AITHU SYSTEM FOR UNSUPERVISED ANOMALOUS SOUND DETECTION

Technical Report

Yufeng Deng[1], *Jia Liu*[1,2], *Jitao Ma*[3], *Xuchu Chen*[1], *Cheng Lu*[3], *Ruhang Xu*[3], *Wei-Qiang Zhang*[1]
[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Tsinghua AI Plus, Beijing 100084, China
[3]School of Economics and Management, North China Electric Power University, Beijing 102206, China
dyf20@mails.tsinghua.edu.cn liuj@tsinghua.edu.cn lucheng1983@163.com
wqzhang@tsinghua.edu.cn

ABSTRACT

This report describes the AITHU system for Task 2 of the DCASE 2021 challenge, *Unsupervised Anomalous Sound Detection for Machine Condition Monitoring under Domain Shifted Conditions*. The task aims to detect audio recordings containing anomalous machine sounds in a test set, when the training dataset itself does not contain any examples of anomalies. Moreover, the task is performed under the conditions that the acoustic characteristics of the training data and the test data are different (i.e., domain mismatch). We perform weighted mixing of data in different sections instead of to distinguish the data in the same part of different fields, and train a neural network to recognize mixed weights. The results of our approach are better than baseline systems for all machine types. In the development set, the official score of our approach is 67.12%.

Index Terms**—** Anomalous Sounds Detection, domain shift, self-supervision

1. INTRODUCTION

Anomalous sound detection (ASD) is the task of identifying whether the sound emitted from a machine is normal or anomalous. In reality, it is difficult to collect enough abnormal sound data from machines, and only enough normal data from machines can be collected. Therefore, the main difficulty of abnormal sound detection is to train a model that can distinguish between normal sound and abnormal sound using only normal sound data. Another challenge is that the task is performed under conditions where the acoustic characteristics of the training data and the test data are different (i.e., domain shift)[1]. In this challenge, all data are divided into two domains, source and target. Each domain has its own training data and test data, but the amount of data in these two domains is very unbalanced, the training data of the target domain is very small, and the ratio of the training data of the source domain to the training data of the target domain is 1000:3. So we have to train a neural network using data in the source domain and a small amount of data in the target domain, and the network should perform well in both source domain and target domain.

2. METHOD

The method we use is a self-supervised learning method. Self-supervision using classification tasks has been previously used for detecting anomalies in [2][3][4]. In these works, the learning task involves networks to discriminate between multiple geometric transformations. We employ a different strategy here. In this task, the data includes different machine types, each machine type is divided into six sections, and each section is divided into source and target domains, further we may use the section information of the data for self-supervised learning. We weighted and mixed the data of different sections, and trained the neural network to recognize the mixed weights. For example, if the data x_0 of section 0 and the data x_1 of section 1 are weighted and mixed in equal proportions, the expected output of the neural network is [0.5, 0.5, 0.0, 0.0, 0.0, 0.0]. When the weight of a section is set to 1, and the weights of other sections are set to 0, the neural network is trained to recognize the data of different sections. In the experiment, we found that when the number of mixed sections is greater than two or there are too many types of mixed, the result will be worse. There are three kinds of blending weights we use, each of which is applied to all sections.

1) Set the weight of one section to 1, and set the weight of other sections to 0.
2) Select two sections to be mixed in equal proportions, and set the weights of other sections to 0.
3) Select two sections to mix with the weights of 0.6 and 0.4, and set the weights of other sections to 0.

All blending is for pre-processed data.

3. EXPERIMENTS

3.1. Pre-Processing

We first performed some preprocessing on the raw audio data[5][6]. The preprocessing is done in a similar way as in the MobileNetV2-based baseline system[1]. First, the raw audio is normalized to zero mean and standard deviation one. Then we computed a mono-channel Short Time Fourier Transform using 1024-sample windows and a hop-size of 512 samples. We weighted the resulting power spectrogram with a mel-scaled filter-bank of 128 filters. The log-mel spectrogram is shown as Figure 1.

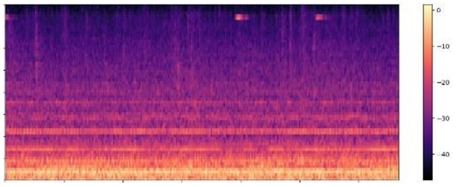

Figure 1: log-mel spectrograms of the sound

Table2: The harmonic mean of the AUC and pAUC in parentheses for machine types over all sections and domains.

Algorithm	Toy Car	Toy Train	Fan	Gearbox	Pump	Slider Rail	Valve	Harmonic mean
Baseline (AE)	62.49 (52.36)	61.71 (53.81)	63.24 (53.38)	65.97 (52.76)	61.92 (54.41)	66.74 (55.94)	53.41 (50.54)	57.27
Baseline (MobileNetV2)	56.04 (52.36)	57.46 (51.61)	61.56 (63.02)	66.70 (59.16)	61.89 (57.37)	59.26 (56.00)	56.51 (52.64)	57.67
Our approach	**73.86 (56.51)**	**65.90 (60.85)**	**72.05 (69.26)**	**72.16 (61.29)**	**71.08 (60.10)**	**68.43 (61.36)**	**84.86 (72.24)**	**67.12**

3.2. Inputs

The inputs to the classifiers are 128 × 256 images, which are the log-mel spectrograms computed using the following parameters:

1. With hop length of 32ms between frames, each input 10s file is split into frames of length 64ms.
2. 1024-FFT and 128 Mel bins are used to featurize each frame.
3. 256 featurized frames are stacked to form a 128 × 256 image.
4. The successive 128 × 256 images have an overlap of 255 frames.

3.3. Network Architecture

We use the model architecture (Table 1) introduced by Koutini et al.[7], a receptive-filed-regularized, fully convolutional, residual network (ResNet)[8].

Table 1: Model architecture for audio classification by Koutini et al.[7]. #K and KS are the number of kernels and kernel size, respectively. Residual Blocks (RB) consist of two Convolutional (Conv) layers with #K kernels, each followed by a Batch Normalization (BN) layer. GAP is a Global Average Pooling Layer. All activation functions are ReLUs. a and b are set to either 1 or 3 to control the receptive filed of the network. c controls the number of convolution filters.

ResNet

Type	#K	KS1	KS2
Conv	$c*2^0$	5	
BN	-	-	
RB	$c*2^0$	3	1
Max Pool	-	2	-
RB	$c*2^0$	3	3
Max Pool	-	2	-
RB	$c*2^0$	3	a
RB	$c*2^0$	3	b
Max Pool	-	2	-
RB	$c*2^1$	1	1
RB	$c*2^2$	1	1
RB	$c*2^2$	1	1
Conv	1	1	-
BN	-	-	-
GAP	-	-	-

Residual Block (RB)

Type	KS
Conv	KS 1
BN	
Conv	KS 2
BN	
Add Input	

3.4. Training

After data preprocessing, the raw audio file becomes a two-dimensional matrix. The first dimension of the matrix represents the number of mel-bins, and the second dimension is the time dimension. When training the neural network, each time a fixed-length time frame is selected as the input, the step interval is one, and the batch size is 32. The KL divergence is used as the loss function, the parameter update rule is Adam, and the neural network is trained for 20 epochs.

3.5. Anomaly score

Assume that the predicted probability of the correct section corresponding to the input data output by the neural network is *p*, then the anomaly score of the input data is calculated by (1). The anomaly score of each test file is the mean value of the anomaly scores of all input data belonging to this file. Each input of the test file is also taken to a fixed-length time frame, with a step interval of one.

$$anomaly\_score = \log\frac{1-p}{p} \qquad (1)$$

4. RESULTS

In this section, we report results using the development set. In Table 2, we report AUC results and pAUC in parentheses for the MobileNetV2-based baseline model, the AE-based baseline model, and our submission for all 7 machines averaged across sections and domains. As we can see from Table 2, the results of our approach are better than baseline systems for all machine types. The harmonic mean of the AUC and pAUC scores over all the machine types, sections and domains is 67.12%.

5. REFERENCES

[1] 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.

[2] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," arXiv preprint arXiv:1906.12340, 2019.

[3] I. Golan and R. El-Yaniv, ``Deep anomaly detection using geometric transformations," arXiv preprint arXiv:1805.10917, 2018.

[4] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," arXiv preprint arXiv:1803.07728, 2018.

[5] R. Tanabe, H. Purohit, K. Dohi, T. Endo, Y. Nikaido, T. Nakamura, and Y. Kawaguchi, "MIMII DUE: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions," arXiv preprint arXiv:2105.02702, 2021.

[6] 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," arXiv preprint arXiv:2106.02369, 2021.

[7] K Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive-field-regularized CNN variants for acoustic scene classification," arXiv preprint arXiv:1909.02859, 2019.

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE CVPR*, vol. Ⅰ, 2016, pp. 770-778.