ANOMALOUS SOUNDS DETECTION USING AUTOENCODER AND CLASSIFICATION METHODS

Technical Report

*Haisheng Lu, Yujie Fu, Huajing Qin, Shijin Huang, Yihan Wang, Chen Deng, Tianchu Yao, Huitian Jiang, Haifeng Wen, Chuang Shi**

University of Electronic Science and Technology of China, Chengdu, China
shichuang@uestc.edu.cn

ABSTRACT

This report described our contribution to Unsupervised Detection of Anomalous Sounds on DCASE 2021 challenge (Task2). Previous research results show that AE and outlier detection is a very effective solution to abnormal sound detection (ASD). This design based on previous research, using IDNN, FREAK and MobileFace Nets to implement unsupervised ASD.

Index Terms— IDNN, FREAK, MobileNetV2, ArcFace

1. INTRODUCTION

At present, in all research directions of sound signals, the most popular should be the research of sound recognition technology. The goal of anomalous sound detection (ASD) is to identify anomalous sounds when only sounds of the "normal" condition are available beforehand. ASD has been used for a variety of purposes, including audio surveillance, animal husbandry, product inspection and predictive maintenance. For the last application, because abnormal sounds usually indicate that the mechanical equipment is malfunctioning. Discovering abnormalities quickly will reduce the number of defective products and prevent the spread of damage.

ASD tasks can be roughly divided into supervised ASD and unsupervised ASD. The difference lies in the definition of abnormal sound [1]. Supervised ASD detects "determined" abnormal sounds, such as gunshots or screams, which is a rare sound event detection (SED). Once anomalies have been defined, even if anomalies are rarer than normal sounds, we can collect a data set of target abnormal sounds. In contrast, we cannot intentionally damage expensive machines in a factory to obtain abnormal sound samples [2]-[4]. Meanwhile, the environment of factory machine operation is relatively complex. It is difficult to obtain a complete set of fault samples and apply supervised learning in fault recognition. Therefore, this type of task is reasonably considered as an unsupervised classification problem.

In this report, according to the requirements of DCASE 2021 challenge task 2, we present four methods for industrial equipment to detect unknown abnormal sounds. The deviation between the normal model and the observed sound is calculated. Deviations are often referred to as "abnormal scores". The normal model represents the concept of normal behavior trained from normal sound training data. When the abnormal score is higher than a predetermined threshold, the observed sound is recognized as an abnormal sound.

2. PROPOSED APPROACH

2.1. Ensemble of classification methods

Inspired by the baseline system, we integrated two classification systems: one is the same as MobileNetV2 in baseline, the other uses MobileFace Nets as proposed [5]-[6]. A general view of our system is shown in Table 1.

Table1: Correspond machine types with models

Machine Type	Classification Model
Fan	MobileNetV2 baseline
Gearbox	MobileNetV2 baseline
Pump	MobileFace Nets
Slider	MobileFace Nets
Toycar	MobileFace Nets
Toytrain	MobileFace Nets
Valve	MobileNetV2 baseline

The input shape of MobileFace Nets is (1×1024×32). We load the audio clips and deal with STFT using librosa package [7], the length of the window is 2046 and the hop length is 512. Then the spectrograms are split into 10×(1024×32) columns (padding with zeros). The architecture of network is shown as Table 2. Column t, c, n, s refers to the expansion factor, output channels, the number of repetitions and stride.

Table2: Architecture of MobileFaceNets

Input	Layer	t	c	n	s
1×1024×32	Conv3×3	/	64	1	2
64×512×16	Depthwise Conv3×3	/	64	1	1
64×512×16	Bottleneck	2	64	5	2
64×256×8	Bottleneck	4	128	1	2
128×128×4	Bottleneck	2	128	6	2
128×64×2	Bottleneck	4	128	1	2
128×32×1	Bottleneck	2	128	2	1
128×32×1	Conv1×1	/	512	1	1
512×32×1	Linear GDConv32×1	/	512	1	1
512×1×1	Linear Conv1×1	/	128	1	1

Through this network, we got the 128-dimension embedding layer. The next step is to apply it to an ArcFace Layer. While testing a section00 sample's anomaly score, the test sample's embedding is compared with data of all the section00 audios. We average the most similar K results as the similarity score. We train our models on a single NVIDIA RTX2070 max-Q GPU. All the hyper-parameters are summarized in Table3.

Table3. Summarization of hyper-parameters

Parameters for audio processing	
Sampling rate	16000Hz
FFT length	2046
FFT hop length	512
ArcFace loss parameters	
Margin Parameter m	0.05
Re-scale Factor s	30
Cosine annealing strategy	
Initial learning rate	0.001
Epochs	100
Other parameters	
Batch size	48
K	10

2.2. IDNN

In the conditional method AE, it attempted to detect anomalies based on reconstruction errors. However, it may performance badly in some machines whose sound is no-stationary, because it is difficult to predict the edge frames. So, we imitate use Interpolation deep neural network (IDNN) to replace the AE [8]. In the IDNN, we remove the center frame, and use the other four frames as inputs. And then the neural net output the center frames. So, the reconstruction error is the difference between the original input and reconstructed output. When we train the model, we only use the normal machine, so we minimize the reconstruction error. The IDNN loss function is:

$$L_{PDNN} = \left\| x_n - D(E(x_{1,\ldots\ldots,n})) \right\|_2^2 \quad (1)$$

We use a fully connected net-work as the architecture, and the structure is :[512 128 128 128 8 128 128 128 128].

Following the baseline model, we split 10s file into 64ms each frame. And hop length is 32ms. And the we use 1024-FFT and 128 Mel bins to extract the feature of each frames. We use 5 frames as inputs.

2.3. VIDNN without KL divergence

In this section we introduce the Variational Interpolation Deep Neural Network (VIDNN), which is an adaptation of the Variational Autoencoder [8].

2.3.1. Model setup

Unlike the normal VAE, which reconstruct the whole input spectrogram from the latent space extracted from whole hidden layer, the IDNN utilizes multiple frames of a spectrogram whose center frame is removed as an input, and it predicts an interpolation of the removed frame as an output as the figure 1.

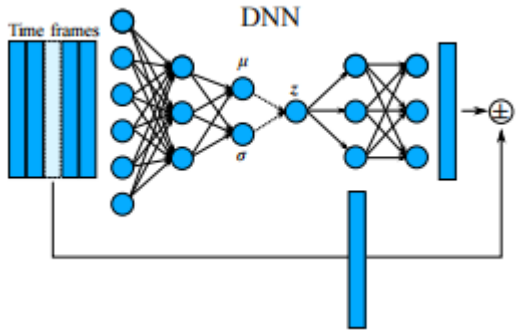

Figure1: Proposed architecture of VIDNN [8]

Also, in our experiment, we coincidently removed the kl Divergence of the VAE and the find out the result has been improved. This situation may be caused by that the elements in latent space are not a simple normal distribution.

For our training dataset is extremely unbalanced, we train one model for one source domain and one target domain to try to decrease this unbalance.

2.3.2. Model architecture

Our model architecture is very closed to the baseline.

Table4: Example of placing a figure with experimental results.

Layer	I	O
Dense	640	128
Dense	128	128
Dense	128	128
Dense	128	128
Dense (μ)	128	8
Dense (σ)	128	8
z=N(μ,σ)	8,8	8
Dense	8	128
Dense	128	128
Dense	128	128
Dense	128	128

2.3.3. Training

Our loss function is MSEloss for predicted frame and real frame. The model is trained for 100 epoch using ADAM update rule. Learning rate was set to 0.001 at the beginning, and then decreased after the 80th epoch.

2.4. FREAK

Inspired by [9], we also tried to use Freak to solve ASD task.

2.4.1. Model setup

FREAK is an adaptation of the WaveNet in frequency domain instead of the time domain. Typical FREAK is trying to predicting the next frame in the spectrogram of a recording of interest.

Inspired by IDNN, we make our FREAK predict the center frame.

2.4.2. Model architecture

Model architecture is shown in next figures.

Table 5: WaveNet Block

Layer	channels	kernel	dilation	groups
CausalConv1d	4	3	1	4
ResidualLayer	4	3	1	4
ResidualLayer	4	3	2	4
ResidualLayer	4	3	4	4
ResidualLayer	4	3	8	4
ResidualLayer	4	3	1	4
ResidualLayer	4	3	2	4
ResidualLayer	4	3	4	4
ResidualLayer	4	3	8	4

Architecture (table 5) is based on WaveNet structure, and frame are treated as channels. Frequency bins are treated as group and processed separately. Residual block is strictly same as standard WaveNet. The skip size is set to 2. Skips of the channels are cat together and reshaped to (batchsize,4*2*group). Then two dense layers are applied for the final predict.

2.4.3. Training

Our loss function is MSEloss for predicted frame and real frame. The model is trained for 40 epochs using ADAM update rule. Learning rate was set to 0.0001 at the beginning, and then decreased after the 80th epoch.

3. EXPERIMENTAL SETUP

3.1. Dataset

The data used for this task comprise ToyADMOS and MIMII data sets which provided by the task organizers [11]-[12]. Data sets is divided into two parts, development datasets and evaluation datasets. And they are split into a training and testing subset. The training datasets only contained normal samples, but the testing datasets contained normal and abnormal samples.
Each subset consists of three sections for each machine type (machine include fan, gearbox, slider, toycar, toytrain, valve). Different sections expressed machine in different domain.

3.2. Pre-Processing

Pre-Processing of task2_2 - task2_4 are same as the baseline. Pre-Processing of task2_1 is described in section 2.1

4. RESULT

The AUC and pAUC on the development dataset are shown below.

Table 6: Result of Classification Methods

Machine Type	AUC	pAUC
Fan	61.8%	64.7%
Gearbox	63.7%	56.2%
Pump	67.2%	56.8%
Slider	69.2%	60.7%
Toycar	63.5%	56.9%
Toytrain	60.4%	53.5%
Valve	64.5%	53.9%
Total	64.2%	57.3%

Table 7: Result of VIDNN

Machine Type	AUC	pAUC
Fan	66.5%	54.5%
Gearbox	70.0%	53.6%
Pump	60.8%	54.5%
Slider	67.6%	56.4%
Toycar	71.4%	58.9%
Toytrain	60.4%	53.5%
Valve	59.0%	50.5%
Total	65.8%	54.5%

Table 8: Result of FREAK

Machine Type	AUC	pAUC
Fan	62.2%	53.4%
Gearbox	65.4%	52.6%
Pump	62.4%	54.7%
Slider	66.4%	55.1%
Toycar	66.0%	54.0%
Toytrain	64.2%	53.9%
Valve	56.5%	50.3%
Total	63.1%	53.4%

Table 9: Result of IDNN

Machine Type	AUC	pAUC
Fan	64.4%	53.14%
Gearbox	68.4%	53.87%
Pump	61.48%	54.37%
Slider	67.80%	56.05%
Toycar	66.60%	57.78%
Toytrain	67.55%	57.78%
Valve	57.37%	50.26%
Total	64.6%	54.2%

Table 10: Result of MobileNetV2-based baseline

Machine Type	AUC	pAUC
Fan	61.56%	63.02%
Gearbox	66.70%	59.16%
Pump	61.89%	57.37%
Slider	59.26%	56.00%
Toycar	56.04%	56.37%
Toytrain	57.46%	51.61%
Valve	56.51%	52.64%
Total	59.84%	56.59%

Table 11: Result of AE baseline

Machine Type	AUC	pAUC
Fan	63.24	53.38%
Gearbox	65.97%	52.76%

Pump	61.92%	54.41%
Slider	66.74%	55.94%
Toycar	62.49%	52.36%
Toytrain	62.21%	53.31%
Valve	53.41%	50.54%
Total	64.6%	54.2%

5. REFERENCES

[1] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," *Proceedings of the 2007 IEEE Conference on Advanced Video and Signal based Surveillance*, London, UK, 2007, pp. 21-26.

[2] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85-126, 2004.

[3] A. Patcha and J. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448-3470, 2007.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detecti on: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1-58, 2009.

[5] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop*, Tokyo, Japan, 2020, pp. 81-85.

[6] Q. Zhou, "Arcface based sound mobilenets for DCASE 2020 task 2," *2020 IEEE AASP Detection and Classification of Acoustic Scenes and Event*s *Task 2*, Tokyo, Japan, 2020.

[7] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "LIBROSA: Audio and music signal analysis in python," *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18-25.

[8] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," *Proceedings of the ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, 2020, pp. 271-275.

[9] P. Daniluk, M. Gozdziewski, S. Kapka, and M. Kosmider, "Ensemble of auto-encoder based systems for anomaly detection," *Proceedings of the 2020 IEEE AASP Detection and Classification of Acoustic Scenes and Event*s *Task 2*, Tokyo, Japan, 2020.

[10] 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 e-prints: 2106.04492*, 2021, pp. 1-5.

[11] 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 e-prints: 2106.02369*, 2021.

[12] 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 e-prints: 2006.05822*, 2021, pp. 1-4.