INSTANCE-LEVEL LOSS BASED MULTIPLE-INSTANCE LEARNING FOR ACOUSTIC SCENE CLASSIFICATION

WON-GOOK CHOI **JOON-HYUK CHANG** **JAE-MO YANG** **HAN-GIL MOON**

ABSTRACT

In acoustic scene classification (ASC) task, an acoustic scene consists of diverse attributes and is inferred by identifying combinations of some distinct attributes among them. This study aims to extract and cluster these attributes effectively using a multiple-instance learning (MIL) framework for ASC. MIL, known as one of the weakly supervised learning methods, is a way to extract instances from input data and infer a scene corresponding to the input data with those unlabeled instances. We develop the MIL framework more suitable for ASC systems, adopting instance-level labels and instance-level loss, which are effective in extracting and clustering instances. As a result, the witness rate increases significantly compared to the framework without instance-level loss and labels. Also in several MIL-based ASC systems, the classification accuracy improves by about 5 to 11% than without instance-level loss. In addition, we designed a fully separated convolutional module which is a low-complexity neural network consisting of pointwise, frequency-sided depthwise, and temporal-sided depthwise convolutional filters. Considering both complexity and performance, our proposed system is more practical compared to previous systems on the DCASE 2019 challenge task 1-A leader board. We surpassed the third-place model by achieving a performance of 82.3% with only the model complexity of 417K, which is at least 40 times fewer than other systems.

***K*eywords** Acoustic scene classification · low-complexity · multiple-instance learning

1 Introduction

Acoustic scene classification (ASC) is a task that aims to aware acoustic environments by considering large contexts and combinations of sound events, mood, room size, culture, etc [1,2]. It is widely accepted as one of the main functions in a machine hearing [3] along with speech recognition, sound event detection and separation [4] for acoustic environmental awareness. The achievement of efficient ASC is also known to the main challenge in the detection and classification of acoustic scenes and events (DCASE) community, which has been of special interest to audio AI researchers using deep neural networks, in recent years [5–8].

ASC considers large contexts among complex acoustic data, unlike sound event detection and sound separation tasks, which usually treat target events or sources. A particular sound event could represent an acoustic scene, but it is not unique; it could also appear in other scenes, as well. For example, a tram sound or noise usually represents a tram scene (i.e., inside the tram); however, it might appear on a street pedestrian near a tramway. In other cases, the combinations of sound events, acoustic noises, and even echoes are part of the acoustic scenes [9]. There may even be no distinct events or attributes, such as silent situations or environments [1,9]. This means that humans can recognize a park's acoustic scene with sounds of moving water and chirping of birds; however, if there are no events, they cannot distinguish whether it is a park or other outside scenes, although it is still a park.

Ever since the adoption of deep neural networks (DNNs), there have been many developments in ASC systems that treat the above problems with end-to-end learning methods. In particular, convolutional neural networks (CNNs) [10–12] significantly affect feature learning from pre-processed raw features, such as the log-mel spectrogram, mel-frequency cepstral coefficients (MFCC), constant-Q transform (CQT), gammatone frequency cepstral coefficients (GFCC), and chromagram [13–15]. The common strategy for designing an ASC system is to extract high-level feature maps that consist of activation scores of events from the raw audio data and classify them with global average pooling (GAP) or fully connected layers (FC layers) to each scene through supervised learning methods.

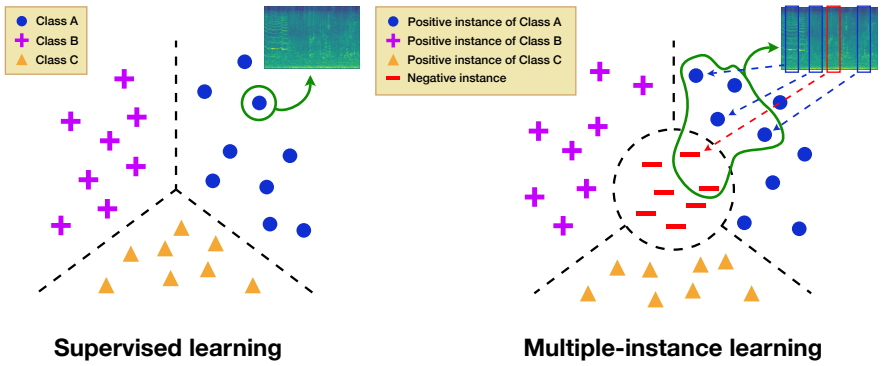


Figure 1: **An illustration of decision boundaries on supervised learning and multiple-instance learning in audio domain.**

There is another strategy that aims to identify distinct sound events from acoustic scenes without manual annotations. Hongwei Song *et al.* [16] first adopted the multiple-instance learning (MIL) [17] trying to detect positive instances (i.e., distinct sound events) that can represent a scene. The MIL is a strategy that infers whether the bag (i.e., input data) is positive or negative and is referenced by a distribution of instances. Considering that bags are labeled with ground truth but not on instances, the MIL-based ASC system is treated as weakly supervised learning owing to the decision method using unlabeled instances. In other words, the bag is determined by the instances which is trained only with the bag-level label without prior information about the instance.

Given that the bag-level loss is back-propagated to the whole model in a previous work, instances are trained indirectly by the bag-level loss. However, because the instances and bags have different label spaces, this way naturally raises questions about whether the instances could be generated well and behave as intended. We hypothesize that directly trained instances would help generate and cluster positive and negative instances. Based on these questions and hypotheses, we reformulate the MIL framework for building ASC systems by adopting instance-level loss. By giving instances positive and negative labels using bag-level ground truth, we can generate the instances much better than those systems that do not have instance-level loss. We also propose an efficient CNN-based network for generating instances, which dramatically reduces the number of parameters. Compared to the DCASE 2019 challenge task 1-A leader board, our proposed method is sufficiently competitive enough for both performance and model complexity.

In the following section, we discuss the background of introducing instance-level loss along with the introduction of MIL. The remainder of this manuscript describes the details of the instance generator and the instance-level loss in Section 3, and we describe the experiments and results in Section 4.

Along with the introduction of MIL, we are talking about the background of introducing instance-level loss.

2 Background: About Multiple-instance learning

In this study, the following three experiments and results are presented for the proposed strategy.

- Introduction of instance-level loss for effective instance learning.
- MIL versus supervised learning based on the proposed MIL framework.
- Objective function according to the viewpoint of label ambiguity

Therefore, this section introduces the background about the MIL related to the above three items.

2.1 Multiple-instance learning

The MIL is a popular learning method [17] that discovers representations of input data into a bag-of-instances (also called a bag), and the instances are defined as representations of entities composing the input data. Let X and f denote an input feature of MIL based system and an instance generator. Then the feature X is mapped to a bag-of-instances by the process $f : X \rightarrow \mathcal{X} \triangleq \{\mathbf{x}_i\}_{i=1}^{n}$, where $\mathcal{X}$, $\mathbf{x}_i$, and n denotes the bag-of-instances, i-th instance, and the number of instances.

The label of X is inferred along by the bag composition with MIL assumptions, such as the standard MIL (SMI) assumption, collective assumption, alternative MIL assumptions, etc [18], and thus the bag classifier is defined based

on the assumptions. The SMI assumption, which is widely used in recent studies as well as past, states that the bag is positive if it contains at least one positive instance, otherwise the bag is negative. In this case, the bag classifier $g(\mathcal{X})$ is:

$$g(\mathcal{X}) = \begin{cases} 1, & \text{if } \exists \mathbf{x} \in \mathcal{X} : h(\mathbf{x}) = 1; \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $h(\mathbf{x})$ detects whether the instance $\mathbf{x}$ is positive or negative. If g is multi-class classifier, (1) is turned to:

$$\begin{aligned} \hat{\mathbf{y}}_i &= \mathbf{h}(\mathbf{x}_i),\ (0 \leq i < N) \\ \hat{Y}_c &= \max_i \hat{y}_{ic},\ (0 \leq c < C) \\ g(\mathcal{X}) &= \operatorname*{argmax}_c \hat{Y}_c \end{aligned} \tag{2}$$

where N and C is the number of instances and classes, $\hat{y}_{ic}$ and $\hat{Y}_c$ is the c-th component of the instance-level and bag-level inference vector $\hat{\mathbf{y}}_i,\ \hat{\mathbf{Y}} \in \mathbb{R}^c$, and $\mathbf{h}$ detects whether the instance is positive or negative per each class. In other words, the classifier detects whether the bag is positive or negative for each class (i.e., the vector $\hat{\mathbf{Y}}$ consists of the best prediction scores on each class), and then determines the class with the highest confidence. Other assumptions consider a distribution, relationship or combination of positive instances to determine the bag. These positive instances are also called witnesses because the bag is determined by only the most dominant instance or the conditions based on the positive instances regardless of the negative instances.

According to the previous studies of MIL in various fields, an instance is defined in a different way, field by field. For example, in the audio field [16, 19–24], an instance is generated from a bundle of frames on a spectrogram. In an image classification task, an instance is from a small patch of the image [17, 25], and for text data, an instance is from a sentence consisting a document [26, 27]. Considering the above studies, an instance is an intuitive representation generated by a sub-concept of input data domain not just a totally abstract latent vector. So we can know what parts (i.e., instances) are contributing to making the data positive.

Instances are defined as components of a bag, so the bag-level space results in the part of subset of instance-level feature space, and the bag-level probability distribution is calculated with the set of the instance-level probability distribution [28]. After all, it is essential to train the instances well because the bag is a group of instances, and the instances are involved in determining the label of the bag. Unlike the previous study of an MIL-based ASC [16], which only has a bag-level loss, we set an instance-level loss using instance-level labels for training the instances directly and clustering the positive and negative instances well. To avoid manual labeling, we offer the instances to *positive or negative labels* using both bag-level ground truths and instances-level prediction scores on each training step. By doing this, we expect that each instance is clustered to be positive or negative in determining its bag and the result is in Section 4.3.2.

Like other MIL approaches, the above strategy maintains the concept of weakly supervised learning. As a side note, we also attempted the MIL framework-based supervised learning, which gives an instance a *bag-level ground truth* rather than a positive or negative label. In fact, one of the differences between MIL and traditional supervised learning is the number of instances is one or more. In other words, in terms of MIL, traditional supervised learning method is treated as single-instance single-label learning which is degenerated version of MIL, so the instance and bag are matched one-to-one [29]. On the other hand, MIL learns the decision boundaries for instances and a final decision is made based on where these instances belong (Fig. 1).

When giving instance-level labels ground truths, it has the same label space in the bag-level and the instance-level, so the instances have a different representation of what we have aforementioned. In other words, setting the bag-level ground truth as an instance-level label is an approach that combines the MIL framework with supervised learning rather than the MIL itself. In terms of traditional supervised learning, it is like an ensemble system that breaks the frame into smaller pieces for inference and chooses the most confident value from the results of each segment. We compare the result1s of the supervised learning approach based on the MIL framework with the weakly supervised learning method, in Section 4.3.2.

2.2 Label ambiguity in MIL

Label ambiguity is an MIL problem caused by the different label spaces at the instance-level and bag-level [17]. Ideally, a positive instance belonging to one scene cannot appear in the other scenes [16]. For example, in Fig. 2, the spectrogram is a park scene bag, and the stippled areas are the groups of distinct instances that can represent each scene. The shaded area indicates where the bag and other groups of positive instances intersect, and ideally, it should not appear. However, the negative instances extracted from other scenes could fall into a positive area owing to the label

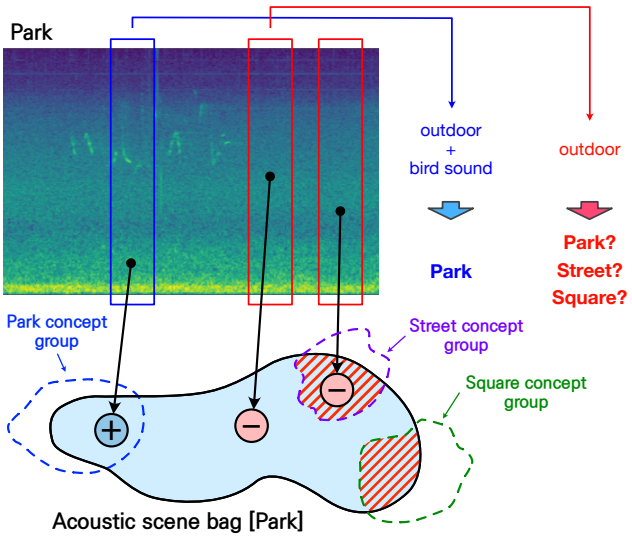


Figure 2: **An example of the instances in an acoustic scene bag.** In the spectrogram of a park scene, the blue instance has a distinct attribute like the bird's sound, whereas the red instances has ambiguous properties. Ideally, there should be no intersection between the street or square concept group and the park bag. However, the intersection appears due to the label ambiguity, so the second negative instance falls into the street concept group.

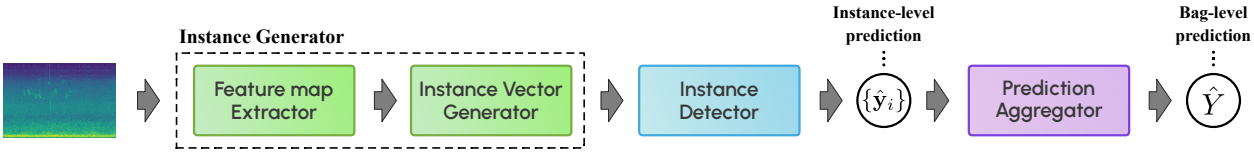


Figure 3: **The overall structure of MIL.**

ambiguity. We consider label ambiguity to be natural because there are common characteristics among the scenes, such as the ambiguity between serene situations of a park and public square. Although an ambiguous instance exists in the bag, its score is lower than the score of the distinct instance (i.e., positive instance) because a negative label is assigned to the ambiguous instance in each training step. Therefore, it has a minor effect while inferring a class of the bag. The above properties about label ambiguity is also shown to several experimental results in section 4.

3 Proposed Strategy

There are two key points in MIL: one is how to generate instance vectors and the other is how to infer the bag-level class using instances. We focus on the first point by adopting an effective instance generator and the instance-level loss.

3.1 Instance generator

The instance generator can be separated into two steps: the feature map extractor and the instance vector generator (Fig. 4).

3.1.1 Feature map extractor

Instead of using the VGG-like feature extractor in [16], we adopt a spatially separable convolutional layer [30] with pointwise (PW) and depthwise (DW) convolution layers [31]. In terms of the complexity, the number of parameters of a standard convolutional layer is:

$$C_{in} \cdot C_{out} \cdot K_H \cdot K_W, \tag{3}$$

where the complexity depends on the multiplication of the input and output channel size C_{in} and C_{out}, and the height and width of the kernel K_H and K_W. A depthwise separable convolution [31] is an efficient version of the convolutional filter while maintaining a receptive field. It consists of two layers: a depthwise convolution and a pointwise convolution.

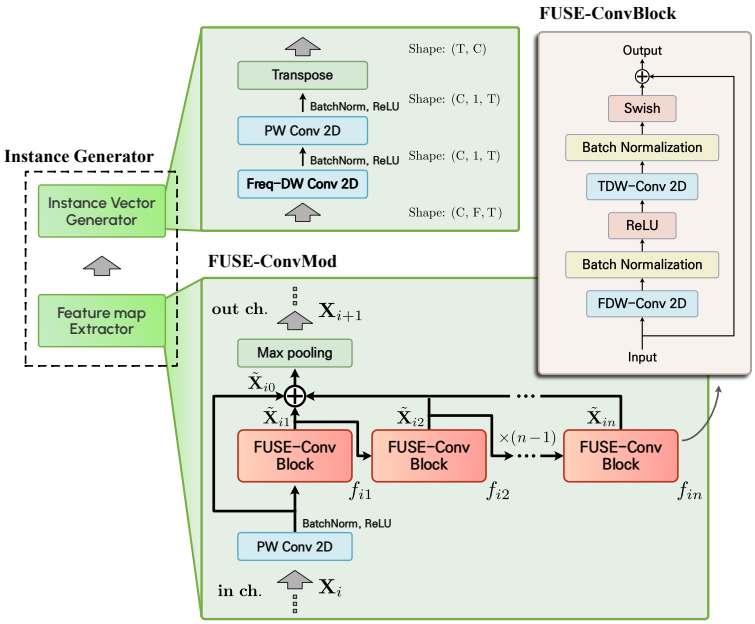


Figure 4: **Architecture of the instance generator.** It consists of the feature map extractor which is made up of several FUSE-ConvMods, and the instance vector generator composed full-sized FDW-Convlayer and PW Conv2d. In the FUSE-ConvMod, n denotes the number of FUSE-ConvBlocks.

The complexity of a depthwise separable convolutional layer is caculated as follows:

$$\underbrace{C_{in} \cdot C_{out}}_{Pointwise} + \underbrace{C_{out} \cdot K_H \cdot K_W}_{Depthwise} = C_{out}(C_{in} + K_H \cdot K_W). \tag{4}$$

The spatial separable convolutional layer has totally factorized convolutional filters. It replaces a 3×3 filter with a combination of 3×1 and 1×3 filters. The complexity of a spatial separable convolutional layer is given below:

$$\begin{aligned} &\underbrace{C_{in} \cdot C_{out}}_{Pointwise} + \underbrace{C_{out} \cdot K_H}_{(K_H \times 1)\ Depthwise} + \underbrace{C_{out} \cdot K_W}_{(1 \times K_W)\ Depthwise} \\ &= C_{out}(C_{in} + K_H + K_W). \end{aligned} \tag{5}$$

When the output channel size C_{out} is equal to the number of filters, thus the complexity of the spatially separable convolutional layer is calculated by the addition operation only instead of multiplication. This means that the larger the filter size and number of channels, the more efficient the complexity.

In contrast to the image data, which have spatial properties, a spectrogram shows different meanings along by a frequency and time axis. Therefore, the kernels from the spatially separable convolution turn into the frequency-side kernel and temporal-side kernels. According to previous studies with the spatially separable convolution in audio fields, the keyword spotting [32] and ASC [11] tasks have been shown to be beneficial in modeling fewer parameters as well as for performance improvements.

The feature map extractor consists of four *FUlly SEperated Convolution* modules (FUSE-ConvMods), which have harmonies of PW, frequency-side DW (FDW), and temporal-side DW (TDW) convolutional layers. The input feature passes a PW convolutional layer followed by several *FUlly SEperated Convolution* blocks (FUSE-ConvBlocks) and is pooled through a max-pooling layer. Mathematically, the output employed by the i-th FUSE-ConvModule is:

$$\begin{aligned} \tilde{\mathbf{X}}_{i0} &= \text{PW-Conv2d}(\mathbf{X}_i) \\ \tilde{\mathbf{X}}_{ik} &= f_{ik}(\tilde{\mathbf{X}}_{ik-1})\ (k = 1...n) \\ \mathbf{X}_{i+1} &= \text{MaxPooling}(\sum_{k=0}^{n} \tilde{\mathbf{X}}_{ik}), \end{aligned} \tag{6}$$

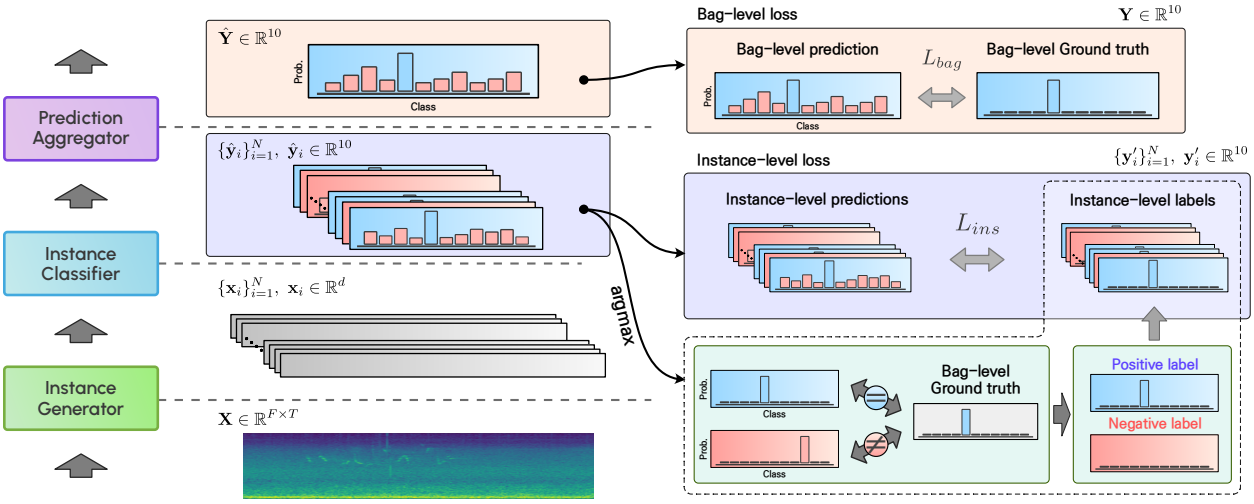


Figure 5: **Schematic diagram of MIL with instance-level loss.** $\mathbf{X}$, $\mathbf{x}_i$, N, and d denote an input spectrogram, an instance, the number of instances, and the dimension of instances, respectively. And $\hat{\mathbf{Y}}$, $\hat{\mathbf{y}}$, $\mathbf{Y}$, and $\mathbf{Y}'_i$ denote the inference of the bag and instance, and the bag-level and instance-level label.

where $\mathbf{X}_i$, N, and f_{ik} denote the input of the i-th FUSE-ConvMod, the number of the FUSE-ConvBlock and k-th block in the i-th FUSE-ConvMod, respectively, as shown in Fig. 4. As an exception, the last FUSE-ConvMod does not carry out pooling.

3.1.2 Instance vector generator

The output of the feature map extractor is still a tensor that shapes (*channel, frequency, frame*). We aggregate the tensor along the frequency axis using the FDW convolutional layer with a full-size (i.e., fully connected) kernel in the instance vector generator. In this study, we obtain the shape of the output from the feature map extractor (*256, 32, 39*); thus, we use the (32×1) FDW convolutional layer, which is followed by affine transform maintaining channels; then, it is reshaped and transposed.

3.2 Loss

As described in Section 2.1, we consider the instance-level loss as well as the bag-level loss for clustering the positive and negative instances. As a result, the total loss can be expressed as follows:

$$L_{total} = L_{bag} + L_{ins}. \tag{7}$$

3.2.1 Bag-level loss

The fundamental objective of the MIL method is to decide whether each bag is positive or negative. The loss is introduced by the mean of the *weighted binary cross entropies* (wBCEs), which is used to calculate the mean of the BCEs for each class by considering the imbalance of positive and negative classes. We also experimented using the cross-entropy (CE) for the bag-level loss which is adopted by most of the classification tasks, and we compared the results of using the mean of the wBCEs and CE loss in Section 4.3.3. Eq. (8) shows the mean of the wBCEs and CE that we adopted for the loss function, where $\overline{wBCEs}$, C, and α is the mean of the wBCEs, the number of the classes, and the class imbalance factor, respectively, and we set α to $C - 1$. If α is one, the mean of the wBCEs is same with the mean of the BCEs, which does not consider the class imbalance.

$$\begin{aligned} \overline{wBCEs}_{bag} &= -\frac{1}{C}\sum_{l=1}^{C}(\alpha \cdot Y_l \ln \hat{Y}_l + (1 - Y_l)\ln(1 - \hat{Y}_l)) \\ CE_{bag} &= -\sum_{l=1}^{C} Y_l \ln \hat{Y}_l \end{aligned} \tag{8}$$

3.2.2 Instance-level loss

Because there are no labels on instances, we assign the labels using the bag-level ground truth to each of them. Eq. (9) and Fig. 5 show how to define the instance-level label using the prediction of instances. We assign a positive label if the maximum prediction score's index among the classes is the same as the ground truth at each instance; otherwise, we assign a negative label. The positive label is defined as a one-hot vector, and the negative label is defined as a zero vector. For the same reason as in the bag-level loss, we use the mean of the wBCEs for instance-level loss, which allowed the model to cluster the distinct instances and negatives. We average the entropies in all classes and averaged them for the bag.

$$
\begin{aligned}
L_{ins} &= -\frac{1}{NC}\sum_{i=1}^{N}\sum_{l=1}^{C}(\alpha \cdot y'_{il} \ln \hat{y}_{il} + (1 - y'_{il})\ln(1 - \hat{y}_{il})) \\
y'_{il} &= \begin{cases} 1, & \text{if } \operatorname{argmax}_l \hat{y}_{il} = t \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
\tag{9}
$$

At the beginning of the training, all parameters are set randomly, and instance-level labels are set as random labels, whether the instances are distinctive or not. Fortunately, because the bag-level loss is calculated with the ground truth, it correctly guides the instance-level label to the right way as the training step progressed. Otherwise, the instance-level labels would fall in far wrong label space.

4 Experiments

4.1 Datasets

To evaluate the performance of the proposed method, we fully experimented on the TAU Urban Acoustic Scenes 2019 dataset [33], recorded to build the ASC system on a single device. It was recorded by using a 48 kHz sampling rate with stereo channels. The 40h of audio recordings consisted of 14,400 segments in 10 different acoustic scenes. Each segment was 10s long. There were 9,185 segments for training, and 4,185 segments for validation. In addition, to compare with the method proposed in [16], we evaluated on the TUT Urban Acoustic Scenes 2018 Mobile [34], as well.

4.2 Training setup

Table 1: **Instance generator.** A row shows a stage (e.g. feature map extractor or instance vector generator). **n**, **c**, and **p** denotes the blocks repeated in the FUSE-ConvMod, the output channel of each network, and pooling or not at the end of the FUSE-ConvMod, respectively.

Stage	Input shape	Network	n	c	p
Feature	1, 256, T	FUSE-ConvMod	3	32	O
map	32, 128, T/2	FUSE-ConvMod	3	64	O
Extractor	64, 64, T/4	FUSE-ConvMod	3	128	O
	128, 32, T/8	FUSE-ConvMod	3	256	X
Instance	256, 32, T/8	FDW-Conv2D	-	256	-
vector	256, 1, T/8	PW-Conv2D	-	256	-
Generator	256, 1, T/8	Reshape	-	-	-
	256, T/8	Transpose	-	-	-

Prior to the feature extraction, each segment was downmixed to 16 kHz and a single channel with averaging. Then, we extracted the log mel-spectrogram for the input feature with a window size of 128ms, hop size of 32ms, and 256 mel bins. We optimized with the SGD optimizer and learning rate scheduler using cosine annealing with warming up the first five epochs for 100 epochs. The initial learning rate, L2 norm, and batch size were 0.06, 0.001, and 48, respectively. The detail of the instance generator is shown in Table 1. In addition, we used a sigmoid function following the (256×10) linear layer for the instance detector and a max-pooling layer for the prediction aggregator. We used the SMI assumption that the bag is positive if there is a positive instances [16, 17].

Table 2: **Comparison of performance with or without instance-level loss.** GAP and FC classifiers skip the instance vector generator in Table 1. With the GAP classifier, feature maps are reformulated into ten channels and averaged into a point after the extractor. With the FC layer, feature maps are pooled to 4×4 size before flattening. Then each node is connected to 64 nodes before the 10 nodes (i.e., the number of classes). BCE follows a sigmoid function, and CE does a softmax function. The learning rate of the GAP and FC are 0.06 and 0.001, respectively. Each performance results on ten random seeds.

Datasets	Extractor	Mel bins	Classifier	Params	Instance loss	Loss function	Accuracy(%)
TUT 2018	VGG-like	40	MIL	454K	X	$\overline{wBCEs}$	64.2 ± 1.1 [16]
					O	$\overline{wBCEs}$	**69.57 ± 0.98**
TAU 2019	VGG-like	256	MIL	1.34M	X	$\overline{wBCEs}$	66.35 ± 0.35
					O	$\overline{wBCEs}$	**77.17 ± 0.06**
	FUSE-ConvMod	256	MIL	139.08K	X	$\overline{wBCEs}$	72.22 ± 0.82
					O	$\overline{wBCEs}$	**79.13 ± 0.27**
					O	$\overline{BCEs}$	**78.57 ± 0.67**
					O	CE	73.41 ± 0.93
			GAP	130.12K	-	$\overline{BCEs}$	71.59 ± 0.10
					-	CE	71.61 ± 0.85
			FC	324.62K	-	$\overline{BCEs}$	72.46 ± 0.83
					-	CE	72.70 ± 1.07

4.3 Results and Discussions

The results of the experiments are shown in Table 2. We formed the four comparison groups to demonstrate the superiority of our proposed method:

- The feature map extractor in [16].
- The MIL with and without the instance-level loss.
- The traditional supervised learning method using a classifier with an FC layer or GAP.
- Whether the bag-level and instance-level loss functions were the mean of the BCE or CE.

We repeated each experiment on ten random seeds.

4.3.1 FUSE-ConvMod

As shown in Table 2, the extractor with the FUSE-ConvMod had significant effects on both the performance and the number of parameters compared to the VGG-like feature map extractor in [16]. The number of parameters decayed by approximately 90%, and the performance improved by 2-6% for each system. As previously mentioned, an image has rectangular-shaped local information, whereas a spectrogram has information along with a harmonic or a combination of a frequency axis and continuity over a time axis. In other words, a spectrogram has a different role on the x- and y-axes in feature maps, unlike image filtering. Therefore, as supported by the experimental results above, applying frequency and temporal filters separately could be more significant than applying a patch-like filter.

In addition, we checked our performance on the leader board[1] of the DCASE challenge 2019 Task1-A, which is one of the most well-known competitions related to scene classification (Table 3). All participants were required to design systems using the TAU Urban Acoustic Scenes 2019 development dataset recorded with the same device in this task. To maximize the performance, we built an ensemble system using three different parametric models, which is a soft voting classifier at the instance-level. Compared with the top 1-3 systems, our systems improved its performance by a significant level while keeping the model complexity very low. Notably, by building an ensemble system, it surpassed the 3rd place system, and although the model size has become tripled, it was still significantly smaller than the other system. This result shows the MIL framework has great potential for building ASC systems.

4.3.2 Effect of instance-level loss

The effect of the instance-level loss was remarkable. As shown in Table 2, training with the instance-level loss improved accuracy by at least 5% compared to otherwise. We demonstrated that training the instances directly by adding the

[1] http://dcase.community/challenge2019/task-acoustic-scene-classification-results-a

Table 3: **DCASE 2019 Task1-A leader board.** We compare the performance of our proposed method with the top 3 systems of the challenge. Each performance is a result of the development dataset. For reference, there are no records of 4th and 5th systems on the development dataset, and they are about 1% lower than the 3rd place system on the evaluation dataset. Also, in this table, we adopted the decision-making with the number of positive instances, not the SMI assumption. We choose the best performance of both MIL and supervised learning (SupL) system which assigns the positive/negative labels and bag-level ground truth to instance-level labels, respectively. The Ours++ is a simple-voting-based ensemble system, which used three differently trained models.

System	Model Complexity	Accuracy(%)
Hangting Chen *et al.* [35]	48M	85.1
Khaled Koutini *et al.* [36]	71M	83.7
Hyeji Seo *et al.* [37]	18M	81.5
Baseline [7]	116K	62.5
Ours(MIL)	139.08K	80.3
Ours(SupL)	139.08K	81.1
Ours++(SupL)	417.24K	82.3

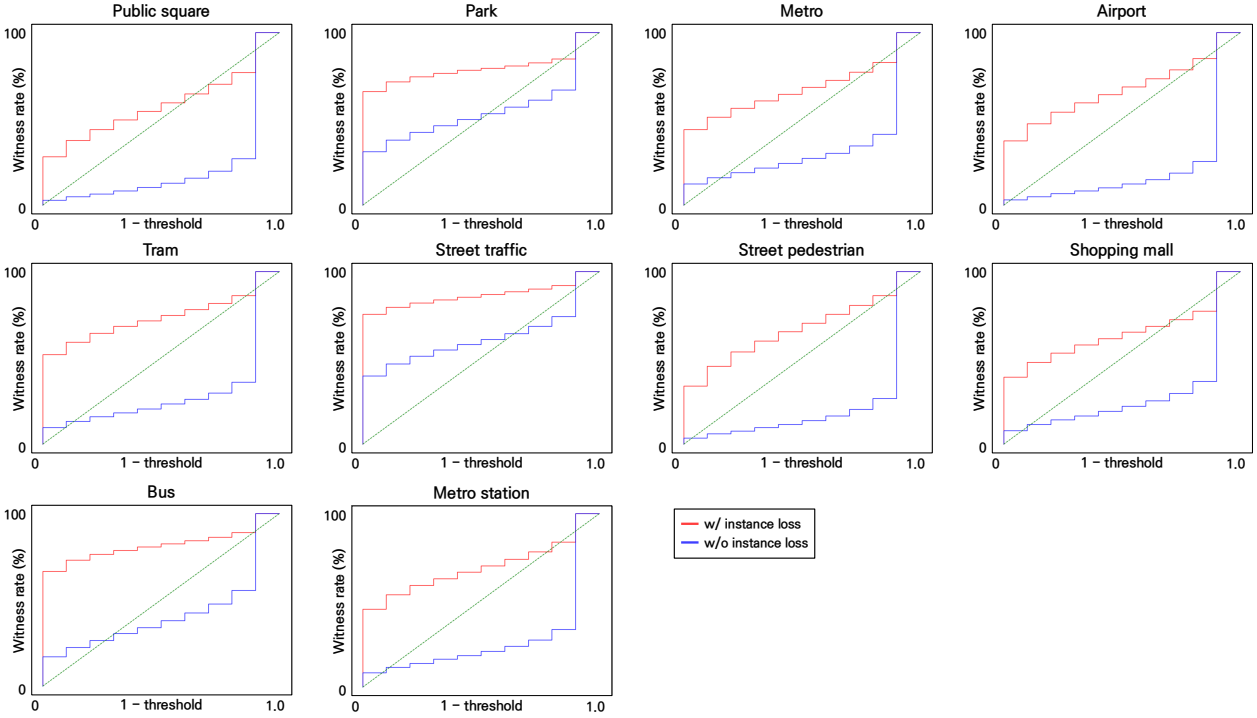


Figure 6: **Witness rate for each class.** We considered a positive instance if the probability was upper than the threshold, else a negative instance. In the park scene, we treated only the probability of the park class, and so did the other scenes.

instance-level loss was more effective than training with the bag-level loss alone, through the witness rate (WR) to the threshold shown in Fig. 6. The WR is introduced by the proportion of positive instances in positive bags [38, 39]. As shown in Fig. 7, the system without the instance-level loss has meager WRs. This means that the system generated poor positive instances, and causes severe class imbalance problems [17]. Meanwhile, the system with the instance-level loss generated richer positive instances; the abundance of positive instances means that the system extracts various features from the data and leads to better performance.

Furthermore, instances are generated and clustered more distinctly using backpropagation of the instance-level loss. We could build a more interpretable ASC system using these instances. In the example in Fig. 8, the instances extracted by the system trained with instance-level loss are so well distributed that we can analyze the relationship between classes and confirm the ambiguity of the label. It shows the tSNE distribution of the positive and negative instances of a

park, shopping mall, and airport scenes. Unlike the park scene, which belongs to an outdoor category, the airport and shopping mall are similar to indoors. Owing to the label ambiguity described in Fig. 2, the negative instances of the airport scenes are partially included in the positive instance's area of the shopping mall scene and same as the negative instances of the shopping mall scenes. However, no negative instances of both airport and shopping mall scenes were included in the positive instance's area of the park scene. In this way, we can figure out the label ambiguity described conceptually in [17], using well-clustered instances via the instance-level loss.

Table 4: **Results according to the instance-level label.** Each performance results on ten random seeds.

System	Instance label	Accuracy(%)
TUT 2018, VGG like	P/N label	69.57 ± 0.98
	Ground truth	70.65 ± 0.41
TAU 2019, VGG like	P/N label	77.17 ± 0.06
	Ground truth	77.98 ± 0.32
TAU 2019, FUSE-ConvMod	P/N label	79.13 ± 0.27
	Ground truth	79.45 ± 0.60

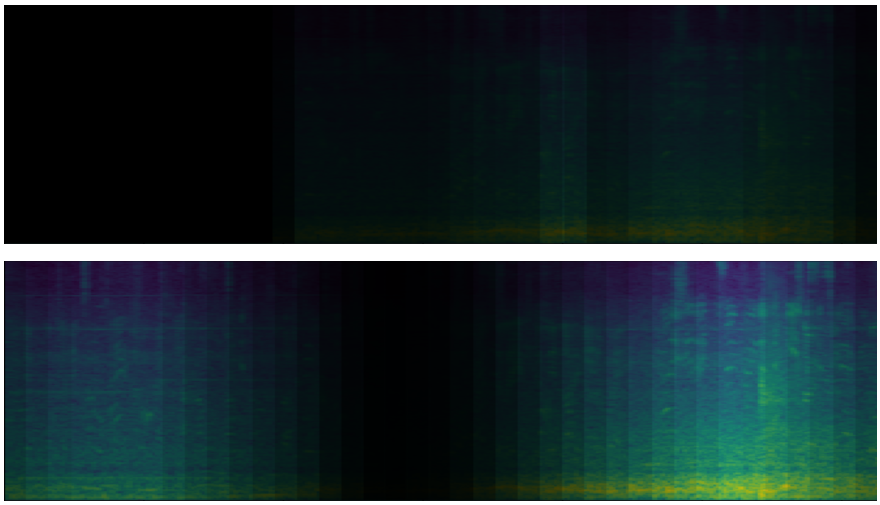


Figure 7: **Masked spectrogram based on prediction scores of instances on a public square spectrogram.** Those spectrograms were extracted from the same segments. The brighter, the more positive instance it was. (Top: w/o instance-level loss, Bottom: w/ instance-level loss.)

The method used to assign instance-level labels is a crucial part of the process. Section 3.2.2 defines the assifnment of positive and negative labels using the bag-level ground truth to cluster the instances. We performed another experiment that assigned the bag-level ground truth to instance-level labels. In general, there are no labels on instances; however, we assumed that the label is known. Therefore, in principle, it is no longer the concept of MIL, and it is no different from supervised learning. Table 4 and Fig. 8 show the difference of the system depending on the method of assigning the instance-level label. For every system in Table 4, each performance slightly improves when giving the ground truth to the instance-level label. However, the tSNE distribution in Fig. 8 exhibits that the method of assigning ground truth to instance-level labels loses its meaning as an instance. In other words, although learning instances using the ground truth might be more practical, the MIL method is so worth researching from the point of view that it is explainable while having only a slight difference in performance.

4.3.3 Objective function

It is crucial to choose an appropriate objective function according to the learning method. As adopting the MIL framework to the multi-classification task, there is a choice between the CE with the softmax and the BCE with the sigmoid as to which objective function to use. When treating the multi-classification task, a prediction score is calculated by softmax function that is generalized of the sigmoid function, and the loss is derived from the CE, in general. However, due to the specificity of the MIL framework to determine whether each instance is positive or

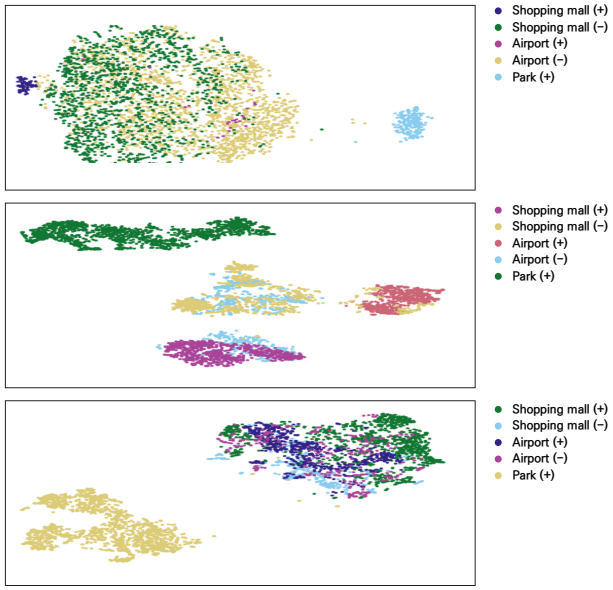


Figure 8: **tSNE distribution of instances.** The first graph shows the distribution of the instances that were trained without the instance-level loss. The others were trained with the instance-level loss: the middle one used positive/negative labels for the instance-level labels, and the bottom one used bag-level ground truth. We set the threshold to 0.9 for the positive instance. The instances of the middle system are well clustered, whereas the others are confusing except the instances of the park. *Indoor*: airport, shopping mall, metro station. *Outdoor*: street pedestrian, public square, street traffic, park. *Transportation*: metro, bus, tram.

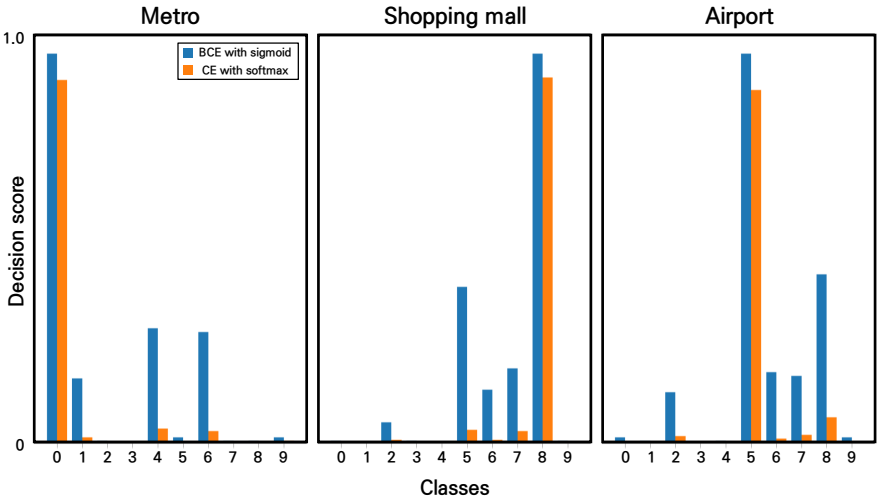


Figure 9: **Averages on bag-level decision scores for each class.** Classes) 0: metro, 1: bus, 2: public square, 3: park, 4: tram, 5: airport, 6: metro station, 7: street pedestrian, 8: shopping mall, 9: street traffic

negative for each class, the mean of the BCEs can be considered, and the sigmoid function is naturally adopted. In this case, the final decision (i.e., bag-level decision) follows the one with the highest bag-level prediction score, but it does not mean the real probability since the sum of the scores is not one.

As shown in Table 2, there is no significant difference in the system's performance in traditional supervised learning (i.e., GAP, FC) whether the loss function is the CE or the mean of the BCEs, whereas there is a big difference when training under the MIL framework. At first, we expected that the softmax function might solve the problem of label ambiguity, hence we assumed that using the softmax with the CE would improve clustering of the distinct instances better. This is because a prediction score following the sigmoid function is independent of each other, but a score

following the softmax whose sum should be one is dependent, so the prediction scores induced by the softmax has high value to one class. However, we found that BCE following sigmoid is more appropriate in MIL, as the experiments in Table 2 tell us empirically. For the instance clustering in MIL, the BCE allows the label ambiguity whereas the CE does not, because the sigmoid function evaluates the probability for each candidate while the softmax function intends to choose only one target. This can be seen in Fig. 9, where the softmax score stands tall on only the target scene, but the sigmoid scores stand out even the non-target scenes, which means ambiguous labels.

The above result confirms that the label ambiguity is natural in MIL; a bag-of-instances may have the property of both non-target scenes and the target scene, such as the sound of a metro can sometimes be perceived as the sound of a metro station. The negative instance of the metro scene could fall in the metro station's concept (i.e., the positive area of the metro station scene), but it does not have much decision-making power when inferring the bag (Fig. 9). This means that we can build a more robust system by designing a MIL-based ASC system that allows the label ambiguity rather than forcing it to classify ambiguous labels.

5 Conclusions

In this study, we proposed an improved MIL framework and instance generator, which is effective in extracting the instances from raw data for ASC. It allows the system to more accurately detect positive instances and describe the relationship between the data by assigning positive and negative labels for instance-level loss. The abundance of positive instances indicates that the system is effective at extracting a wide range of distinct features from the data. In addition, from a practical perspective, we achieved a score of 82.3% while working on the TAU Urban Acoustic Scenes 2019 dataset using the bag-level ground truth to instance-level labels.

References

[1] L. D. Pham, I. V. McLoughlin, H. Phan, and R. Palaniappan, "A robust framework for acoustic scene classification.," in *Proc. Interspeech*, 2019, pp. 3634–3638.

[2] X. Bai, J. Du, Z. R. Wang, and C. H. Lee, "A hybrid approach to acoustic scene classification based on universal acoustic models," in *Proc. Interspeech*, 2019, pp. 3619–3623.

[3] R. F. Lyon, "Human and machine hearing," *Cambridge University Press*, 2017.

[4] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R Hershey, "Universal sound separation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 175–179.

[5] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: analysis of DCASE 2021 challenge systems," *arXiv preprint arXiv:2105.13734*, 2021.

[6] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions," in *Proc. the Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop*, 2020, pp. 56–60.

[7] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. the Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop*, 2018, pp. 9–13.

[8] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: tasks, datasets and baseline system," in *Proc. the Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop*, 2017, pp. 85–92.

[9] A. Hüwel, K. Adiloğlu, and J. Bach, "Hearing aid research data set for acoustic environment recognition," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 706–710.

[10] S. Suh, S. Park, Y. Jeong, and T. Lee, "Designing acoustic scene classification models with cnn variants," Tech. Rep., *the Detection and Classification of Acoustic Scenes and Events* (*DCASE*) *Challenge*, 2020.

[11] B. Kim, S. Yang, J. Kim, and S. Chang, "Qti submission to DCASE 2021: residual normalization for device-imbalanced acoustic scene classification with efficient design," Tech. Rep., *the Detection and Classification of Acoustic Scenes and Events* (*DCASE*) *Challenge*, 2021.

[12] D. Battaglino, L. Lepauloux, N. Evans, F. Mougins, and F. Biot, "Acoustic scene classification using convolutional neural networks," in *IEEE Audio and Acoustic Signal Processing (AASP) Challenge on Detec. 2016*.

[13] S. Si, J. Wang, H. Sun, J. Wu, C. Zhang, X. Qu, N. Cheng, L. Chen, and J. Xiao, "Variational information bottleneck for effective low-resource audio classification," in *Proc. Interspeech*, 2021, pp. 591–595.

[14] J. Sharma, O.-C. Granmo, and M. Goodwin, "Environment sound classification using multiple feature channels and attention based deep convolutional neural network," in *Proc. Interspeech*, 2020, pp. 1186–1190.

[15] L. Pham, I. McLoughlin, H. Phan, and R. Palaniappan, "A robust framework for acoustic scene classification," in *Proc. Interspeech*, 2019, pp. 3634–3638.

[16] H. Song, J. Han, S. Deng, and Z. Du, "Acoustic scene classification by implicitly identifying distinct sound events," in *Proc. Interspeech*, 2019, pp. 3860–3864.

[17] M. A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: a survey of problem characteristics and applications," *Pattern Recognition*, *Elsevier*, 2018, vol. 77, pp. 329–353.

[18] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The knowledge engineering review*, *Cambridge University Press*, 2010, vol. 25, no. 1, pp. 1–25.

[19] Y. Wang, "Polyphonic sound event detection with weak labeling," *PhD thesis*, *Languages Technologies Inst., Carnegie Mellon University*, 2018.

[20] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *Proc. of the 24th Association for Computing Machinery (ACM) International Conference on Multimedia*, 2016, pp. 1038–1047.

[21] A. Kumar and B. Raj, "Audio event and scene recognition: a unified approach using strongly and weakly labeled data," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3475–3482.

[22] Yang Liu, Alexandras Neophytou, Sunando Sengupta, and Eric Sommerlade, "Cross-modal spectrum transformation network for acoustic scene classification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 830–834.

[23] K. X. He, Y. H. Shen, and W. Q. Zhang, "Hierarchical pooling structure for weakly labeled sound event detection," in *Proc. Interspeech*, 2019, pp. 3624–3628.

[24] S. Hong, Y. Zou, and W. Wang, "Gated multi-head attention pooling for weakly labelled audio tagging," in *Proc. Interspeech*, 2020, pp. 816–820.

[25] J. Kumar, J. Pillai, and D. Doermann, "Document image classification and labeling using multiple instance learning," in *IEEE International Conference on Document Analysis and Recognition*, 2011, pp. 1059–1063.

[26] B. Lutz, N. Pröllochs, and D. Neumann, "Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning," *arXiv preprint arXiv:1901.00400*, 2018.

[27] S. Angelidis and M. Lapata, "Multiple instance learning networks for fine-grained sentiment analysis," *Transactions of the Association for Computational Linguistics*, *MIT Press*, 2018, vol. 6, pp. 17–31.

[28] G. Doran and S. Ray, "Multiple-instance learning from distributions," *The Journal of Machine Learning Research*, *JMLR. org*, 2016, vol. 17, no. 1, pp. 4384–4433.

[29] Z. H. Zhou, M. L. Zhang, S. J. Huang, and Y. F. Li, "Multi-instance multi-label learning," *Artificial Intelligence*, 2012, vol. 176, no. 1, pp. 2291–2320.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 2818–2826.

[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[32] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," *arXiv preprint arXiv:2106.04140*, 2021.

[33] M. Mandel, J. Salamon, and D. P. W. Ellis, *"Proceedings of the detection and classification of acoustic scenes and events 2019 workshop (DCASE 2019)"*, *New York University*, 2019.

[34] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," *arXiv preprint arXiv:1807.09840*, 2018.

[35] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," Tech. Rep., *the Detection and Classification of Acoustic Scenes and Events* (*DCASE*) *Challenge*, 2019.

[36] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Acoustic scene classification and audio tagging with receptive-field-regularized cnns," Tech. Rep., *the Detection and Classification of Acoustic Scenes and Events* (*DCASE*) *Challenge*, 2019.

[37] H. Seo and H. Park, "Acoustic scene classification using various pre-processed features and convolutional neural networks," Tech. Rep., *the Detection and Classification of Acoustic Scenes and Events* (*DCASE*) *Challenge*, 2019.

[38] F. Li and C. Sminchisescu, "Convex multiple-instance learning by estimating likelihood ratio," in *Proc. Advances in Neural Information Processing Systems*. *Citeseer*, 2010, vol. 10, pp. 1360–1368.

[39] Y. Han, Q. Tao, and J. Wang, "Avoiding false positive in multi-instance learning," in *Proc. Advances in Neural Information Processing Systems*. *Citeseer*, 2010, vol. 23, pp. 811–819.