ARCFACE BASED SOUND MOBILENETS FOR DCASE 2020 TASK 2

Technical Report

Qiping Zhou

PFU SHANGHAI Co., LTD
46 Building 4~5 Floors, 555 GuiPing Road
XuHui District, Shanghai 200233, CHINA
qpzhou.pfu@cn.fujitsu.com

ABSTRACT

In this report, we propose our anomalous sounds detection neural network for the DCASE 2020 challenge's Task 2 (Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring) [1] [2]. The main challenge of this task is to detect unknown anomalous sounds under the condition that only normal sound samples have been provided as training data. We propose a metric learning model based on additive angular margin loss (ArcFace) [3]. In order to learn the embedding efficiently, a CNN architecture based on MobileFaceNets [4] is employed.

Index Terms— DCASE2020, anomalous sounds detection, metric learning, ArcFace, MobileFaceNets

1. INTRODUCTION

Since the main challenge of this task is that only normal sound samples have been provided for training, anomalous sounds features could not be learned through a supervised two-class classification solution. In [5], the set of anomalous sounds are considered as the complementary set of normal sounds and simulated anomalous sounds. Inspired by this assumption, we consider the sounds that are not similar to the observed normal sounds as anomaly.

In this task, we train a machine ID classifier for each machine type's sound, which tries to identify each machine ID under a certain machine type. The classification head is abandoned in the final anomaly score calculation procedure and the reset MobileFaceNets based feature extractor part is used for measuring cosine similarity in embedded space.

1.1. DCASE 2020 Task2 Dataset

The data used for this task comprises parts of ToyADMOS [6] and the MIMII Dataset [7] consisting of the normal/anomalous operating sounds of six types of toy/real machines. Each recording is a single-channel (approximately) 10-sec length audio sampled at 16,000 Hz that includes both a target machine's operating sound and environmental noise. The environmental noise samples were recorded in several real factory environments. All the training data (normal) in development dataset and additional training dataset is used for training our models, and the performance is evaluated by using the test data in development dataset.

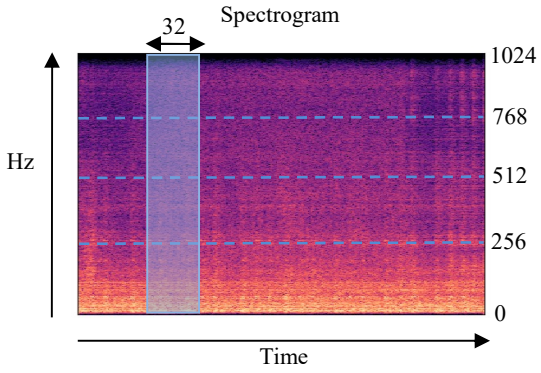

Figure 1: Audio preprocessing

1.2. Audio preprocessing

Follow [8], we load the audio clips with their raw sampling rate (16,000 Hz), and the spectrogram is adopted through a Short-Time Fourier Transform (STFT). We use librosa package [9] to apply STFT, the length of the window (nFFT) is 2046, the hop length is 512, so the height of the spectrogram is 1024 (1 + nFFT/2). Then we split the spectrogram into 32 columns for all data and finally a standardization is applied for data normalization.

2. PROPOSED SOLUTIONS

2.1. ArcFace Loss

We use ArcFace loss to obtain discriminative embedding features. ArcFace achieved state-of-the art performance in face verification/recognition task, we found that it is also powerful in sound recognition task. The loss function of ArcFace is formulated as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}} \quad (1)$$

ArcFace loss has two hyper-parameter: m and s. In this task, we fix the margin parameter m=0.05 and the re-scale factor s=30, respectively.

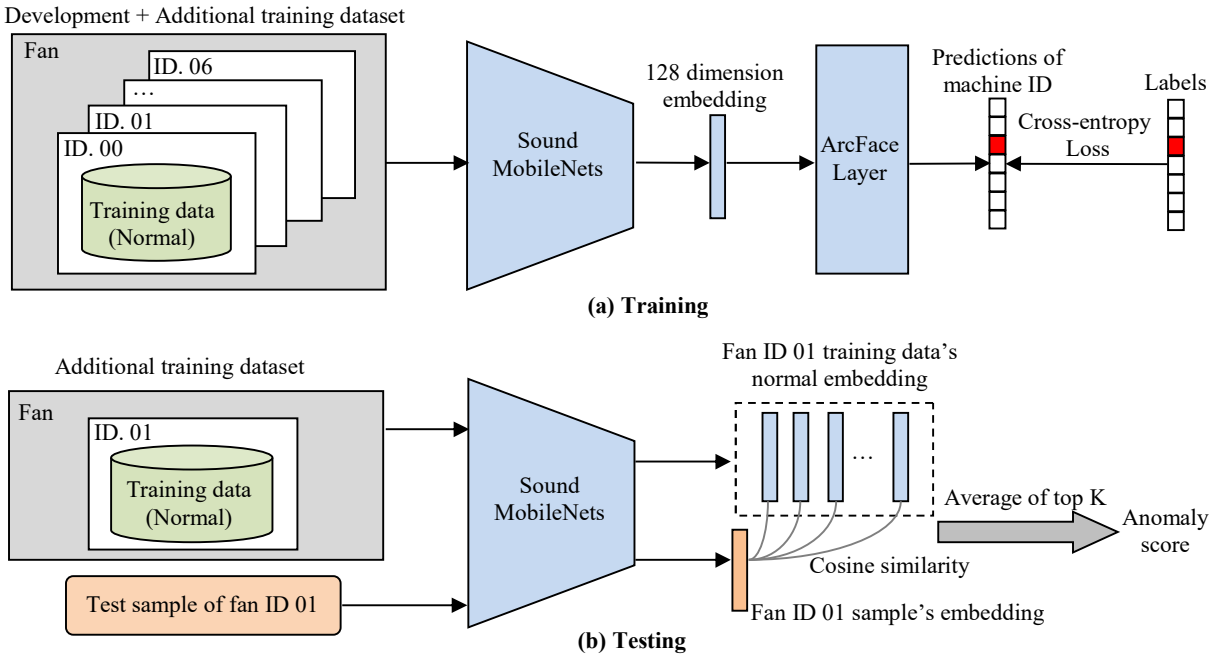

Figure 2: (a) We train machine ID classifier for each machine type. For example, we use the training data of all fan's IDs to train our fan ID classifier. (b) To calculate fan ID 01 test sample's anomaly score, the test sample's embedding is compared with all the fan ID 01's embedding of training data. We average the most similar K results as the similarity score. The final anomaly is calculated by "1-similarity score".

2.2. Network Architectures

We use the residual bottlenecks proposed in MobileNetV2 [10] as our main building blocks and a deeper network based on MobileFaceNet is used. Follow [4], we use PReLU [11] as the non-linearity and the embedding feature dimension is set to 128. In this task, we tried two networks, in our network 1, the network's input shape is (1024×32×1), which is the output shape of the audio preprocessing discussed in chapter 1.2. In network 2, we fold up the audio preprocessing results along the frequency axis to 4 channels, so the input shape is (256×32×4), and a Squeeze-and-Excite [12] block is inserted after the first 3x3 convolutional layer. The detailed structure of our networks are shown in Table 1 and Table 2. The column t refers to the expansion factor, c refers to output channels, n refers to the number of repetitions, s refers to stride.

Input	Operator	t	c	n	s
1024×32×1	conv3x3	-	64	1	2
512×16×64	depthwise conv3x3	-	64	1	1
512×16×64	bottleneck	2	64	5	2
256×8×64	bottleneck	4	128	1	2
128×4×128	bottleneck	2	128	6	2
64×2×128	bottleneck	4	128	1	2
32×1×128	bottleneck	2	128	2	1
32×1×128	conv1x1	-	512	1	1
32×1×512	linear GDConv32x1	-	512	1	1
1×1×512	linear conv1x1	-	128	1	1

Table 1: Architecture of network 1

Input	Operator	t	c	n	s
256×32×4	conv3x3	-	128	1	2
128×16×4	squeeze-and-excite	-	128	1	-
128×16×64	depthwise conv3x3	-	128	1	1
128×16×64	bottleneck	2	128	5	2
64×8×128	bottleneck	4	256	1	2
32×4×256	bottleneck	2	256	6	2
16×2×256	bottleneck	4	256	1	2
8×1×256	bottleneck	2	256	2	1
8×1×256	conv1x1	-	512	1	1
8×1×512	linear GDConv8x1	-	512	1	1
1×1×512	linear conv1x1	-	128	1	1

Table 2: Architecture of network 2.

Our network 1 use 1.02 million parameters and network 2 use 3.53 million parameters.

2.3. Training

We train our models on a single NVIDIA GTX1080Ti GPU. We use SGD to optimize models and learning rate is scheduled by a cosine annealing strategy [13]. The initial learning rate is set to 0.05 and 10 epochs for one cycle. We train our network from scratch without using any pre-trained model. All the hyper-parameters is summarized in Table 3.

Parameters for signal processing	
Sampling rate	16,000 Hz
FFT length	2046 pts
FFT hop length	512 pts
ArcFace loss parameters	
Margin Parameter (m)	0.05
Re-scale factor (s)	30
Cosine annealing strategy	
Initial learning rate	0.05
Epochs of one cycle	10
Other parameters	
Batch size	48
K (for embedding's similarity calculation)	10

Table 3: Summarization of hyper-parameters

3. EVALUATION RESULT

3.1. Evaluation Method

This task is evaluated with the area under the receiver operating characteristic (ROC) curve (AUC) and the partial-AUC (pAUC). The pAUC is calculated as the AUC over a low false-positive-rate (FPR) range [0, p]. In this task, p=0.1 is used.

3.2. Results

The AUC and pAUC on the development dataset are shown in table 4 and table 5. Baseline column shows the official data which is averaged on 10 independent trials. Ours column shows our best results on 3 independent trials.

Machine	**Baseline**		**Ours**	
Type	**AUC**	**pAUC**	**AUC**	**pAUC**
ToyCar	78.77%	67.58%	93.99%	89.23%
ToyConveyor	72.53%	60.43%	64.83%	57.23%
fan	65.83%	52.45%	88.12%	83.12%
pump	72.89%	59.99%	91.59%	81.52%
slider	84.76%	66.53%	99.99%	99.95%
valve	66.28%	50.98%	92.98%	84.56%

Table 4: Results of network 1

Machine	**Baseline**		**Ours**	
Type	**AUC**	**pAUC**	**AUC**	**pAUC**
ToyCar	78.77%	67.58%	93.56%	88.51%
ToyConveyor	72.53%	60.43%	68.57%	60.89%
fan	65.83%	52.45%	85.92%	80.59%
pump	72.89%	59.99%	92.17%	80.88%
slider	84.76%	66.53%	99.58%	97.84%
valve	66.28%	50.98%	90.05%	76.55%

Table 5: Results of network 2

4. CONCLUSIONS

This technique report briefly presents our ArcFace based anomalous sound detection MobileNets for the task 2 of DCASE2020 challenge. Our method significantly outperforms the baseline system and our network 1 use less than 1.1 million parameters.

5. REFERENCES

[1] http://dcase.community/workshop2020/.

[2] Koizumi, Y, et al. "Description and Discussion on DCASE2020 Challenge Task2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring." (2020).

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou.Arcface: Additive angular margin loss for deep face recognition. In CVPR, 2019

[4] Chen, Sheng, et al. "MobileFaceNets: Efficient CNNs for Accurate Real-time Face Verification on Mobile Devices." (2018).

[5] Koizumi Y , Saito S , Kawachi H U Y , et al. Unsupervised Detection of Anomalous Sound based on Deep Learning and the Neyman-Pearson Lemma[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2018, PP(99).

[6] Yuma Koizumi, Shoichiro Saito, Hisashi Uematsu, Noboru Harada, and Keisuke Imoto. ToyADMOS: a dataset of miniature-machine operating sounds for anomalous sound detection. In Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 308–312. November 2019.

[7] Harsh Purohit, Ryo Tanabe, Takeshi Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi. 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), 209–213. November 2019.

[8] Dong, Oh , and Y. Il . "Residual Error Based Anomaly Detection Using Auto-Encoder in SMD Machine Sound." Sensors 18.5(2018):1308-.

[9] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and Music Signal Analysis in Python," in Proceedings of the 14th Python in Science Conference, Kathryn Huff and James Bergstra, Eds., 2015, pp. 18 – 24.

[10] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks. CoRR, abs/1801.04381 (2018)

[11] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: CVPR (2015)

[12] Hu, Jie , et al. "Squeeze-and-Excitation Networks." IEEE Transactions on Pattern Analysis and Machine Intelligence PP.99(2017).

[13] I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with restarts. CoRR, abs/1608.03983 (2016)