In other words, it is not necessary that  $\sum_l \hat{\mathbf{Y}}_{il} \neq 1$ . Instead, each node of  $\hat{\mathbf{Y}}_i$  is an independent posterior of detecting distinct sound events for the corresponding class. During testing, the label with the highest posterior is assigned to the test recording.

At this point, we would like to highlight and explain why we suggest that the instance detectors in Eq. (3) will detect *distinct* instances for each class. Suppose two similar detectors  $\mathbf{w}_l$  and  $\mathbf{w}_{l'}$  were learned for the scene  $l$  and  $l'$  respectively. Then there must be instances which co-activate both label  $l$  and  $l'$ . This contradicts the fact that the positive bag of the scene  $l$  must be the negative bag of the scene  $l'$  (for  $l' \neq l$ ). Thus the detector for each scene must find one distinct pattern for that scene.

## 2.5. The multi-temporal scale (MTS) module

For the *bag-of-instances* representations generated using the previously mentioned CNN-MIL model, each instance vector is (indirectly) connected to all the frequency bins of the input spectrogram. Meanwhile, each instance vector reaches only a limited (about 36 frames) temporal receptive field (TRF). Therefore, to cover both transient sound events patterns (e.g., birds singing) and the long-lasting sound events (e.g., an engine idling), a multi-temporal scale (MTS) module is proposed to improve over the CNN-MIL model.

As shown in the Fig. 3, dilated convolution [32] is adopted to exponentially increase the TRF of each instance vector. The MTS module consists of three stacked 1D dilated convolution layers, with a filter size of 3, stride of 1 and dilation rate of  $r = 1, 2, 4$  respectively. In this way, the TRF of the last layer is seven times the TRF of the input layer. Batch normalization and ReLU are applied after each dilated convolution layer and proper zero padding is added to keep the 'time' axis of the feature map fixed. At last, four feature-maps are concatenated over the 'feature' dimension, and a 1D convolution with 256 filters of size 1 is used to combine the four feature maps. This module could be employed right after the *instance generator* of the CNN-MIL model.

## 2.6. The multi-detector (MD) module

As described in the introduction, acoustic scenes usually consist of multiple events. Considering instance vectors are high-level representations for sound events, the distribution of instance vectors inside a bag are inevitably multi-modal. Thus,

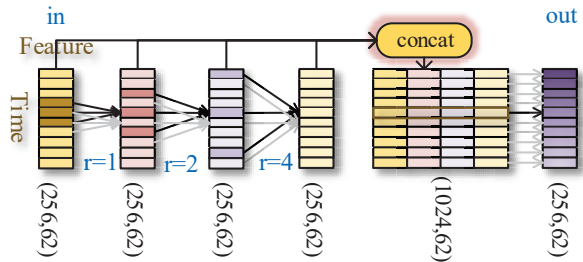


Figure 3: Block diagram of the multi-temporal scale module.

there might be multiple distinct sound events for each scene. In the CNN-MIL model, only one 'template'  $w_l$  is learned for each scene  $l$ , which runs contrary to the multi-modality of the sound events. Therefore, we further propose to use *multiple* distinct instance detectors for each scene instead of one detector, inspired by the *sub-concepts* layer presented in [33].

$$\begin{aligned} \mathbf{a}_{ijlk} &= \mathbf{w}_{lk}^T \mathbf{x}_{ij} + \mathbf{b}_{lk} \\ \mathbf{a}_{ijl} &= \max_k \{\mathbf{a}_{ijlk}\} \\ \hat{\mathbf{y}}_{ijl} &= \frac{e^{\mathbf{a}_{ijl}}}{\sum_l e^{\mathbf{a}_{ijl}}} \end{aligned} \quad (6)$$

where,  $\mathbf{w}_{lk}, \mathbf{x}_{ij} \in \mathbb{R}^d, \mathbf{b}_{lk} \in \mathbb{R},$   
 $\forall l \in [1..C], \forall k \in [1..K]$

As shown in Eq. (6), we allow the model to learn at most  $K$  detectors ( $\{\mathbf{w}_{lk}\}_{k=1}^K$ ) for each scene  $l$ , where  $K$  is a hyperparameter and is set by preliminary experiments. Then, the max pooling function is used to aggregate evidence from the  $K$  detectors. This means a distinct sound event for scene  $l$  is said to be identified if any of the detectors of the scene  $l$  found a match. At last, we apply a *softmax* layer to normalize the evidences over the scene labels  $l$ . This means if one instance is said to be a distinct sound event for one scene, it could not be a distinct sound event for other scenes at the same time. This multi-detector (MD) module can replace the single detector (SD) module as in Eq. (3).

# 3. Experiments

## 3.1. Dataset

For our experiments, we used the development set of DCASE2018 Task1 Subtask B [34], which is the largest freely available dataset for ASC. Materials from the device A (high-quality) are utilized, which contain single-channel audios with a sampling rate of 44.1 kHz. The dataset consists of ten *acoustic scene* classes, where each scene has 864 segments of 10 seconds in length, resulting in a total of 24 hours of audios. The default official partition of training and testing folds is adopted.

## 3.2. Experimental setups

For input features, we follow the configurations of the official baseline of the DCASE2018 challenge [34]. The log mel-spectrogram is firstly extracted from each audio wave, with a frame length of 40 ms, 50% hop size, and 40 mel-bands. Therefore, a feature map of shape (40, 500) is generated for each audio sample and fed into the proposed models. Models are trained using an Adam [35] optimizer with a batch size of 256 and an initial learning rate of 0.001. We decay the learning rate

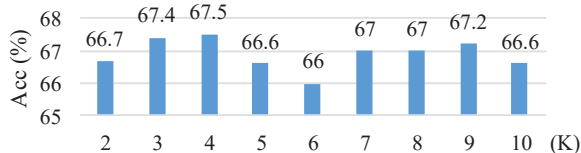


Figure 4: Influence of the hyper-parameter  $K$

with a factor of 0.5 when the validation accuracy does not improve for 3 consecutive epochs, which contributes marginally to performance. We train the models for 50 epochs and the results with the highest accuracy are reported. The models are implemented using Pytorch [36], and we have made our code publicly available at <https://github.com/hackerekcah/distinct-events-asc>

### 3.3. Experimental results

#### 3.3.1. Selection of hyper-parameter $K$

The hyper-parameter  $K$  in Eq. (6) controls the maximum number of distinct sound events that could be detected for each scene. To examine how it affects performance, we replace the SD module in the CNN-MIL model with the proposed MD module and gradually increase the value of  $K$  from 2 to 10. The results are plotted in Fig. 4. From this it can be seen that increasing the value of  $K$  does not necessarily improve performance. The model achieves highest accuracy at  $K = 4$ . We speculate that with large  $K$ , the model may just learn duplicate sound event detectors. Thus in the following section, we set  $K = 4$  for the MD module.

#### 3.3.2. Performances and discussions

Table 1 presents the performance of our proposed models. All models were trained and tested 10 times by varying the random seeds. The mean and standard deviation of the performance from these 10 independent trials are reported. For comparison, we include results from the official baseline [34] as well as the best-performing single (as opposed to *fusion*-based methods) model [16] we could find in the literature. The results are directly extracted from the reference papers.

As can be seen, although our proposed models have not yet achieved the state-of-the-art [16], all the proposed models improve over the official baseline by a large margin. In addition, to evaluate the proposed MTS and MD module, we proposed four models that form the ablation study for the two modules. Comparing model pairs (① vs ②) and (③ vs ④), we can see that the multi-temporal scale (MTS) module improved the results to a minor extent. Alongside this, comparing the model pairs (① vs ③) and (② vs ④), we can see that the MD module moder-

Table 1: Performance comparison of the models. Models in each row are named and described in terms of inclusion (✓) or exclusion (×) of the MTS and / or MD module.

Models	MTS	MD	Acc(%)
Baseline [34]	-	-	58.9 ( $\pm 0.8$ )
Modified Xception [16]	-	-	<b>76.9</b>
① CNN-MIL	×	×	64.2 ( $\pm 1.1$ )
② CNN-MTS-MIL	✓	×	65.4 ( $\pm 0.7$ )
③ CNN-MD-MIL	×	✓	66.5 ( $\pm 0.8$ )
④ CNN-MTS-MD-MIL	✓	✓	<b>68.3</b> ( $\pm 0.9$ )

True class	street_traffic	airport	metro_station	metro	park	tram	street_pedestrian	bus	shopping_mall	public_square	Recall
street_traffic	213	4					13		16		86.6%
airport	2	207	26			2	4		13	11	78.1%
metro_station	4	22	201	6	1	4	2	7	11	1	77.6%
metro	2	4	24	200			20		6	2	76.6%
park	2	5	6		199	4	7		1	18	82.2%
tram	2		5	45	3	180		25	1		69.0%
street_pedestrian	16	12	12	2	1	1	160		11	32	64.8%
bus	1		9	22	4	52		154			63.6%
shopping_mall			119	5					144	5	51.6%
public_square	44	1	6		26	1	20	7		111	51.4%

Figure 5: The confusion matrix of the CNN-MTS-MD-MIL model. The recall for each class is shown on the right.

ately improved the performance in both cases, which suggests allowing detecting of multiple distinct sound events is important for ASC. Finally, combining the two modules, we achieve the highest accuracy (68.3%) of all our proposed models.

Further insight about the proposed MIL based ASC system can be obtained by analyzing the confusion matrix. As shown in Fig. 5, the worst case is when the system predicts 'airport' instead of 'shopping mall'. This situation can happen when the distinct sound events detectors learned for the 'airport' during training actually exist in the 'shopping mall' during testing. In addition, confusions are observed between scenes with similar prominent events, such as 'metro', 'tram' and 'bus'. We expect that this confusion can be reduced by combining evidence from previous strategies.

A number of factors could be investigated to further improve performance. For example, the *prediction aggregator* has been proven to affect the performance of the MIL model significantly for sound event detection [31] [37], it remains to be seen how this would affect our models. Furthermore, CNN embeddings pretrained from large scale sound event dataset may be utilized to guide the *instance generator*. Moreover, an interesting and perhaps valuable product of the MIL model is the *instance-level* predictions. This information may be further exploited in some way for better inferring *bag* labels.

## 4. Conclusions

In this paper, we presented a new strategy for ASC, which recognizes acoustic scenes by identifying distinct sound events. Distinct sound events are not predefined by the user, instead, they are identified implicitly by using an MIL framework. We show that reasonable results can be achieved by using the proposed CNN-MIL model. Furthermore, we show that the proposed MTS and MD modules consistently improve the basic CNN-MIL model, highlighting that modeling the multi-temporal scale and multi-modal nature of sound events is important. Additionally, the proposed modules are not restricted to ASC and may be applied to other related tasks, such as sound event detection and bird sound detection. Finally, this study also provides an opportunity for future combinations of this strategy with previous ones.

## 5. Acknowledgements

This research is supported by the National Natural Science Foundation of China under grant No. U1736210.

## 6. References

- [1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.
- [2] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.
- [3] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, 2006.
- [4] R. Cai, L. Lu, A. Hanjalic, H. Zhang, and L. Cai, "A flexible framework for key audio effects detection and auditory context inference," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1026–1039, 2006.
- [5] T. Heittola, A. Mesaros, A. J. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *2010 18th European Signal Processing Conference (EUSIPCO)*, 2010, pp. 1272–1276.
- [6] J.-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.
- [7] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time–frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 142–153, 2015.
- [8] S. Abidin, R. Togneri, and F. Sohel, "Enhanced lbp texture features from time frequency representations for acoustic scene classification," in *Proc. ICASSP*. IEEE, 2017, pp. 626–630.
- [9] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Feature learning with matrix factorization applied to acoustic scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1216–1229, 2017.
- [10] P. Sharma, V. Abrol, and A. Thakur, "Ase: Acoustic scene embedding using deep archetypal analysis and gmm," in *Proc. Interspeech*, 2018, pp. 3299–3303.
- [11] H. Song, J. Han, and S. Deng, "A compact and discriminative feature based on auditory summary statistics for acoustic scene classification," in *Proc. Interspeech*, 2018, pp. 3294–3298.
- [12] H. Eghbal-zadeh, B. Lehner, M. Dorfer, and G. Widmer, "A hybrid approach with multi-channel i-vectors and convolutional neural networks for acoustic scene classification," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2749–2753.
- [13] H. Zeinali, L. Burget, and J. H. Cernocky, "Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge," in *Proc. DCASE2018 Workshop*, November 2018, pp. 202–206.
- [14] J. Li, W. Dai, F. Metzke, S. Qu, and S. Das, "A comparison of deep learning methods for environmental sound detection," in *Proc. ICASSP*. IEEE, 2017, pp. 126–130.
- [15] H. Chen, P. Zhang, H. Bai, Q. Yuan, X. Bao, and Y. Yan, "Deep convolutional neural network with scalogram for audio scene modeling," in *Proc. Interspeech*, 2018, pp. 3304–3308.
- [16] Y. Liping, C. Xinxing, and T. Lianjie, "Acoustic scene classification using multi-scale features," in *Proc. DCASE2018 Workshop*, November 2018, pp. 29–33.
- [17] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," DCASE2018 Challenge, Tech. Rep., September 2018.
- [18] Z. Ren, Q. Kong, K. Qian, M. Plumbley, and B. Schuller, "Attention-based convolutional neural networks for acoustic scene classification," in *Proc. DCASE2018 Workshop*, November 2018, pp. 39–43.
- [19] T. Zhang, K. Zhang, and J. Wu, "Temporal transformer networks for acoustic scene classification," in *Proc. Interspeech*, 2018, pp. 1349–1353.
- [20] —, "Multi-modal attention mechanisms in lstm and its application to acoustic scene classification," in *Proc. Interspeech*, 2018, pp. 3328–3332.
- [21] S. Mun, S. Park, D. K. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane," in *Proc. DCASE2017 Workshop*, November 2017, pp. 93–102.
- [22] Z. Teng, K. Zhang, and J. Wu, "Data independent sequence augmentation method for acoustic scene classification," in *Proc. Interspeech*, 2018, pp. 3289–3293.
- [23] V. T. Peltonen, A. J. Eronen, M. P. Parviainen, and A. P. Klapuri, "Recognition of everyday auditory scenes: potentials, latencies and cues," in *Audio Engineering Society Convention 110*. Audio Engineering Society, 2001.
- [24] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial intelligence*, vol. 201, pp. 81–105, 2013.
- [25] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 1038–1047.
- [26] Y. Wang, "Polyphonic sound event detection with weak labeling," Ph.D. dissertation, Carnegie Mellon University, 2018.
- [27] A. Kumar and B. Raj, "Audio event and scene recognition: A unified approach using strongly and weakly labeled data," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 3475–3482.
- [28] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. Hadley, A. S. Hadley, and M. G. Betts, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *The Journal of the Acoustical Society of America*, vol. 131, no. 6, pp. 4640–4650, 2012.
- [29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37, Lille, France, Jul 2015, pp. 448–456.
- [31] Y. Wang, J. Li, and F. Metzke, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," *CoRR*, vol. abs/1810.09050, 2018.
- [32] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015.
- [33] J. Feng and Z. Zhou, "Deep MIML network," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 1884–1890.
- [34] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. DCASE2018 Workshop*, November 2018, pp. 9–13.
- [35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
- [36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.
- [37] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 11, pp. 2180–2193, 2018.