ROBUST ANOMALY SOUND DETECTION FRAMEWORK FOR MACHINE CONDITION MONITORING

Technical Report

Ying Zeng[1,3], *Hongqing Liu*[1], *Lihua Xue*[1], *Yi Zhou*[1], *Lu Gan*[2]

[1] School of Communication and Information Engineering,
Chongqing University of Posts and Telecommunications, Chongqing, China
[2] College of Engineering, Design and Physical Science, Brunel University, London UB8 3PH, U.K.
[3] AI Lab, Xiaomi Corporation, Beijing, China
{S200101158, S210131279}@stu.cqupt.edu.cn, {hongqingliu, zhouy}@cqupt.edu.cn
Lu.Gan@brunel.ac.uk, zengying1@xiaomi.com

ABSTRACT

This technical report describes our team's submission to DCASE 2022 Task 2. In this report, we propose a robust training framework for anomalous sound detection, which includes feature preprocessing, model pretraining, joint loss, and anomaly score selection. The experimental results show that our anomalous sound detection model outperforms the official model, with an average performance improvement of 22.08% based on the official scoring method.

Index Terms— Anomalous sound detection, feature preprocessing, robust detection

1. INTRODUCTION

In this task, the purpose of anomalous sound detection is to detect whether the sound emitted from the target machine type is abnormal. The issue is that there are no anomalous sound samples, and therefore, a binary classifier cannot be directly trained to determine whether it is anomalous. This task is often regarded as an unsupervised one. The basic idea is we need to learn the inherent properties of normal samples through normal sound samples. The common method is to train a classifier according to different machine types as auxiliary tasks [1, 2]. This is based on a realistic assumption that there are often more than one machine of the same kind in the factory. There are often some differences in the sounds emitted by different machine IDs. A classifier can be built to distinguish machine IDs. In the test phase, the probability confidence is used as the anomaly score, and the probability is passed through a manually set function to make a larger probability output a smaller anomaly score.

This approach worked well in last year's competition [3], but it did not perform well for certain machine types. Note that it is very easy to directly train a classifier to distinguish between different IDs of the same machine type using a neural network, which means that the model is often overfitted. To address this issue, we propose a robust anomalous sound detection framework. First, we use all machine types to train a classifier that can simultaneously distinguish different machine types with different IDs. Next, we fine-tune the model parameters to train a dedicated classifier for each machine type. In the testing phase, negative log-likelihood, Mahalanobis distance, and cosine similarity are optional as outlier scores.

Table 1: Model architecture, where k is the number of section IDs, t indicates the expansion factor, c is the output channels, n denotes the number of Inverted residuals blocks, and s is the stride. The first layer of each sequence has a stride s and others use stride 1.

Operator	t	c	n	s
Conv2d 3x3	-	64	-	2
Conv2d 3x3	-	64	-	1
Blockneck	2	128	2	2
Blockneck	4	128	2	2
Blockneck	4	128	2	2
Conv2d 1x1	-	512	-	1
Linear GDConv2d	-	512	-	1
Linear Conv2d 1x1	-	128	-	1
Dropout	-	-	-	-
Linear	-	k	-	-

2. PROPOSED METHOD

2.1. Model

During experiments, we found that using classic models in the image classification domain such as Inception [4] and Xception [5], did not actually improve detection performance. In this work, we train our anomalous sound detection model based on mobilefacenent [6] with our modifications to better fit this task, including reducing the depth and adding a linear branch in bottleneck that does not use any activation function. The model structure is provided in Table 1.

2.2. Anomaly score

The calculation of anomaly scores is very important. We found that the results obtained by using different calculation anomaly scores for the same model are often very different. This indicates that the model might be good enough, but we only need to select appropriate anomaly scores for different machine types.

In addition to the calculation of anomaly scores according to the model output probability proposed by Baseline. We also choose Mahalanobis distance [7] and cosine similarity as options for anomaly scores.

Table 2: Experiment configurations, where m represents Mahalanobis distance, p represents probability confidence.

	ToyCar	ToyTrain	Bearing	Fan	Gearbox	Slider	Valve
w/o Centerloss	True	True	True	False	True	False	False
Anomaly score	m	m	p	p	m	p	m
Highpass filter	True	True	False	False	True	False	True

Table 3: Anomaly detection results for different machine types.

	Method	Baseline AE	Baseline MobV2	Ours
ToyCar	AUC(source)	**90.41%**	58.92%	86.14%
	AUC(target)	34.81%	51.95%	**72.63%**
	pAUC	52.74%	52.36%	**62.07%**
ToyTrain	AUC(source)	76.32%	57.57%	**82.44%**
	AUC(target)	23.35%	45.79%	**68.23%**
	pAUC	50.48%	51.52%	**64.91%**
Bearing	AUC(source)	54.42%	60.55%	**74.87%**
	AUC(target)	58.38%	60.09%	**87.45%**
	pAUC	51.98%	56.96%	**69.72%**
Fan	AUC(source)	**78.59%**	70.75%	75.00%
	AUC(target)	47.18%	48.22%	**66.60%**
	pAUC	57.52%	56.94%	**67.00%**
Gearbox	AUC(source)	68.93%	69.19%	**87.36%**
	AUC(target)	62.64%	56.23%	**91.35%**
	pAUC	58.49%	56.07%	**75.60%**
Slider	AUC(source)	77.95%	65.05%	**93.34%**
	AUC(target)	47.67%	38.40%	**86.11%**
	pAUC	55.78%	54.73%	**80.36%**
Valve	AUC(source)	52.01%	67.66%	**92.46%**
	AUC(target)	49.46%	57.75%	**96.78%**
	pAUC	50.36%	62.64%	**88.56%**
All	Harmonic mean	52.61%	56.01%	**78.09%**

On the public test set, we will calculate all anomaly scores, and choose the most appropriate anomaly scores for different machine types.

2.3. Domain generalization

Since the target domain of each machine ID has only 10 samples, we did not use common domain generalization methods. We saved the average embeddings of the source and target domains of each machine ID during the training process, and the computation of the covariance matrix used all the ID's embeddings without distinguishing the source and target domains. When calculating the Mahalanobis distance, since we do not know if the audio belongs to the source domain or the target domain, we calculate the Mahalanobis distance between the embedding and the average embedding of the source and target domains, and then take the minimum value as the distance.

2.4. Audio preprocessing

Observing the spectrum, it is found that the features of some machine types are mainly concentrated in high frequencies. In light of this finding, we pass the features through a high-pass filter before passing through the Mel filter. For some machine types, experiments show that preprocessed features are more suitable for anomaly detection.

2.5. Model ensemble

The classifier is only an auxiliary task for anomaly detection. After training several epochs, we observed that the accuracy of the verification set is as high as 99%, but the detection performance is not completely positively correlated with the accuracy. In order to obtain a robust anomaly detection model, we saved the parameters of each model during the training process. According to the AUC performance of the model on the test set, we average several best epoch model parameters, and finally save the average embedding and covariance matrix of each machine ID to calculate the Mahalanobis distance and cosine distance.

3. EXPERIMENTS

3.1. Dataset and audio processing

All experiments are based on the DCASE 2022 task 2 dataset [8, 9, 10], which includes 7 machine types. The log Mel spectrum is used as the input feature. To compute STFTs and Mel-spectrograms, we use 1024 window size and 512 hop size. The number of Mel-frequency bins is 128. After this, we obtain a 128x313 input feature, where 313 is the time frame. According to the Baseline suggestions, we further divide the features to fit the model.

3.2. Experimental settings

We use pytorch for experiments. In training, the model is trained for 50 epochs with Adam as the optimizer, where the batch size is 128 and the learning rate is 0.001. In the fine-tuning phase, the only difference is that the batch size is 64.

The total training data set includes 7 machine types, and each machine type has 6 machine IDs. Since each machine ID has only 1000 training samples, it is convenient to directly train a dedicated classifier for each machine type. The embedding extracted by the model is not necessarily valid. Therefore, we first divide the data into 42 categories, and use cross entropy (CE) loss to train a classifier. After the training is completed, we fine-tune a model for each machine type separately. In the second stage, in addition to CE loss, we also use centerloss. In Table 2, we summarize the detailed experimental setup.

4. RESULTS AND DISCUSSIONS

We evaluate the detection performance using the area under the receiver operating characteristic curve (AUC) and the partial AUC (pAUC) with $p = 0.1$. Table 3 shows our experimental results. Compared with the baseline, our anomalous sound detection model significantly improves the detection performance. It is worth noting that result presented here did not use model ensemble. After the ensemble, the performance is slightly improved or decreased depending on the machine type, but our submissions are all based on ensemble, because the effect of the public test set does not fully represent the effect on the final blind test set.

5. REFERENCES

[1] J. Lopez, G. Stemmer, P. Lopez-Meyer, P. S. Singh, J. A. del Hoyo Ontiveros, and H. A. Courdourier, "Ensemble of complementary anomaly detectors under domain shifted conditions," DCASE2021 Challenge, Tech. Rep, Tech. Rep., 2021.

[2] K. Morita, T. Yano, and K. Tran, "Anomalous sound detection using cnn-based features by self supervised learning," *Tech. Rep., DCASE2021 Challenge*, 2021.

[3] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring under domain shifted conditions," *arXiv preprint arXiv:2106.04492*, 2021.

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

[5] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[6] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Chinese Conference on Biometric Recognition*. Springer, 2018, pp. 428–438.

[7] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[8] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," *In arXiv e-prints: 2205.13879*, 2022.

[9] 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 *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 1–5.

[10] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," *In arXiv e-prints: 2206.05876*, 2022.