A SPEAKER RECOGNITION APPROACH TO ANOMALY DETECTION

Technical Report

Jose A. Lopez [1], *Hong Lu* [1], *Paulo Lopez-Meyer* [2], *Lama Nachman* [1], *Georg Stemmer*[3], *Jonathan Huang*[4]

[1] Intel Corp, Intel Labs, 2200 Mission College Blvd., Santa Clara, CA 95054, USA, {jose.a.lopez, hong.lu, lama.nachman}@intel.com
[2] Intel Corp, Intel Labs, Av. Del Bosque 1001, Zapopan, JAL, 45019, Mexico, {paulo.lopez.meyer}@intel.com
[3] Intel Corp, Intel Labs, Lilienthalstraße 15, 85579, Neubiberg, Germany, {georg.stemmer}@intel.com
[4] Work done at Intel, 2200 Mission College Blvd., Santa Clara, CA 95054, USA, {jonathan.huang}@ieee.org

ABSTRACT

We discuss our unsupervised speaker-recognition-based submission to the DCASE 2020 Challenge Task 2. We found that a speaker-recognition approach enables the use of all the training data, even from different machine types, to detect anomalies in specific machines. Using this approach, we obtained AUCs close to, or greater than, 0.9 for 5 out of 6 machines. We also discuss the modifications needed to surpass the baseline score for the ToyConveyor data.

Index Terms— DCASE, anomaly detection, anomalous sounds, machine condition monitoring, machine health monitoring, speaker recognition.

1. INTRODUCTION

The DCASE 2020 Challenge Task 2 is concerned with identifying anomalous behavior from a target machine using sound recordings [1]. A major difference between this task and other DCASE tasks is that it is not supervised. Accordingly, the available training data only contains samples from the normal-state distributions.

In our submission, we used the data provided from the "development" and "additional" datasets [2, 3]. Although these data do not contain abnormal samples, they do contain other information – they contain the machine type and ID number. We leveraged this information to train a deep neural network (DNN) to identify which machine ID an input sample belongs to, similar to speaker-recognition approaches.

The DNN architecture used here is composed of a (Mel or STFT) spectrogram layer, followed by a 2D CNN encoder, followed (optionally) by stats-pooling layers, and capped with either a fully-connected (FC) or added margin softmax (AMS) layer [4]. We employ two scoring methods and take the best one for a given machine type.

2. METHODOLOGY

In this section we detail our implementation, DNN training strategy, and scoring methods.

Fan	Pump	Slider	Valve	ToyCar	ToyConv
STFT	MEL	MEL	MEL	MEL	MEL
encoder	encoder	encoder	encoder	encoder	encoder
stats pool	stats pool	stats pool	stats pool	stats pool	FC
AMS	AMS	AMS	AMS	AMS	

Table 1: High-level Architecture

2.1. Data Processing

The DCASE 2020 Task 2 dataset consists of 10s audio files that include the sound of the target machine and environmental noise. There are six types of machine categories. ToyCar and ToyConveyor are from the ToyADMOS dataset [5]. Valve, Pump, Fan, and Slider are from the MIMII dataset [6]. Within each machine category there are a number of machine IDs, for a total of 41 possible sound categories. The interested reader is referred to the dataset references for details on the recording procedures. However, all the audio files contain a single-channel and use a 16kHz sampling rate.

For spectral features, we used the package nnAudio to transform input audio into either a Mel or STFT spectrogram [7]. The optimal spectrogram settings varied with machine type. We provide these settings in Section 3.

2.2. Network Architectures

A high-level description of the architectures can be found in Table 1. All architectures utilized the 2D CNN encoder shown in Table 2. The encoder utilizes progressively smaller kernel sizes and ends with a max pool layer. The stats pooling was performed using a variant of the Xvector from [8] discussed in [4]. Table 3 shows the details of the variant used here.

Except for ToyConveyor, which used a simple fully-connected layer as the final layer, the other models used the additive margin softmax layer discussed in [4].

2.3. Training Strategy

All the models use the same training strategy, except for the ToyConveyor model which we discuss separately. At training time, a

Layer	Input	Output	Kernel	Stride
conv 2D	1	64	7x7	1x1
batchnorm 2D	64	64		
relu	64	64		
conv 2D	64	32	5x5	1x1
relu	32	32		
conv 2D	32	6	3x3	1x1
max pool	6	6	2x2	2x2
relu	6	6		

Table 2: Encoder

Layer	Input	Output	Kernel	Stride
batchnorm 1D	348			
conv 1D	348	348	3x3	1x1
relu	348	348		
batchnorm1D	348			
conv 1D	348	348	3x3	1x1
relu	64	64		
batchnorm 1D	348			
conv 1D	348	348	1x1	1x1
relu	348	348		
batchnorm 1D	348			
conv 1D	348	1500	1x1	1x1
batchnorm 1D	3000			
fully connected	3000	128		
dropout(0.1)				

Table 3: Xvector Short

contiguous 10/7 second clip was randomly sampled from the training files. Batches of 64-128 such samples were used during each epoch, for between 100 and 200 epochs to ensure all the data are sampled.

At the output of the spectrogram layer, before inputting to the encoder, we subtracted the column-wise mean, and divided by the column-wise standard deviation, of all the training spectrograms of the same machine type.

We used a different strategy to obtain the best result from the ToyConveyor data. We divided each 10s sample into 7 parts and used the 7 parts as a batch. Thus, each batch was composed of data from the same ToyConveyor machine ID[1]. With a probability 1/2, we simulated anomalies by corrupting 1 of the 7 parts by linearly combining the part spectrogram with a spectrogram from another ToyConveyor machine ID according to (1).

$$S_i = \lambda S_j + (1 - \lambda) S_i \tag{1}$$

where S_i is the spectrogram of ToyConveyor ID i and $i \neq j$. In contrast to the so-called *mix-up* data augmentation method, we did not randomly select λ – we fixed λ to 0.03 and randomly selected the machine ID j. Selecting a larger λ resulted in the model (too) easily identifying the anomaly, leading to overfitting.

For all models, we used a categorical cross-entropy loss function with l_1 regularization on the encoder weights and the Adamax optimizer with the default learning rate.

[1] In contrast to other models, the output of the ToyConveyor anomaly detector only has 7 classes, one for each ToyConv. ID and 1 *Other* class.

2.4. Scoring

We used two scoring methods. The first is simply 1 minus the softmax probability of the specific machine ID. Clearly, if the model is certain a sample belongs to machine ID i, the ith output will be close to 1, resulting in a lower anomaly score. Conversely, as the uncertainty increases, so will the anomaly score. In the ToyConveyor case, since the model output categories also include an *Other* class, we add this softmax probability to the anomaly score as well.

The second scoring method is the cosine distance between the average normal embedding[2], recorded at training time, and the embedding of the test sample computed at test time. Generally, the two scoring methods did not produce very different scores. However, we selected the scoring method that produced the best AUC + pAUC.

3. RESULTS

We summarize the results of our work in Table 4. Using our approach, the ToyConveyor case proved the most difficult, followed by Fan. The Slider and Valve were the easiest to obtain good results for, followed by ToyCar and Pump.

	Fan	Pump	Slider	Valve	ToyCar	ToyConv
batch size	64	64	64	128	64	7
no. Mels	128	256	128	128	128	128
no. FFT	1024	1024	1024	1024	1024	1024
hop	512	80	80	512	80	512
fmin	1	100	10	0	10	0
fmax	4000	7700	7700	8000	4000	4000
scoring	cos dist.	cos dist.	cos dist.	softmax	softmax	softmax
AUC	0.8823	0.9321	0.9997	0.9989	0.9573	0.7417
pAUC	0.8057	0.8619	0.9982	0.9941	0.9032	0.6586

Table 4: Scoring Results

4. CONCLUSIONS

We have outlined our speaker-recognition approach to the DCASE 2020 Challenge Task 2 which uses the machine IDs themselves to make an unsupervised problem supervised. It is our intuition that this approach succeeded because the spectral content of these machines is sufficiently similar to make the task of separating the machine IDs challenging. This was less so for the Toy- classes, especially for ToyConveyor. For example, in one experiment we classified the ToyConveyor IDs and used data from the remaining machines as Other, during training both the training and validation accuracies quickly exceeded 99%. This led us to look to data augmentation methods for ToyConveyor, as described in Section 2.3. Consequently, we expect that this approach may not work in cases where the spectral content of the machine sounds differs greatly.

5. REFERENCES

[1] 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*

[2] The embedding is the 128D output of the stats pooling layer.

e-prints: 2006.05822, June 2020, pp. 1–4. [Online]. Available: https://arxiv.org/abs/2006.05822

[2] ——, "Dcase 2020 challenge task 2 development dataset," Mar. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3678171

[3] ——, "DCASE 2020 Challenge Task 2 Additional Training Dataset," Apr. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3727685

[4] J. Huang and T. Bocklet, "Intel Far-Field Speaker Recognition System for VOiCES Challenge 2019," in *Proc. Interspeech 2019*, 2019, pp. 2473–2477. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2894

[5] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. 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)*, November 2019, pp. 308–312. [Online]. Available: https://ieeexplore.ieee.org/document/8937164

[6] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. 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)*, November 2019, pp. 209–213. [Online]. Available: http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Purohit_21.pdf

[7] K. W. Cheuk, H. H. Anderson, K. Agres, and D. Herremans, "nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolution neural networks," *ArXiv*, vol. abs/1912.12055, 2019.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.