UNSUPERVISED DETECTION OF ANOMALOUS SOUNDS BASED ON DICTIONARY LEARNING AND AUTOENCODER

Chenxu Zhang[1]*, Yao Yao*[1]*, Yuxuan Zhou*[2]*, Guoheng Fu*[3]*, Shengchen Li*[3]*, Gang Tang*[2]*, Xi Shao*[1]

[1]Nanjing University of Posts and Telecommunications, China
(1018010429, 1019010528, shaoxi)@njupt.edu.cn
[2]Beijing University of Chemical Technology, China
nemozyx@163.com, tanggang@mail.buct.edu.cn
[3]Beijing University of Posts and Telecommunications, China
(fgh, shengchen.li)@bupt.edu.cn

ABSTRACT

The DCASE2020 Challenge Task2 is to develop an unsupervised detection system of anomalous sounds for six types of machine. In this paper, we proposed two methods. One is to use auditory traditional features and dictionary learning (DL) to train a dictionary. Another is to use auditory spectral features and deep learning method to train an autoencoder (AE). Both of our proposed methods achieve an improvement comparing to the baseline system, and better performance can be obtained by using the mixture of two methods. Experiments prove the practicability of the proposed methods for anomaly detection.

Index Terms— Unsupervised anomaly detection, Auditory traditional features, Dictionary learning, Log Mel-filter bank, Autoencoder

1. INTRODUCTION

Anomaly detection in sound (ADS) has received much attention. Since abnormal sounds may indicate symptoms of errors or malicious activity, timely detection of them can prevent such problems. In particular, ADS has been used for a variety of purposes, including audio surveillance [1], animal husbandry [2], product inspection and predictive maintenance [3].

Unsupervised ADS [4], [5] is to detect unknown anomalous sounds under the condition that has not been observed. Since actual abnormal sounds rarely occur and have high variability, in this paper, we aim to detect unknown abnormal sounds based on unsupervised methods. We proposed two different methods to perform ADS tasks based on dictionary learning [6], [7] and autoencoder [8], [9], respectively, and obtain the best results of them.

The remainder of paper is organized as follows. In section 2, we present about proposed dictionary learning method. In section 3, we present about proposed autoencoder method. The experimental results are discussed in section 4. Finally, conclusions are given in section 5.

2. DICTIONARY LEARNING METHOD

As shown in Figure 1, the system developed mainly contains three parts: feature extraction, dictionary learning and one-class SVM classifier [10]. We first use normal audio in training set to train a dictionary and a classifier, then predict on the testing set according to the trained model. Details of each part are discussed below.

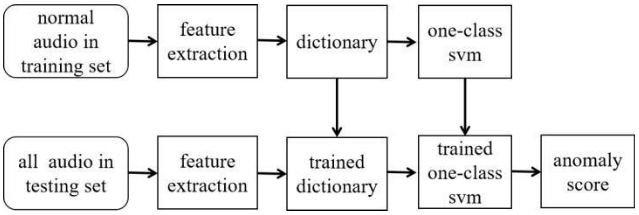

Figure 1: Framework of the proposed system.

2.1. Feature Selection

Traditional features of auditory signals are used in the dictionary learning based system. Since audio signals contain noise, especially our signal-to-noise ratio of audio signals is low, deep network only uses the features of the data distribution level which is not easy to distinguish normal and abnormal sounds. Traditional features contain fault information, which is more conducive to learning the distribution related to abnormalities, thereby improving the classification accuracy. In addition, traditional features have good stability, considering the generalization of the system.

Setting the length of sample window to 1024 points, hop size to 512 points, each audio in the dataset divided into more than 300 frames, then the traditional features are calculated for each frame and get a 16-dimensional feature which are normalized before the dictionary learning. Table 1 shows the 16 features we selected.

Table 1: the name and definition of selected traditional features

Name	Definition
Variance (Var)	$\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - x)^2}$
Square root amplitude (X_r)	$(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\lvert x_i \rvert})^2$
Kurtosis Index (I_p)	$\frac{X_p}{X_{rms}}$

Crook Index (C_w)	$\frac{\frac{1}{N}\sum_{i=1}^{N}(\lvert x_i\rvert - x)^2}{X_{rms}^3}$
Slope (β)	$\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{x}_{\mathrm{i}}^{4}$
Effective Value (X_{rms})	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}x_i^2}$
Pulse Index (C_f)	$\frac{X_p}{\bar{X}}$
Waveform index (S)	$\frac{X_{rms}}{\bar{X}_{\mathrm{abs}}}$
Kurtosis (X_p)	$\frac{max(x_i) - min(x_i)}{2}$
Margin Index (C_e)	$\frac{X_{rms}}{\bar{X}}$
Root mean square Frequency (RMSF)	$\sqrt{\frac{\sum f^2\mathrm{S}(f)}{\sum \mathrm{S}(f)}}$
Mean square frequency (MC)	$\frac{2}{\mathrm{N}}\sum \mathrm{S}(f)$
Fourier sum of squares (E)	$\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{S}^2(\mathrm{n})$
Frequency variance (VF)	$\frac{\sum(f-\mathrm{FC})^2\mathrm{S}(f)}{\sum \mathrm{S}(f)}$
Center of gravity (FC)	$\frac{\sum f\mathrm{S}(f)}{\sum \mathrm{S}(f)}$
Frequency standard deviation (RVF)	$\sqrt{\mathrm{VF}}$

Note: $\bar{X}_{abs}$ presents the absolute value, *N* presents the number of samples, f=(1:N/2)*f_s/N，f_s presents the sampling frequency, $S(f)$ presents the frequency spectrum of vibration signal during sampling time.

2.2. Dictionary Learning

As the audio feature inputs, a set of over-complete bases [11] is used, hence, an approximate representation of the original audio segment can be obtained (i.e. $Y \approx DX$) under the condition of satisfying a certain sparsity or reconstruction error T_0. This representation problem can be described as:

$$\min_{D,X} \|Y - DX\|_F^2, s.t. \forall i, \|\boldsymbol{x}_i\|_0 \leq T_0, \tag{1}$$

The goal of dictionary learning is to minimize the reconstruction error and make the coefficient matrix as sparse as possible to obtain a more concise representation of the signal and reduce the complexity of the model.

For the audio feature *Y* which has been framed, we randomly sample 5 percent of the frame feature as training data to participate in dictionary training due to the limitation of RAM. In sparse representation stage, OMP algorithm [12] is used to obtain the coefficient matrix of the corresponding dictionary. K-SVD algorithm [13] is used in training dictionary phase. Until the algorithm converges, the joint optimization of dictionary and matrix is finally completed. Using the trained dictionary, the sparse representation coefficient matrix *X'* of the input sample *Y'* within the reconstruction error of 10^{-7} can be obtained.

2.3. Outlier Detection

After the dictionary created, the sparse representation coefficient matrix of the input normal sample in training set is used to train a one-class SVM classifier. The classifier can study the distribution of the inputs and can be used as a discriminator in testing phase. If the input is similar to the training data, it will output 1, otherwise, it will output -1.

Here, the deviation between a normal model and an observed sound is calculated, the deviation is often called the "*anomaly score*". Two different scoring methods are set to get it. For the first approach percentage scoring is used, an audio is divided into more than 300 frames. After each frame is predicted by the classifier, a label of 1 or -1 will be obtained. The score is determined by calculating the proportion of -1 labels in all frames. For the second approach continuous scoring is used, the label of each frame obtained by the above method, then traverse the labels of these consecutive frames. If we encounter -1, we get 1 point. If the next label is still -1, then the score of the next label is 2. If continue continuously, the score of the next label is 4. And so on, until you encounter 1. Then when -1 is encountered again, repeat the previous process. This scoring method can be described as:

$$\mathrm{Score} = \sum a_i * p_i, \tag{2}$$

$$a_i = \begin{cases} 0 & if \quad the\ i-th\ label\ = 1 \\ 1 & if \quad the\ i-th\ label\ = -1 \end{cases}$$

$$p_i = \begin{cases} 1, & if\ \ i=1 \\ 1, & if\ a_i = 1\ \ and\ \ a_{i-1} = 0 \\ 2 * p_{i-1}, & if\ a_i = 1\ \ and\ \ a_{i-1} = 1 \\ 0, & other \end{cases}$$

Through the above method, we get the audio anomaly score *S*, the higher the score, the greater the possibility of abnormal. The observed sound x_τ is identified as an anomalous one when the anomaly score is higher than a pre-defined threshold value *Φ*.

$$decision = \begin{cases} 0(normal) & if\ S(x_\tau) \leq \Phi \\ 1(anomaly) & if\ S(x_\tau) > \Phi \end{cases}, \tag{3}$$

2.4. Post-processing

We set different kernel type and an upper bound for all machine in one-class SVM classifier training phase. For different types of machine, we try 4 types kernel and set upper bound range from 0 to 1 to find most robust parameters.

To evaluate the performance of our method, the post-processing step translates the anomaly scores into AUC and pAUC. The AUC is a traditional performance measure of anomaly detection. The pAUC is an AUC calculated with FPRs ranging from 0 to p with respect to the maximum value of 1.

3. AUTOENCODER METHOD

As shown in Figure 3, the autoencoder system developed mainly contains three parts: feature extraction, autoencoder and scoring part. Normal audio in training set is used to train an autoencoder, then reconstruct all audio in testing set according to the trained model. Details of each part are discussed below.

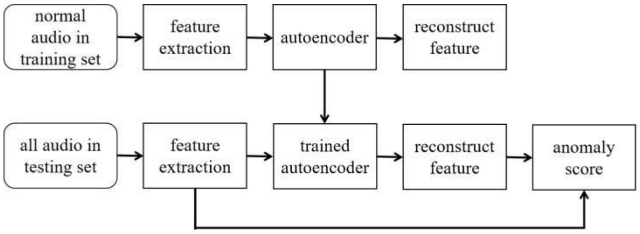

Figure 2: Framework of the proposed system.

3.1. Feature Selection

Spectral features of auditory signals are used in the autoencoder based system. Setting the length of sample window to 64ms, hop size to 32ms, number of filters to 128 and maximum frequency to 8000 Hz to get log mel-filter banks, each audio is divided into more than 300 frames with 128-dimensional which are standardized before training.

3.2. Autoencoder

The purpose of using autoencoder is to learn representation of the input features by using two neural networks *E* and *D*, which are called the encoder and decoder, respectively.

For the audio feature x which has been framed, the encoder network *E* convert x into a latent feature z, then the audio feature x is reconstructed from z by decoder network *D*. We obtain a reconstructed feature x' through autoencoder.

In this paper, we modify the structure of autoencoder in baseline system, the encoder consists of one input FCN layer, 2 hidden FCN layers, and one output FCN layer. The hidden units are set to 64, 32, 16 and 8, respectively, considering the relationship between the amount of training data and the total amount of parameters in training phase. The decoder structure is corresponding to the encoder.

3.3. Outlier Detection

In training phrase, we use extracted features of normal audio in training set to train autoencoder. Since autoencoder is trained to learn representation of normal feature, in testing phrase, the reconstruction error would be small if x is normal. Thus, we use reconstruction error as anomaly score, the score is defined as

$$score = \|x - D(E(x))\|_2^2, \tag{4}$$

Through the above method, we get the audio anomaly score, the higher the score, the greater the possibility of abnormal. A pre-defined threshold value Φ is used to identify anomalous audio as illustrated in (3). We also use the post-processing step mentioned in sec 2.4 to get AUC and pAUC.

4. EXPERIMENTAL RESULTS

In this part, we discuss the performance of the proposed methods and compare to the DCASE 2020 task2 baseline system[14], [15], [16]. We compare average performance for six types of machine and performance for every machine using two proposed methods to obtain the best results.

Table 2: Average AUC of the baseline, dictionary learning (DL) and autoencoder (AE) method for six types of machine

Machine	Baseline (%)	DL (%)	AE (%)
fan	65.83	72.55	70.62
pump	72.89	67.46	76.40
slider	84.76	79.49	82.00
valve	66.28	72.33	60.22
toycar	78.77	49.25	80.23
toyconveyor	72.53	53.67	71.75

Table 3: AUC of the baseline, dictionary learning (DL) and autoencoder (AE) method for all machine

Machine	Id	Baseline (%)	DL (%)	AE (%)
fan	0	54.41	69.07	55.53
	2	73.40	72.00	81.04
	4	61.61	82.24	59.87
	6	73.92	66.88	86.03
pump	0	67.15	82.90	67.03
	2	61.53	83.90	65.73
	4	88.33	53.03	98.06
	6	74.55	50.00	74.78
slider	0	96.19	96.93	93.34
	2	78.97	68.63	77.28
	4	94.30	86.21	92.39
	6	69.59	66.17	61.78
valve	0	68.76	54.00	59.66
	2	68.18	87.90	63.66
	4	74.30	82.90	68.03
	6	53.90	64.50	49.55
toycar	1	81.36	50.00	77.80
	2	85.97	47.00	84.03
	3	63.30	50.00	69.55
	4	84.45	47.00	89.64
toyconveyor	1	78.07	53.00	80.74
	2	64.16	54.00	64.76
	3	75.35	53.00	69.75

4.1. Dictionary Learning Results

Setting different kernel type and an upper bound for all machine. We compare the method of percentage scoring and continuous scoring for different types of machine and choose the better one for them. Then we obtain the AUC for all machine.

The comparation of average performance is shown in Table 2. For the machine type of fan and valve, the average performance achieves a significant increment in AUC from 65.83, 66.28 to 72.55 and 72.33, respectively. For other machine types, the average performance exhibited a litter bit worse AUC than the baseline.

In Table 3, we compare the performance for all machine. For most machine, the performance improves a lot. Especially, the performance of fan4, pump2 and valve2 improved about 20% compared with the baseline. Although average performance for the machine type of pump and slider exhibited a litter bit worse, some separate machines such as pump0, pump2 and slider0 have better performance than baseline system. However, same as average performance, toycar and toyconveyor become worse.

4.2. Autoencoder Results

After training autoencoder used normal audio in training set, anomaly score obtained for all audio in testing set. Then the anomaly score translated to AUC to evaluate the performance.

The comparation of average performance is shown in Table 2. For the machine type of fan, pump and toycar, the average performance achieves an increment in AUC. For other machine types, the average performance exhibited a litter bit worse AUC than the baseline.

In Table 3, we compare the performance for all machine. For most machine, the performance improves a lot. Although average performance for the machine type of toyconveyor exhibited a litter bit worse, toyconveyor1 have better performance than baseline system. Same as average performance, slider and valve become worse.

4.3. Experiments Summary

According to the AUC results shown in Table 2 and Table 3, both of our proposed methods outperform the baseline for most machine. Using the mixture of two methods to perform DCASE task2 can obtain better performance shown in Table 4. For fan and pump, the two proposed methods are complementary, using DL for fan0, fan4, pump0 and pump4, using AE for the remaining machine will obtain best performance. For toycar and toyconveyor, the performance of using AE is better than DL. For valve, DL obtain a high performance. For slider, we get a litter worse performance than baseline, AE is finally used considering the generalization of the system.

Table 4: Average AUC of the baseline and mixture system for six types of machine

Machine	Baseline (%)	Mixture (%)
fan	65.83	79.60
pump	72.89	84.91
slider	84.76	82.00
valve	66.28	72.33
toycar	78.77	80.23
toyconveyor	72.53	71.75

5. CONCLUSION

In this work, we proposed two methods for task2 of DCASE 2020 challenge. Overall, both of our proposed methods outperform the baseline system among most machine. According to the AUC results on development set, only use one method of them is not sufficient to perform well in this task, so the final system is a mixed approach.

To further improve the system, future work can be done by 1) training the model on additional datasets to set most suitable kernel type and an upper bound for all machine. 2) modify the structure of autoencoder to get better performance. 3) select suitable super frame for both methods.

6. REFERENCES

[1] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, “Audio surveillance of roads: A system for detecting anomalous sounds,” IEEE transactions on intelligent transportation systems, 2015, 17(1): 279-288.

[2] Y. Chung, S. Oh and J. Lee, “Automatic detection and recognition of pig wasting diseases using sound data in audio surveillance systems,” Sensors, 2013, 13(10): 12929-12942.

[3] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, “Optimizing Acoustic Feature Extractor for Anomalous Sound Detection Based on NeymanPearson Lemma,” 25th European Signal Processing Conference (EUSIPCO). IEEE, 2017: 698-702.

[4] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection : A Survey,” ACM computing surveys (CSUR), 2009, 41(3): 1-58.

[5] Y. Koizumi, S. Saito, H.U.Y. Kawachi, and N. Harada, “Unsupervised Detection of Anomalous Sound based on Deep Learning and the Neyman-Pearson Lemma,” IEEE/ACM Transactions on Audio, Speech and Language Processing, 2018, 27(1): 212-224.

[6] M. Dandan, Y. Yuan, and Q. Wang, “A Sparse Dictionary Learning Method for Hyperspectral Anomaly Detection with Capped Norm,” IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2017: 648-651.

[7] L. Tong, S. Liu, and H. Zha, “Incoherent dictionary learning for sparse representation,” Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012). IEEE, 2012: 1237-1240.

[8] E. Marchi, F. Vesperini, F.Eyber, “A Novel Approach for Automatic Acoustic Novelty Detection Using a Denoising Autoencoder with Bidirectional LSTM Neural Networks,” IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015: 1996-2000.

[9] Y. Kawaguchi, T. Endo, “How can we detect anomalies from subsampled audio signals?” IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2017: 1-6.

[10] P. Roberto, G. Gu, and W. Lee, “Using an Ensemble of One-Class SVM Classifiers to Harden Payload-based Anomaly Detection Systems,” Sixth International Conference on Data Mining (ICDM'06). IEEE, 2006: 488-498.

[11] D.M. Malioutov, M. Cetin, and A.S. Willsky, “Optimal sparse representations in general overcomplete bases,” IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, 2004, 2: ii-793.

[12] A. Michal , M. Elad, and A. Bruckstein, “K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation,” IEEE Transactions on Signal Processing, 2006, 54(11): 4311-4322.

[13] W. Huang, H. Sun, J. Luo, “Periodic feature oriented adapted dictionary free OMP for rolling element bearing incipient fault diagnosis,” Mechanical Systems and Signal Processing, 2019, 126: 137-16.

[14] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, “ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection,” IEEE Workshop on

Applications of Signal Processing to Audio and Acoustics (WASPAA). IEEE, 2019: 313-317.

[15] H. Purohit, R. Tanabe, K. Ichige, T. Endo, Y. Nikaido, and K. Suefusa, “MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection,” In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019), 2019: 209–213.

[16] 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,” In arXiv e-prints: 2006.05822, June 2020: 1–4.