SUB-CLUSTER ADACOS BASED UNSUPERVISED ANOMALOUS SOUND DETECTION FOR MACHINE CONDITION MONITORING UNDER DOMAIN SHIFT CONDITIONS

Technical Report

Yudai Asai

PFU Limited., Kanagawa, Japan
asai.yudai.pfu@fujitsu.com

ABSTRACT

This technical describes our approaches for the DCASE 2021 Challenge Task 2. Our approaches are based on deep metric learning using sub-cluster AdaCos loss and outlier detection using GMM and One-Class SVM. To tackle the difficulties of domain shift conditions, first we trained our model with only source domain data, and then, fine-tuned with source and target domain data.

We achieved an averaged area under the curve (AUC) of 66.12% and averaged partial AUC ($p = 0.1$) of 58.18% on the test data in development dataset.

Index Terms— Unsupervised Anomaly Detection, Deep Metric Learning, Machine Condition Monitoring

1. INTRODUCTION

In this technical report, we describe our approaches for the DCASE 2021 Challenge Task 2, *Unsupervised Anomalous Sound Detection for Machine Condition Monitoring under Domain Shifted Conditions* [1]. The goal of this task is to determine whether the condition of a machine is normal or anomalous from a sound emitted by the machine using a model trained on the dataset which contains only normal condition sound.

In this task, ToyADMOS2 [2] and MIMII DUE [3] datasets are used. These dataset contains normal/anomalous operating sounds of seven types of machines. The data of each machine type is divided to six subsets, called sections. And for each section, sounds recorded in source and target domains are provided.

Our approaches are based on deep metric learning using sub-cluster AdaCos loss [4] and outlier detection using GMM and One-Class SVM [5]. We used ResNet [6] based model to extract feature vector from log-mel spectrogram.

2. PROPOSED APPROACH

2.1. Preprocessing

First, short-time Fourier transform (STFT) is applied using Hann window. Window length and hop length are 1024 and 512, respectively. Then, STFT spectrogram is converted to Mel spectrogram using 128 bandpass filters. Finally, Mel spectrogram is converted to dB scale and standardized by mean and standard deviation calculated from all training data.

Table 1: Architecture of modified ResNet

Operation	Structure	Output size
Input	-	313×128
Conv2D	$7 \times 7, \text{stride} = 2$	$157 \times 64 \times 16$
Residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2, \text{stride} = 1$	$79 \times 32 \times 16$
Residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2, \text{stride} = 1$	$40 \times 16 \times 32$
Residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2, \text{stride} = 1$	$20 \times 8 \times 64$
Residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2, \text{stride} = 1$	$10 \times 4 \times 128$
MaxPool2D	$10 \times 1, \text{stride} = 1$	$1 \times 4 \times 128$
Flatten	-	512
Linear	-	128

2.2. Feature extractor

We used modified version of ResNet [6] to extract feature vector from each spectrogram. Model architecture is same as proposed in [4]. Model architecture is shown in Table 1. In this model leaky ReLU activation function [7] is used and batch normalization [8] layer is added after each convolution layer.

2.3. Loss function

In order to reduce intra-class distance and increase inter-class distance in feature vector space, we adopted sub-cluster AdaCos [4] cosine-based softmax loss. This loss function is modified version of AdaCos [9], relaxing the restriction of number of clusters for each class and allowing to use *mixup* [10] technique.

Let x_i denotes feature vector of i-th sample in mini-batch and W_j be sub-cluster center of class j learned by sub-cluster AdaCos, probabilities of i-th sample belonging to class j are given by

$$P_{i,j} = \sum_{l \in \mathcal{M}^{(j)}} \frac{\exp(s \cdot \cos\theta_{i,l})}{\sum_{k=1}^{CS} \exp(s \cdot \cos\theta_{i,k})} \tag{1}$$

where $\mathcal{M}^{(j)}$ denotes all sub-clusters belonging to class j and $\cos\theta_{i,j}$ is calculated by cosine similarity $\cos\theta_{i,j} = \langle x_i, W_j \rangle / \|x_i\|\|W_j\|$. s is scaling factor dynamically updated during training, and C is number of classes and S is number of sub-clusters. In this task $C = 84$ and we used $S = 32$.

2.4. Training strategy

The training data in development dataset in this task is highly unbalanced. Specifically, for each section, 1000 source domain data are provided while only 3 target domain data are provided. To mitigate this difficulty, we conducted training in following 2-steps:

- Step 1: Train model only with source domain data. Two source domain data are randomly picked for each sample in mini-batch.
- Step 2: Fine-tune model with source and target domain data. One source domain and target domain data are randomly picked for each sample in mini-batch.

We used mixup technique to augment training data by interpolating two samples and their one-hot encoded labels. We adopted uniform distribution for sampling mixing ratio. Since generated labels are no longer one-hot encoded, we trained our entire model with KL divergence loss between outputs of sub-cluster AdaCos and generated labels in both Step 1 and Step 2.

2.5. Outlier detection

To detect anomalies from feature vectors, we trained outlier detection models using feature vectors extracted from training data in development dataset.

For source domain data, we trained Gaussian Mixture Model (GMM)s which number of components is equal to number of sub-clusters for each combination of machine types and sections using learned sub-cluster centers as initial mean vectors of GMM. Then, anomaly score of each test data is calculated by largest negative log-likelihood of all components of GMM.

For target domain data, we trained One-Class Support Vector Machine (OCSVM)s [5] instead of GMM due to lack of samples in training data. Anomaly scores are calculated by signed distance to the separating hyperplane.

3. EXPERIMENTS

3.1. Training

We implemented our model using PyTorch [11] and Catalyst [12] and trained it for 150 epochs with source domain and fine-tuned for 5000 iterations with source and target domain data. In both step we used AdamP optimizer [13] with learning rate 0.001, weight decay 10^{-5}, and batch-size 256. When fine-tuning our model, parameters other than the last residual block and sub-cluster AdaCos layer are frozen. We trained just a single model which handles all machine types, sections, and domains.

3.2. Results

Harmonic mean of area under the curve (AUC) scores and partial AUC ($p = 0.1$) scores in parentheses calculated on the test data in development dataset are shown in Table 2. Baseline 1 and 2 means Autoencoder-based baseline and MobileNetV2-based baseline, respectively.

Table 2: DCASE 2021 Task 2 Experimental Results

Machine type	Baseline 1	Baseline 2	Ours
Fan	63.24 (53.38)	61.56 (63.02)	61.65 (61.44)
Gearbox	65.97 (52.76)	66.70 (59.16)	65.59 (53.09)
Pump	61.92 (54.41)	61.89 (57.37)	64.58 (57.40)
Slider	66.74 (55.94)	59.26 (56.00)	70.85 (58.78)
ToyCar	62.49 (52.36)	56.04 (56.37)	61.84 (55.79)
ToyTrain	61.71 (53.81)	57.46 (51.61)	65.63 (56.02)
Valve	53.41 (50.54)	56.51 (52.64)	74.20 (67.47)

4. 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," *CoRR*, vol. abs/2106.04492, 2021.

[2] 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.

[3] 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," *CoRR*, vol. abs/2105.02702, 2021.

[4] K. Wilkinghoff, "Sub-cluster adacos: Learning representations for anomalous sound detection," accepted at International Joint Conference on Neural Networks (IJCNN).

[5] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778.

[7] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the International Conference on Machine Learning*, Atlanta, Georgia, 2013.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, ser. JMLR Workshop and Conference Proceedings, F. R. Bach and D. M. Blei, Eds., vol. 37. JMLR.org, 2015, pp. 448–456.

[9] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "Adacos: Adaptively scaling cosine logits for effectively learning deep face representations," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 10 823–10 832.

[10] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8024–8035.

[12] S. Kolesnikov, "Accelerated deep learning R&D," https://github.com/catalyst-team/catalyst, 2018.

[13] B. Heo, S. Chun, S. J. Oh, D. Han, S. Yun, G. Kim, Y. Uh, and J.-W. Ha, "AdamP: Slowing down the slowdown for momentum optimizers on scale-invariant weights," in *International Conference on Learning Representations (ICLR)*, 2021.