ANOMALOUS SOUND DETECTION USING CNN-BASED MODELS AND ENSEMBLE

Technical Report

Ryosuke Tanaka, Keisuke Ikeda, Shiiya Aoyama, Satoshi Tamura

Gifu University
Faculty of Engineering Yanagido 1-1, Gifu, Gifu 5011193, Japan
ryosuke@asr.info.gifu-u.ac.jp

ABSTRACT

This paper presents our efforts for DCASE2023 Challenge Task2. We explore three schemes: (1) sound anomaly detection based on state-of-the-art image processing techniques with machine type classifiers, (2) anomaloous detection based on the same image processing in addition to the inpainting strategy, (3) anomaly detection utilizing machine setting classification to enhance the performance, and (4) anomaly detection by composing existing detectors in the ensemble manner. Experiments were conducted to evaluate our approaches.

Index Terms— sound spectrogram, convolutional neural network, masking, inpainting, classification, ensemble.

1. INTRODUCTION

Anomaly detection is a technique to detect abnormal data, using statistics, machine learning and deep-learning technology. Since there are high demands to predict or detect any failure in industrial fields, many researchers have devoted their efforts to accomplish a high-performance anomaly detection technique. Nowadays most anomaly detection schemes have employed deep learning. Although we have succeeded to build good detection methods, there are still several issues in this field; we need to adapt the prepared detector to any circumstance different from the one in which the detection model was trained; In terms of availability, it is also expected to build a detector for new machines using existing detectors or data for different machines.

From these above standpoints, in this paper we propose anomaly detection techniques for DCASE2023 Task2 [1]. We investigate following approaches; (1) we apply state-of-the-art anomaly detection schemes developed for image processing and computer vision fields to sound spectrograms, incorporating machine type classification to improve the detection model, (2) we utilize the same image processing as (1), not with the clssification but with the inpainting scheme that is often used in the computer vision field, (3) we involve another classifier, that is to predict the machine setting, to enhance anomaly detection performance, and (4) we compose existing anomaly detection models for a new machine type, just like an ensemble method.

We tested these methods using the DCASE2023 development dataset. In this paper we also report the experimental results. Among the above schemes, (1), (2) and (3) were evaluated respectively. After that, we also investigated the performance based on (4).

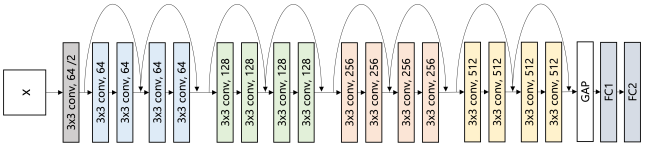

Figure 1: Feature extraction from a sound spectrogram image.

2. METHODOLOGY

We propose several approaches in this paper. Given audio data, sound spectrograms are firstly obtained. We then explore some strategies: image-based anomaly detection and combination of other classifiers. Regarding the classifiers, we investigate machine type classification and machine setting classification. In addition, we apply an ensemble-based integration using anomaly detectors developed above.

2.1. Sound anomaly detection using image processing

We firstly introduce an anomaly detection method based on image processing. An image corresponding to given audio data can be easily acquired. In addition, a lot of deep-learning-based image processing techniques have been proposed in many tasks, including anomaly detection. Thus, we believe that it is reasonable to employ image processing to enhance the anomaly detection performance and robustness.

2.1.1. Imaging

First of all, the same acoustic processing in the baseline is carried out, followed by converting audio waveform into mel-spectrogram; a 128×128 image is obtained. Note that standardization is performed using mean and variance of the training data, not only to training but also to testing data.

2.1.2. Anomaly detection

In the DCASE2022 sound anomaly detection task, it was shown that classification using sections and machine types is effective [1]. There is only one section in this task for one machine, however, machine type classification must be still useful. In our approach, a ResNet18-based model [2] is chosen as a feature extractor, followed by another model consisting of two linear layers and nonlinear activation functions as a feature transformer. Figure 1 illustrates the feature computation model. We build the model from scratch, using the DCASE2023 data. The conventional cross-entropy loss

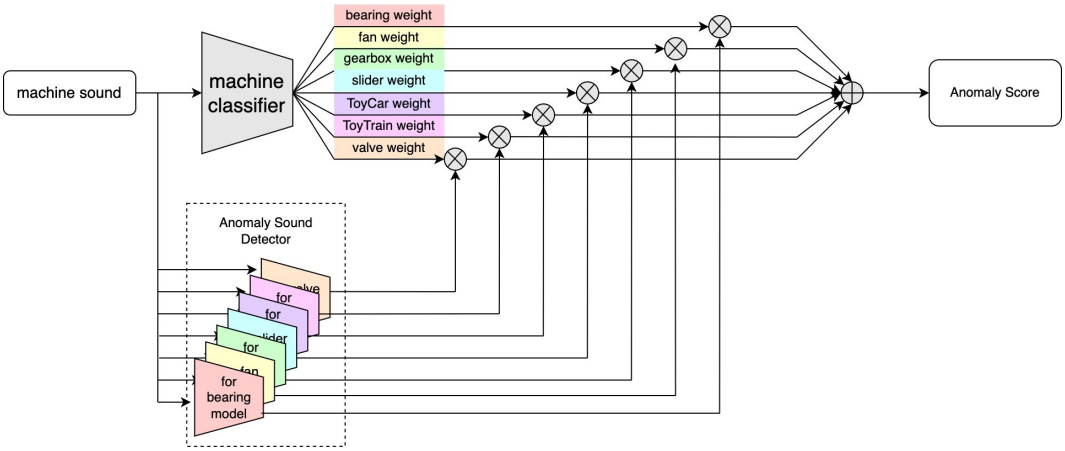

Figure 2: An ensemble scheme using existing anomaly detectors and machine classifier.

is then adopted for model training, while several data augmentation techniques are applied, such as time masking and frequency masking which are based on image processing, in addition to Gaussian noise overlapping. Anomaly detection is finally carried out; a Mahalanobis distance is calculated for a given feature vector using mean vector and covariance matrix from normal data in the training data set of particular machine type.

2.1.3. *Inpainting*

InTra [3] was proposed as one of image anomaly detection schemes. The method firstly splits a given image into patches, followed by masking one patch. After that, a model tries to reconstruct the masked patch according to the remaining patches. The original Intra model composed a Transformer model [4], while we employed a ResNet18-based model shown in Figure 1 instead. The model has a Convolutional Neural Network (CNN) architecture. In addition, the reconstruction model consists of nonlinear layers.

In our approach, we apply a 128×2 mask that covers all the frequency bins in two time frames. We randomly put the mask in all the training images. As data augmentation, we only apply Gaussian noise. When testing, we firstly create 64 masked images having different mask positions from a given image. Next, we compute reconstruction errors for all the masked images. The average is finally obtained as an anomaly score.

2.2. Anomaly detection utilizing machine setting classification

As discussed above, employing classification tasks simultaneously is useful when building anomaly detection models. In spite that the DCASE2023 data set has only one section, we found several machine settings for some machine types. Therefore, we try to introduce a model predicting the setting type; in addition, we generate simulated data from the original ones assuming different machine settings such as speed or voltage.

2.2.1. *Preprocessing*

Similar to our first approach, we at first obtain a 128×128 sound spectrogram image. Before applying the imaging technique, we transform the original waveform assuming the higher or lower frequency based on different speed or voltage settings. We choose the rate from 0.8 to 1.3, and the rate selected is also used as a label for model training.

2.2.2. *Anomaly detection*

We adopt a CNN autoencoder based on VGG [5], in addition to one linear layer as a classifier. We put a 32×32 mask randomly on an input training image, and the model is built so as to reconstruct the masked part, and to predict the machine setting simultaneously. Given a testing image, the model computes an intermediate vector as an output of its encoder. We employ a Mahalanobis distance of the obtained vector as an anomaly score.

2.3. Anomaly detection composing existing models

When developing a deep-learning-based anomaly detection system, it is recommended to collect training data as much as possible. However in practice, it is sometimes hard and there are highly demands to overcome this issue. We think it might be possible to make the anomaly detection model, by composing existing anomaly detection models for the other machines, which are built using numerous training data.

2.3.1. *Model composing*

For the DCASE2023 Task2, there are seven machine types involved in the development set. In the additional training and testing datasets, we have the other seven machine types. In this study, we try to compose anomaly detection models for the development set to obtain a new model for one machine type in the former data sets.

First, we prepare anomaly detection schemes each for one machine type in the development set, to measure anomaly scores. Second, for a given sound data sample, we train a classifier to predict a machine type; the classifier outputs seven probabilities each which corresponds to one machine type, e.g. valve, ToyTrain, and so on. The outputs can be then regarded as similarity scores. Next, we put sound data of one machine type, e.g. Shaker, Grinder, and so on, in the additional training dataset into the classifier. By calculating mean values of output results, we can estimate how the target machine type is close to each machine type in the development set. Let us denote the weight for the i-th machine type by C_i, subject to their sum is 1.

Table 1: AUC [%] for source data in baseline and proposed methods.

Machine		Baseline1	Baseline2	Method1	Method2	Method3
ToyCar	sec00	70.10	**74.53**	56.08	61.40	48.00
ToyTrain	sec00	57.93	55.98	58.92	52.00	**61.52**
bearing	sec00	65.92	55.75	**69.64**	67.88	48.40
fan	sec00	80.19	**87.10**	47.74	44.60	45.56
gearbox	sec00	60.31	**71.88**	66.68	57.00	51.60
slider	sec00	70.31	84.02	**93.72**	61.56	53.04
valve	sec00	55.35	56.31	**62.20**	61.52	46.40

Table 2: AUC [%] for target data in baseline and proposed methods.

Machine		Baseline1	Baseline2	Method1	Method2	Method3
ToyCar	sec00	46.89	43.42	48.28	51.48	**54.68**
ToyTrain	sec00	57.02	42.45	59.08	**69.96**	48.48
bearing	sec00	55.75	55.28	57.44	**58.84**	53.20
fan	sec00	36.18	45.98	**53.68**	50.60	44.80
gearbox	sec00	60.69	**70.78**	66.04	60.92	54.44
slider	sec00	48.77	73.29	**96.72**	53.44	47.00
valve	sec00	50.69	51.40	**55.12**	33.00	41.00

Table 3: pAUC [%] in baseline and proposed methods.

Machine		Baseline1	Baseline2	method1	method2	method3
ToyCar	sec00	**52.47**	49.18	48.10	49.58	49.95
ToyTrain	sec00	48.57	48.13	49.26	50.11	**54.37**
bearing	sec00	50.42	51.37	**56.11**	54.42	49.00
fan	sec00	59.04	**59.33**	54.53	58.32	47.74
gearbox	sec00	53.22	54.34	**55.58**	51.58	53.74
slider	sec00	56.37	54.72	**81.16**	58.95	50.26
valve	sec00	51.18	51.08	**54.11**	50.11	50.26

Finally, by using the values and anomaly detectors mentioned above, we can now compute an anomaly score for a given sample x in the testing dataset as:

$$A_{composed}(x) = \sum_{i=1}^{7} C_i A_i(x) \tag{1}$$

where $A_i(x)$ represents an anomaly score obtained from the i-th machine type. Figure 2 depicts the overview diagram of the above process.

Note that, in this paper we use a pretrained ResNet50 model for ImageNet as the classifier, by modifying the last layer to adjust our task. We then updated all the parameters in the model using the DCASE2023 development dataset.

3. EXPERIMENT

We conducted experiments only using the development set to evaluate our methods explained in the last section. Regarding the last approach, it is impossible to apply the scheme as it is because we do not know the correct label for the test dataset, we estimated the performance in the development set instead; we chose two machine types as target ones, for each which an anomaly detector was composed using those for the rest five machine types.

Tables 1 and 2 show AUC results for source and target data, respectively. And Table 3 indicates pAUC results for all seven machine types in the development dataset. In these tables "Baseline1" and "Baseline2" indicate baseline systems provided by the task organizer; the former is the autoencoder-based approach, while the latter uses a Mahalanobis distance. In addition, "Method1" corresponds to a scheme using machine type classification with a Mahalanobis distance mentioned in Section 2.1, "Method2" indicates a method using inpainting also introduced in Section 2.1, and "Method3" is based on the machine settings classification method explained in Section 2.2.

Table 4: AUC [%] for source data in baseline and ensemble systems.

Machine		Baseline1	Baseline2	Ours
ToyTrain	sec00	57.93	55.98	34.36
valve	sec00	55.35	56.31	51.36

Table 5: AUC [%] for target data in baseline and ensemble systems.

Machine		Baseline1	Baseline2	Ours
ToyTrain	sec00	57.02	42.45	62.28
valve	sec00	50.69	51.40	44.76

Next, we evaluated AUC and pAUC using our ensemble-based strategy proposed in Section 2.3. In this case, we employed anomaly detectors based on the inpainting strategy. The target machine types are ToyTrain and valve, each of which anomalous detection was composed from those for the other five machines. Tables 4

Table 6: pAUC [%] in baseline and our ensemble systems.

Machine		Baseline1	Baseline2	Ours
ToyTrain	sec00	48.57	48.13	**50.32**
valve	sec00	51.18	51.08	**52.11**

and 5 represent AUC results for source and target data, respectively. Table 6 finally shows pAUC results.

4. REFERENCES

[1] Noboru Harada, et al. "First-shot anomaly detection for machine condition monitoring: a domain generalization baseline." *In arXiv, 2303.00455*, 2023.

[2] Kaiming He, et al. "Deep residual learning for image recognition." *In CVPR, pp.770-778*, 2016.

[3] Jonathan Pirnay, et al. "Inpainting transformer for anomaly detection." *In arXiv, 2104.13897*, 2021.

[4] Vaswani, Ashish, et al. "Attention is all you need." *In Advances in neural information processing systems, vol.30, pp.6000-6010*, 2017.

[5] Simonyan, K, Zisserman, A. "Very deep convolutional networks for large-scale image recognition." *In arXiv, 1409.1556*, 2014.