THE NU SYSTEMS FOR DCASE 2024 CHALLENGE TASK 2

Technical Report

Takuya Fujimura[1]*, Ibuki Kuroyanagi*[1]*, Tomoki Toda*[2]

[1] Graduate School of Informatics, Nagoya University, Nagoya, Japan
[2] Information Technology Center, Nagoya University, Nagoya, Japan

ABSTRACT

In this report, we present our developed anomalous sound detection (ASD) systems for DCASE 2024 Challenge Task 2. We propose three methods to improve ASD systems based on a discriminative approach. First, we enhance a discriminative feature extractor by using multi-resolution spectrograms as input and implementing new training strategy and data augmentation for its training. Second, we generate pseudo-attribute labels to effectively train the discriminative feature extractor even for some machines without any attribute labels, where the pseudo-attribute labels are obtained by self-supervised learning using artificially processed data as negative samples. Third, we utilize Audioset as an external training dataset to further improve ASD performance, where we carefully extract useful samples from it using a pre-trained feature extractor. Our developed ensemble system has achieved 67.26% in the official scores calculated as a harmonic mean of the area under the curve (AUC) and partial AUC ($p = 0.1$) over all machine types and domains in the development set.

***Index Terms*—** anomalous sound detection, discriminative method, pseudo labels, core-set selection

1. INTRODUCTION

This report describes our submitted systems for the DCASE 2024 Challenge Task 2 [1]. This task focuses on anomalous sound detection (ASD) which aims to detect mechanical failures from sounds emitted by a machine. This year, the organizers set the following five conditions, inheriting four conditions from previous years: (1) training data includes only normal sounds, (2) domain shifts occur, (3) tuning for each machine type is not possible, and (4) a limited number of machines are available for a machine type, and additionally introducing a new condition (5) no attribute information is available for some machine types. Although the attribute information is useful to improve performance [2]–[7], we need to develop new techniques without using it.

ASD methods are classified into generative and discriminative approaches [8], where the discriminative approach often achieves better performance [9], [10]. The discriminative approach trains the feature extractor to classify differences in normal sounds (i.e., machine types and attribute information). During inference, the distance between the observation and normal sound calculated in the discriminative feature space is used as the anomaly score, assuming anomalous sounds are not correctly classified.

Our system is based on the state-of-the-art discriminative method [9], [11], [12] and we propose various techniques to improve its performance in the 2024 Challenge setting. First, we enhance the discriminative feature extractor by utilizing multi-resolution spectrograms together with new training techniques. Second, we generate and utilize pseudo-attribute labels for training to deal with the fifth requirement introduced this year. Third, to address the fourth requirement, we utilize Audioset [13] as an external data resource, proposing a core-set selection method for finding useful samples. We conduct an experimental evaluation of our systems using the test data of the DCASE 2024 Challenge Task 2 development dataset [14], [15]. The results show that all of the proposed techniques are effective and our submitted systems significantly outperform the official baseline system and the previous state-of-the-art system. Specifically, the official baseline system [16], the previous state-of-the-art system [9], and our system has achieved 55.45%, 63.62%, and 67.26% in official scores, respectively.

2. STATE-OF-THE-ART METHOD IN THE 2023 CHALLENGE SETTING

We describe the state-of-the-art discriminative method in the 2023 Challenge setting we have been able to confirm performance [9], [11], [12].

2.1. Structure of feature extractor

The feature extractor receives an amplitude spectrum and an amplitude spectrogram of an input audio signal [12]. Since static information is captured by the spectrum, temporal mean normalization (TMN), which subtracts the temporal mean, is applied to the spectrogram to capture dynamic information. The each network extracts D-dimensional feature from each input and obtains $\boldsymbol{z}_i^{\mathrm{cat}}$ as a final output for the i-th audio signal $\boldsymbol{x}_i \in \mathbb{R}^T$ as follows:

$$\boldsymbol{z}_i^{\mathrm{cat}} = [\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)}] \in \mathbb{R}^{2D}, \tag{1}$$

where T is the length of the time-domain audio signal and $\boldsymbol{z}_i^{(m)} = g_m(f_m(\boldsymbol{x}_i))$. $f_1(\cdot)$ and $f_2(\cdot)$ apply discrete Fourier transform (DFT) and short-time Fourier transform (STFT) to the audios signal, respectively. $g_1(\cdot)$ and $g_2(\cdot)$ are neural networks carefully designed to prevent trivial projection (i.e., no bounded activation function and no bias term) [12], [17]. The effectiveness of these techniques has been shown in [12].

2.2. Training method of feature extractor

2.2.1. Sub-cluster AdaCos

As a classification loss function, angular margin loss, which minimizes the cosine distance between the extracted feature and the corresponding class center, is widely used in ASD task [3], [4], [9]. The sub-cluster AdaCos (SCAC) [11] is its improved version and achieves high ASD performance by using multiple class centers for a single class label. SCAC uses fixed class centers to prevent trivial projection [12].

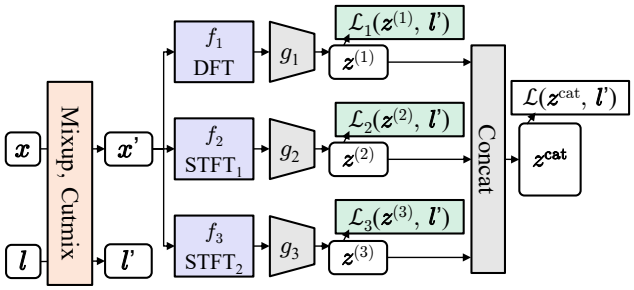

Figure 1: Structure and training method for our feature extractor

2.2.2. Mixup

In the ASD task, data augmentation technique, mixup [18] is widely used and its effectiveness has been shown [3], [9], [10], [12]. Mixup linearly interpolates audio signals and class labels between randomly selected two training samples, respectively, as follows:

$$\boldsymbol{x}'_i = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j, \tag{2}$$

$$\boldsymbol{l}'_i = \lambda \boldsymbol{l}_i + (1-\lambda)\boldsymbol{l}_j, \tag{3}$$

where $\boldsymbol{l}_i \in \{0,1\}^C$ is a onehot label for $\boldsymbol{x}_i$, C is the number of label classes, and $\lambda \in [0,1]$ is a mixing coefficient. For general classification tasks, data augmentation techniques such as mixup improve performance by preventing overfitting to specific training data. In the ASD task, $\boldsymbol{x}'_i$ can also be interpreted as being used as a pseudo-anomalous sound since $\boldsymbol{l}'_i$ increases the distance from the class center of normal data.

2.2.3. FeatEx

Recently, a new training method FeatEx [9] has been proposed that significantly improves the ASD performance. FeatEx trains a feature extractor with the following loss function.

$$\mathcal{L}_{\mathrm{cat}}(\boldsymbol{z}_i^{\mathrm{cat}}, \boldsymbol{l}'_i) + \mathcal{L}_{\mathrm{ex}}(\boldsymbol{z}_i^{\mathrm{ex}}, \boldsymbol{l}_i^{\mathrm{ex}}), \tag{4}$$

where $\mathcal{L}_{\mathrm{cat}}(\cdot,\cdot)$ is an originally used discriminative loss function with the fixed class centers and FeatEx additionally uses $\mathcal{L}_{\mathrm{ex}}(\boldsymbol{z}_i^{\mathrm{ex}}, \boldsymbol{l}_i^{\mathrm{ex}})$ with trainable class centers. $\boldsymbol{z}_i^{\mathrm{ex}}$ and $\boldsymbol{l}_i^{\mathrm{ex}}$ are calculated using randomly selected i-th and j-th samples as follows:

$$\boldsymbol{z}_i^{\mathrm{ex}} = [\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(2)}] \in \mathbb{R}^{2D}, \tag{5}$$

$$\boldsymbol{l}_i^{\mathrm{ex}} = [\boldsymbol{0}, 0.5 \cdot \boldsymbol{l}'_i, 0.5 \cdot \boldsymbol{l}'_j] \in [0,1]^{3C}, \tag{6}$$

where $\boldsymbol{0}$ is a C-dimensional zero vector. With the additional loss $\mathcal{L}_{\mathrm{ex}}(\boldsymbol{z}_i^{\mathrm{ex}}, \boldsymbol{l}_i^{\mathrm{ex}})$, the feature extractor needs to identify whether the information from the different audio signal is mixed or not, resulting more information being captured [9].

2.3. Backend

The backend responsible for calculating the anomaly score is constructed using features $\boldsymbol{z}^{\mathrm{cat}}$ extracted from the training data. As a preliminary step, k-means clustering is applied to the features of the source domain. The anomaly score is then determined by the smallest distance between the cluster centers of the source domain and all features of the target domain.

3. PROPOSED METHOD

We use the system described in Sec. 2 as our baseline system and propose various techniques to improve it.

3.1. Feature extractor

To improve the performance regardless of the requirement of the 2024 Challenge (i.e., lack of the attribute labels), we employ three techniques for the feature extractor as shown in Fig. 1.

3.1.1. Multi-resolution spectrograms

We extend the conventional method by adding an amplitude spectrogram of the different resolution $f_3(\boldsymbol{x}_i)$ to the input features. We expect that it gives multiple perspectives to capture anomalies.

3.1.2. Subspace loss

We propose a subspace loss that achieves the same performance improvement effect as FeatEx in a simpler way. First, $\mathcal{L}_{\mathrm{ex}}(\cdot,\cdot)$ in the FeatEx can be interpreted as it encourages each network $g_m(\cdot)$ to identify the class labels from only the corresponding input features $\boldsymbol{z}_i^{(m)} = g_m(f_m(\boldsymbol{x}_i))$ without combined feature $\boldsymbol{z}_i^{\mathrm{cat}}$. We then use the following loss function, replacing $\mathcal{L}_{\mathrm{ex}}(\cdot,\cdot)$ with the additional subspace loss functions $\mathcal{L}_m(\cdot,\cdot)$.

$$\mathcal{L}_{\mathrm{cat}}(\boldsymbol{z}_i^{\mathrm{cat}}, \boldsymbol{l}'_i) + \sum_{m=1}^{M} \mathcal{L}_m(\boldsymbol{z}_i^{(m)}, \boldsymbol{l}'_i), \tag{7}$$

where $M = 3$ is the number of input features and $\mathcal{L}_m(\cdot,\cdot)$ has the trainable centers. The subspace loss is more parameter-efficient than FeatEx with respect to the number of input features M. For the trainable class center, FeatEx requires $DCSM(M+1)$ parameters whereas subspace loss requires only $DCSM$ parameters where S is the number of sub-clusters in SCAC. In this respect, the subspace loss is well-suited for using multi-resolution spectrograms. A detailed analysis of the subspace loss is our future work.

3.1.3. Cutmix

We newly adopt cutmix [19] as a data augmentation technique. Cutmix is used in the image classification tasks and it replaces the linear interpolation in mixup with the exchange of image patches. We apply cutmix to the time-domain signal as follows:

$$\boldsymbol{x}'_i = \boldsymbol{m}^{(\lambda)} \odot \boldsymbol{x}_i + (1 - \boldsymbol{m}^{(\lambda)}) \odot \boldsymbol{x}_j, \tag{8}$$

$$\boldsymbol{l}'_i = \lambda \boldsymbol{l}_i + (1-\lambda)\boldsymbol{l}_j, \tag{9}$$

where $\boldsymbol{m}^{(\lambda)} \in \{0,1\}^T$ is a binary mask and $\boldsymbol{m}^{(\lambda)}$ set a consecutive λT-sample segment to 1. While mixup can be interpreted as a process that generates pseudo anomalies in the entire signal, cutmix generates pseudo anomalies only in a certain segment. Because they are expected to have different effects, we use both of them.

3.2. Pseudo-attribute labels

To improve the performance in the 2024 Challenge setting, we introduce pseudo-attribute labels for the machine without ground truth attribute labels. The proposed method consists of three stages: (1) constructing a feature space that reflects the differences in the machine sounds, (2) generating pseudo-attribute labels by clustering in the feature space, and (3) training the feature extractor for ASD using the obtained pseudo-attribute labels.

3.2.1. Construction of feature space

First, we construct the feature space that reflects the difference in the machine sounds. In the preliminary experiments, we found that large-scale pre-trained models are not useful for constructing such feature space because they tend to reflect the difference in the noise

Algorithm 1: Proposed bottom-up ensemble clustering

Input: Set of amplitude spectrograms of the target machine $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$, two pre-trained feature extractor $g^{(1)}_{\mathrm{attr}}$ and $g^{(2)}_{\mathrm{attr}}$ with different seed, the maximum number of clusters k, and threshold θ.
Output: Result of clustering and corresponding labels
Function `kmeans` ($\mathcal{Z}$, k) **:**
  Obtain k clusters $\mathcal{C} = \{\mathcal{C}_i \mid i = 1, \ldots, k\}$ by applying k-means clustering to features $\mathcal{Z}$.
  return $\mathcal{C}$
Function `merge` ($\mathcal{C}$, L) **:**
  Find the pair of clusters $(\mathcal{C}_{i_1}, \mathcal{C}_{i_2})$ in $\mathcal{C}$ with the smallest distance between centroids.
  Remove clusters $\mathcal{C}_{i_1}$ and $\mathcal{C}_{i_2}$ from $\mathcal{C}$.
  Add $\mathcal{C}_{\mathrm{new}} = \mathcal{C}_{i_1} \cup \mathcal{C}_{i_2}$ to $\mathcal{C}$.
  Remove labels L_{i_1} and L_{i_2} from L.
  Add $L_{\mathrm{new}} = L_{i_1} \cup L_{i_2}$ to L.
  return $\mathcal{C}$, L, $\mathcal{C}_{\mathrm{new}}$, L_{new}
Function `assign` ($\mathcal{C}^{\mathrm{tgt}}$, L^{tgt}, $\mathcal{C}^{\mathrm{ref}}_{\mathrm{new}}$, $L^{\mathrm{ref}}_{\mathrm{new}}$) **:**
  for $i = 1, k$ **do**
    if L^{tgt}_i *is* $\emptyset$ *and* $|\mathcal{C}^{\mathrm{tgt}}_i \cap \mathcal{C}^{\mathrm{ref}}_{\mathrm{new}}| / |\mathcal{C}^{\mathrm{tgt}}_i| > \theta$ **then**
      $L^{\mathrm{tgt}}_i \leftarrow L^{\mathrm{ref}}_{\mathrm{new}}$
    end
  end
  return $\mathcal{C}^{\mathrm{tgt}}$, L^{tgt}
$\mathcal{C}^{\mathrm{ref}} =$ `kmeans` $(\{g^{(1)}_{\mathrm{attr}}(\boldsymbol{X}_i) \mid i = 1, \ldots, n\}, k)$
Initialize cluster labels: $L^{\mathrm{ref}}_i \leftarrow \{i\}$ for $i = 1$ to k
$\mathcal{C}^{\mathrm{tgt}} =$ `kmeans` $(\{g^{(2)}_{\mathrm{attr}}(\boldsymbol{X}_i) \mid i = 1, \ldots, n\}, k)$
Initialize cluster labels: $L^{\mathrm{tgt}}_i \leftarrow \emptyset$ for $i = 1$ to k
for $i = 1, k$ **do**
  $\mathcal{C}^{\mathrm{tgt}}, L^{\mathrm{tgt}} \leftarrow$ `assign` $(\mathcal{C}^{\mathrm{tgt}}, L^{\mathrm{tgt}}, \mathcal{C}^{\mathrm{ref}}_i, L^{\mathrm{ref}}_i)$
end
for $i = 1, k-1$ **do**
  $\mathcal{C}^{\mathrm{ref}}, L^{\mathrm{ref}}, \mathcal{C}^{\mathrm{ref}}_{\mathrm{new}}, L^{\mathrm{ref}}_{\mathrm{new}} \leftarrow$ `merge`$(\mathcal{C}^{\mathrm{ref}}, L^{\mathrm{ref}})$
  $\mathcal{C}^{\mathrm{tgt}}, L^{\mathrm{tgt}} \leftarrow$ `assign` $(\mathcal{C}^{\mathrm{tgt}}, L^{\mathrm{tgt}}, \mathcal{C}^{\mathrm{ref}}_{\mathrm{new}}, L^{\mathrm{ref}}_{\mathrm{new}})$
end
return $\mathcal{C}^{\mathrm{tgt}}$ and L^{tgt}

rather than the difference in the operating sound. This problem motivates us to construct the noise-robust feature space from scratch. The proposed method trains a feature extractor $g_{\mathrm{attr}}(\cdot)$ for each machine type with the following $\mathcal{L}_{\mathrm{triplet}}$.

$$\mathcal{L}_{\mathrm{triplet}} = \max(d_{\mathrm{pull}} - d_{\mathrm{push}} + d_{\mathrm{margin}}, 0), \tag{10}$$

$$d_{\mathrm{pull}} = ||g_{\mathrm{attr}}(\boldsymbol{X}_{\mathrm{anchor}}) - g_{\mathrm{attr}}(\boldsymbol{X}_{\mathrm{positive}})||_2, \tag{11}$$

$$d_{\mathrm{push}} = ||g_{\mathrm{attr}}(\boldsymbol{X}_{\mathrm{anchor}}) - g_{\mathrm{attr}}(\boldsymbol{X}_{\mathrm{negative}})||_2, \tag{12}$$

where d_{margin} is hyperparameter for a margin. $\boldsymbol{X}_{\mathrm{anchor}}$, $\boldsymbol{X}_{\mathrm{positive}}$, and $\boldsymbol{X}_{\mathrm{negative}}$ are obtained as follows:

$$\boldsymbol{X}_{\mathrm{anchor}} \sim \{\boldsymbol{X}_i, \mathrm{Noise}(\boldsymbol{X}_i)\} \quad \text{with equal probability}, \tag{13}$$

$$\boldsymbol{X}_{\mathrm{positive}} = \mathrm{Noise}(\boldsymbol{X}_i), \tag{14}$$

$$\boldsymbol{X}_{\mathrm{negative}} \sim \{\mathrm{Resize}(\boldsymbol{X}_i), \mathrm{Noise}(\mathrm{Resize}(\boldsymbol{X}_i)), \boldsymbol{X}_j, \mathrm{Noise}(\boldsymbol{X}_j)\} \quad \text{with equal probability}, \tag{15}$$

where $\boldsymbol{X}_i = f_{\mathrm{attr}}(\boldsymbol{x}_i)$ and $\boldsymbol{X}_j$ are amplitude spectrograms of different signals from the same target machine type, $\mathrm{Noise}(\cdot)$ adds the sound of non-target machine type, and $\mathrm{Resize}(\cdot)$ resizes the spectrogram in both the time and frequency axes. Minimizing d_{pull} helps to reduce the effect of noise and maximizing d_{push} reflects

Algorithm 2: Proposed core-set selection

Input: Original data $\mathcal{D}^{\mathrm{org}}$, external data $\mathcal{D}^{\mathrm{ext}}$, pre-trained ASD system $h(\cdot)$, percentages for the percentiles q, and maximum number of samples $n_{\max}$
Output: core-set $\mathcal{D}^{\mathrm{core}}$ for $\mathcal{D}^{\mathrm{ext}}$
Calculate anomaly score $\boldsymbol{a}^{\mathrm{org}}$ of samples in $\mathcal{D}^{\mathrm{org}}$ with $h(\cdot)$
Obtain the q-th percentile value a^{th} of $\boldsymbol{a}^{\mathrm{org}}$
$\mathcal{D}^{\mathrm{core}} \leftarrow \emptyset$
$\boldsymbol{x} \leftarrow \arg\min_{\boldsymbol{x} \in \mathcal{D}^{\mathrm{ext}}} h(\boldsymbol{x})$
while $|\mathcal{D}^{\mathrm{core}}| \leq n_{\max}$ *and* $h(\boldsymbol{x}) < a^{\mathrm{th}}$ **do**
  $\mathcal{D}^{\mathrm{core}} \leftarrow \mathcal{D}^{\mathrm{core}} \cup \{\boldsymbol{x}\}$
  $\mathcal{D}^{\mathrm{ext}} \leftarrow \mathcal{D}^{\mathrm{ext}} \setminus \{\boldsymbol{x}\}$
  $\boldsymbol{x} \leftarrow \arg\min_{\boldsymbol{x} \in \mathcal{D}^{\mathrm{ext}}} h(\boldsymbol{x})$
end

the difference in the machine sound regardless of noise. The feature extractor $g_{\mathrm{attr}}(\cdot)$ is used solely for obtaining pseudo-attribute labels and is different from $g_m(\cdot)$.

3.2.2. Obtaining pseudo-attribute labels by clustering

After training $g_{\mathrm{attr}}(\cdot)$ for each machine type, we extract features from the training data and obtain pseudo-attribute labels through clustering. However, it is generally difficult to obtain proper labels without correctly setting the number of clusters. To solve this problem, we propose a bottom-up ensemble clustering as summarized in Algorithm 1. The algorithm assigns a label of $\mathcal{C}^{\mathrm{ref}}$ to $\mathcal{C}^{\mathrm{tgt}}$ when the appropriate granularity is reached, performing bottom-up clustering on $\mathcal{C}^{\mathrm{ref}}$. The appropriate granularity is determined using two feature spaces in an ensemble manner (i.e., $|\mathcal{C}^{\mathrm{tgt}}_i \cap \mathcal{C}^{\mathrm{ref}}_j| / |\mathcal{C}^{\mathrm{tgt}}_i| > \theta$). The algorithm outputs $\mathcal{C}^{\mathrm{tgt}}$ and L^{tgt} and provides a set of labels for each audio signal because the labels are merged through bottom-up clustering.

3.2.3. Training with pseudo-attribute labels

We train a feature extractor for ASD with L^{tgt}. Since L^{tgt} contains multiple labels for each training sample, we replace the original loss function $\mathcal{L}_{\mathrm{org}}(\cdot, \cdot)$ with the following loss function $\mathcal{L}_{\mathrm{multi}}(\cdot, \cdot)$.

$$\mathcal{L}_{\mathrm{multi}}(\boldsymbol{z}_i, L^{\mathrm{tgt}}_i) = \min_{l_{i,j} \in L^{\mathrm{tgt}}_i} \mathcal{L}_{\mathrm{org}}(\boldsymbol{z}_i, \mathrm{onehot}(l_{i,j})), \tag{16}$$

where L^{tgt}_i is a i-th set of labels and $\mathrm{onehot}(\cdot)$ transforms label index to onehot label.

3.3. Core-set selection from the external dataset for ASD

The variety of the training data for each machine type has been restricted since the 2023 Challenge. While leveraging external data resources could potentially alleviate this limitation, it is crucial to carefully select useful samples to avoid a data imbalance between original and external datasets. Therefore, we propose a core-set selection method for ASD as summarized in Algorithm 2. The algorithm selects audio signals with low anomaly scores that can be interpreted as misclassified samples between $\mathcal{D}^{\mathrm{org}}$ and $\mathcal{D}^{\mathrm{ext}}$. We use Audioset [13] as $\mathcal{D}^{\mathrm{ext}}$ and generate its label by concatenating a class label of Audioset (i.e., mid) and a machine type predicted by discriminative feature extractor of h. In the experiment, only the following classes of Audioset were manually selected as $\mathcal{D}^{\mathrm{ext}}$: Vehicle, Motorboat, Ship, Power windows, Skidding, Air brake, Propeller, Helicopter, Engine, Dental drill, Chainsaw, Medium engine, Heavy engine, Accelerating, Sliding door, Microwave oven, Hair

Table 1: Evaluation results. The values represent the harmonic mean of AUC and pAUC over all domains. "official" represents the official score obtained by harmonic mean over all machine types. * indicates a machine type with no ground truth attribute label. MR, SL, PA, and AS indicate the use of Multi-Resolution spectrograms, Subspace Loss, Pseudo-Attribute labels, and Audioset, respectively. Options in the parentheses are used for the ensemble of anomaly scores.

ID	System	bearing	fan	gearbox*	slider*	ToyCar	ToyTrain*	valve	official
	Official baseline (MSE)	60.26	59.70	64.38	57.47	46.11	54.33	49.77	55.35
	Official baseline (Mahalanobis)	54.78	54.81	**68.79**	62.06	48.20	48.48	53.46	55.02
	FeatEx+Mixup	62.51	59.02	62.74	83.77	**54.85**	59.76	70.17	63.62
	MR+SL+Mixup	**71.83**	57.29	60.45	81.00	54.43	61.52	72.64	64.42
	MR+SL+Cutmix	69.21	57.94	63.48	77.49	53.23	61.96	72.83	64.21
	MR+SL+(Mixup, Cutmix)	71.66	57.69	62.47	80.02	54.21	61.69	72.93	64.72
④	MR+SL+(Mixup, Cutmix)+PA	69.71	57.90	66.53	90.08	52.19	**69.40**	75.23	66.91
②	MR+SL+(Mixup, Cutmix)+AS	70.16	**60.78**	64.26	79.97	53.52	61.40	73.09	65.16
③	MR+SL+(Mixup, Cutmix)+PA+AS	70.12	60.77	64.31	**91.11**	51.98	69.35	**75.88**	**67.26**
①	(②, ③, ④)	70.17	60.07	65.20	88.50	52.47	69.22	74.90	67.05

dryer, Electric toothbrush, Vacuum cleaner, Electric shaver, Gears, Pulleys, Mechanical fan, Air conditioning, Printer, Sawing, Filling, and Sanding.

4. EXPERIMENTAL EVALUATIONS

4.1. Experimental setups

We conducted an experimental evaluation using the DCASE 2024 Task 2 Challenge development dataset (ToyADMOS2 [14], MIMII DG [15]) and additional training dataset. The development dataset included training and test data of seven machine types: bearing, fan, gearbox, valve, slider, ToyCar, and ToyTrain. Also, the additional training datasets included training data of the other nine machine types. The training data included 1,000 samples of normal data for each machine type, of which 990 samples are in the source domain and 10 samples are in the target domain. The test data of the development dataset included 50 samples per machine type, domain, and normal/anomalous category. Each recording was a 6 to 12-second single channel signal sampled at 16 kHz.

For the STFT parameters, the DFT size of $f_2(\cdot)$, $f_3(\cdot)$, and $f_{\mathrm{attr}}(\cdot)$ were 4096, 128, and 1024, respectively. The window function was the hann window, with the same size as the DFT size. The frame shift was half of the DFT size, and frequency bins in the range of 200 Hz to 8000 Hz were used. Also, we applied TMN to the $f_{\mathrm{attr}}(\cdot)$. The signal-to-noise ratio of $\mathrm{Noise}(\cdot)$ was randomly selected from $[-5, 5)$. The scale of $\mathrm{Resize}(\cdot)$ was randomly selected from $[0.5, 0.8)$ or $[1.2, 1.5)$. For the hyperparameters of bottom-up ensemble clustering described in Sec. 3.2.2, we set k to 16 and θ to 0.95, respectively. For the hyperparameters of core-set selection described in Sec. 3.3, we set q to 100 (i.e., maximum anomaly score) and $n_{\max}$ to 100, respectively. We trained the feature extractors $g_1(\cdot)$, $g_2(\cdot)$, and $g_3(\cdot)$ for 18 epochs, and $g_{\mathrm{attr}}(\cdot)$ for 6 epochs, with AdamW [20] of a fixed learning rate of 0.001, respectively. Each feature extractor consisted of the ResNet architecture similar to that in [9]. The mixup was applied with 50% probability and the cutmix was applied with 75% probability. The number of sub-clusters S in SCAC was set to 16. Additionally, we fixed the scale parameter of SCAC since not applying mixup with 100% probability could cause overflow. The batch size was 64 for $g_1(\cdot)$, $g_2(\cdot)$, and $g_3(\cdot)$, and 32 for $g_{\mathrm{attr}}(\cdot)$. During inference, we used the conventional backend described in Sec. 2.3. We averaged the anomaly scores for each audio signal using scores obtained from the checkpoints of 14, 16, and 18 epochs, and five different seeds (a total of 15 scores).

As the metrics, we used the area under the receiver operating characteristic (ROC) curve (AUC) and partial AUC (pAUC) with $p = 0.1$. As in the official evaluation, we calculated the AUC of each domain using the normal sounds in that domain and the anomalous sounds from both domains, and we calculated pAUC using sounds in both domains.

4.2. Experimental results

Table 1 shows the performance of the official baseline, FeatEx (our baseline), and our systems. For the pre-trained system h in the core-set selection, we used the MR+SL+Mixup+PA with an ensemble of anomaly scores across five different seeds at the 18th epoch. First, we can see that FeatEx outperforms the official baseline system, and the proposed combination of multi-resolution spectrograms and subspace loss outperforms FeatEx. Additionally, the ensemble of a system using mixup and a system using cutmix achieves slightly better performance than each system individually. The use of pseudo-attribute labels significantly improves the performance of the machine type with no ground truth attribute labels. Training with Audioset further enhances performance, achieving 67.26% in the official scores.

5. CONCLUSION

In this report, we presented our systems for DCASE 2024 Challenge Task 2. Our systems are based on the state-of-the-art discriminative ASD method, and we further improved performance with three techniques. First, we enhanced the feature extractor using multi-resolution spectrograms, subspace loss, and cutmix. Second, we introduced pseudo-attribute labels and proposed bottom-up ensemble clustering to obtain proper labels. Third, we utilized Audioset as an external data resource for the training and proposed a core-set selection method for finding useful samples. The experimental evaluation using the development dataset demonstrated the effectiveness of the proposed techniques and our system achieved 67.26% in the official score.

6. ACKNOWLEDGMENT

This paper was partly supported by a project, JPNP20006, commissioned by NEDO, and JSPS KAKENHI Grant Number JP20H00102.

7. REFERENCES

[1] T. Nishida, N. Harada, D. Niizumi, *et al.*, "Description and discussion on DCASE 2024 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," *In arXiv e-prints: 2406.07250*, 2024.

[2] K. Dohi, K. Imoto, N. Harada, *et al.*, "Description and discussion on DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," *In arXiv e-prints: 2305.07828*, 2023.

[3] J. Jie, "Anomalous sound detection based on self-supervised learning," DCASE2023 Challenge, Tech. Rep., 2023.

[4] Z. Lv, B. Han, Z. Chen, Y. Qian, J. Ding, and J. Liu, "Unsupervised anomalous detection based on unsupervised pretrained models," DCASE2023 Challenge, Tech. Rep., 2023.

[5] A. Jiang, Q. Hou, J. Liu, *et al.*, "Thuee system for first-shot unsupervised anomalous sound detection for machine condition monitoring," DCASE2023 Challenge, Tech. Rep., 2023.

[6] K. Wilkinghoff, "Fraunhofer fkie submission for task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," DCASE2023 Challenge, Tech. Rep., 2023.

[7] Y. Zhou and Y. Long, "Attribute classifier with imbalance compensation for anomalous sound detection," DCASE2023 Challenge, Tech. Rep., 2023.

[8] T. Fujimura, K. Imoto, and T. Toda, "Discriminative neighborhood smoothing for generative anomalous sound detection," *arXiv preprint arXiv:2403.11508*, 2024.

[9] K. Wilkinghoff, "Self-supervised learning for anomalous sound detection," in *Proc. ICASSP*, 2024, pp. 276–280.

[10] I. Kuroyanagi, T. Hayashi, K. Takeda, and T. Toda, "Improvement of serial approach to anomalous sound detection by incorporating two binary cross-entropies for outlier exposure," in *Proc. EUSIPCO*, 2022, pp. 294–298.

[11] K. Wilkinghoff, "Sub-cluster adacos: Learning representations for anomalous sound detection," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[12] K. Wilkinghoff, "Design choices for learning embeddings from auxiliary tasks for domain generalization in anomalous sound detection," in *Proc. ICASSP*, 2023, pp. 1–5.

[13] J. F. Gemmeke, D. P. Ellis, D. Freedman, *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017, pp. 776–780.

[14] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *Proc. DCASE*, 2021, pp. 1–5.

[15] K. Dohi, T. Nishida, H. Purohit, *et al.*, "Mimii dg: Sound dataset for malfunctioning industrial machine investigation and inspection for domain ggeneralization task," in *Proc. DCASE*, 2022.

[16] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "First-shot anomaly detection for machine condition monitoring: A domain generalization baseline," *Proceedings of 31st European Signal Processing Conference (EUSIPCO)*, pp. 191–195, 2023.

[17] L. Ruff, R. Vandermeulen, N. Goernitz, *et al.*, "Deep one-class classification," in *International conference on machine learning*, 2018, pp. 4393–4402.

[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2018.

[19] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[20] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.