Article

UNSUPERVISED ANOMALOUS SOUND DETECTION FOR MACHINE CONDITION MONITORING USING CLASSIFICATION-BASED METHODS

Yaoguang Wang [1], Yaohao Zheng [2],Yunxiang Zhang [2],Yongsheng Xie [*],Sen Xu [*] ,Ying Hu [2], Liang He [1,2,†]

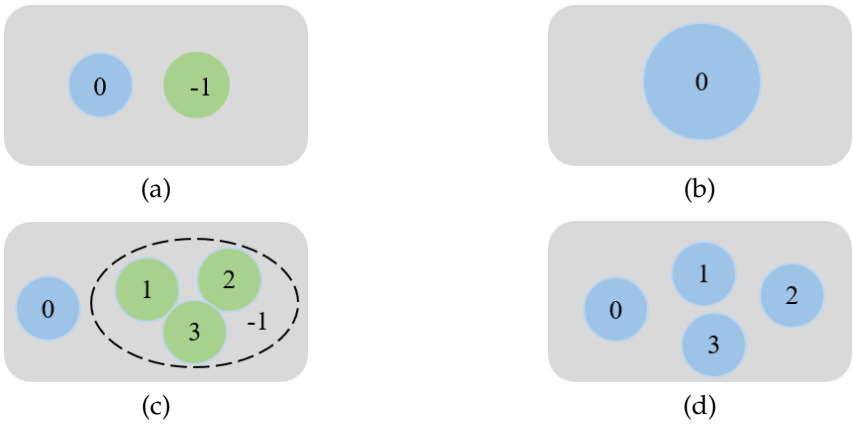


Figure 1. Different data settings. (b) stands for unsupervised settings, (a), (c) and (d) represent three forms of supervised setting respectively. Blue area denotes normal samples, green denotes anomalous or outlier samples, grey denotes unavailable samples.

[1] Department of Electronic EngineeringDepartment of Electronic Engineering, and Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
[2] School of Information Science and Engineering, Xinjiang University, Urumqi, China
[*] Engaged in power system operation control management and power big data
[†] Correspondence: heliang@mail.tsinghua.edu.cn; Tel.: (+86)010-62771680

Abstract: The task of unsupervised anomalous sound detection (ASD) is challenging for detecting anomalous sounds from a large audio database without any annotated anomalous training data. Many unsupervised methods were proposed, but previous works have confirmed that the classification-based models far exceeds the unsupervised models in ASD. In this paper, we adopt two classification-based anomaly detection models: (1) Outlier classifier is to distinguish anomalous sounds or *outliers* from the normal; (2) ID classifier identifies anomalies using both the confidence of classification and the similarity of hidden embeddings. We conduct experiments in task 2 of DCASE 2020 challenge, and our ensemble method achieves an averaged area under the curve (AUC) of 95.82% and averaged partial AUC (pAUC) of 92.32%, which outperforms the state-of-the-art models.

Keywords: Unsupervised anomalous sound detection, classification-based model, Outlier classifier, ID classifier

Citation: Lastname, F.; Lastname, F.; Lastname, F. UNSUPERVISED ANOMALOUS SOUND DETECTION FOR MACHINE CONDITION MONITORING USING CLASSIFICATION-BASED METHODS. *Preprints* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

1. Introduction

ASD is the task to identify whether the sound is normal or anomalous. This technique is commonly used in audio surveillance [1][2], machine condition monitoring [3], etc. In the case of machine condition monitoring, we hope to monitor the operation of the machine through acoustic characteristics, because sound-based anomaly detection is flexible and the cost can be reduced by bringing the microphone close to different machines to detect anomalies. It can avoid the huge loss caused by serious failure that find out the early fault of the machine and carry out maintenance effectively.

ASD includes supervised-ASD and unsupervised-ASD. For supervised-ASD, the training data contains both normal and anomalous sounds as shown in Fig.1-(a), the

supervised binary classification model is suitable for anomaly detection. Since the machine works normally most of the time, it is difficult to collect a large number of anomalous sounds, and the pattern of anomalous sounds emitted from a target machine is not clear. Only normal sounds are provided as training data as shown in Fig.1-(b), which makes ASD an unsupervised task. The "*Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring*" task of *Detection and Classification of Acoustic Scenes and Events 2020* (DCASE 2020) [4], has attracted many researchers to submit systems, and their systems ranked on public data sets [5][6].

Some methods use unsupervised models to learn the essential characteristics of normal sounds so that find the subspace where the normal samples are located, and the sounds outside the subspace are judged as anomalous. [4] adopts an autoencoder as the anomaly detector, the model is trained with reconstruction error on normal samples and the anomaly scores are derived from the reconstruction error. An x-vector based model using L3-Net embeddings for anomalous sound detection has been proposed in [8]. [9] combines the Siamese Network feature extractor with KNN anomaly detector, the Siamese Network extracts required features and then the KNN trained on the features performs anomaly detection. [10] adopts Masked Autoregressive Flows to learn the density of normal sounds and uses the negative log-likehood as the anomaly score. Some works have demonstrated that the use of machine ID information significantly improves the ASD performance [11-15]. In Fig.1-(c) and Fig.1-(d), data sets from other machine IDs are added to the training data. [13] divides the training data into two categories, the normal sounds of a specific machine ID are regarded as positive samples, and the normal sounds of other machines IDs are considered as negative samples. [11][14] treat the different machine IDs as different categories, and [12] adds anomalous samples through data augmentation.

In this paper, we adopt two methods for anomaly detection. The first method is to train an Outlier classifier based on Fig.1-(c) setting. The model distinguishes anomalies from the normal, and its output is directly used as the anomaly score of the unseen sound. Another method trains an ID classifier based on Fig.1-(d) setting, its output is the probability that the unseen sound belongs to the corresponding machine ID, and its opposite number is taken as the anomaly score. At the same time, we calculate the similarity of embeddings between the normal sounds and the unseen sounds for anomaly detection.

2. Proposed Method

[11-15] show that the supervised classifier substantially outperforms the unsupervised methods across most machine types in anomalous sound detection. In these works, unsupervised anomaly detection is reframed as a supervised classification problem. CNN has demonstrated its good performance for audio classification, such as ResNet[16], MobileFaceNet[17], MobileNetV3[18]. In this section, we adopt two classifiers based on above popular architectures to obtain decision boundary for identifying whether the unseen sound is normal or anomalous.

2.1. Outlier classifier for binary classification

In order to solve anomaly detection problem in a supervised manner, we obtain training set containing normal and anomalous samples according to Fig.1-(c). For each specific machine ID, we assign the audio clips of this machine ID as positive samples and the other machine IDs in the same domain as negative samples.

2.1.1. Attention-based audio classification network

[13] adopts this network in anomalous sound detection by changing the filters sizes slightly and outperforms the most methods across all machine types and IDs. In this paper, we add Convolutional Block Attention Module (CBAM) [19] which contains of Channel-attention module (CAM) and Spatial-attention module (SAM), they are concerned about "*what*" and "*where*" the audio events happen respectively. CAM can be regarded as a process of selecting relevant semantic features based on context semantics. When the

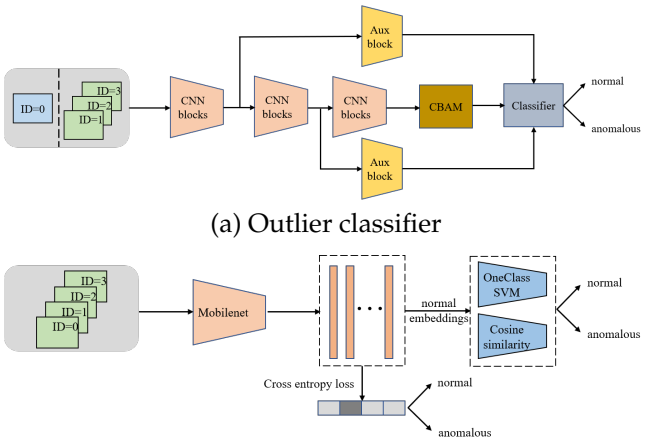


Figure 2. (a) The Outlier classifier distinguishs the outliers which are considered as anomalies from the normal, and it directly outputs the anomaly scores of unseen sounds. (b) The ID classifier identifies different machine IDs. For the ID classifier, we use two methods for anomaly detection as shown in Fig.2-(b), the first is to calculate the similarity between unseen sounds and normal sounds using embeddings extracted from the hidden layer, another method uses the confidence of classification.

network wants to predict the "*fan*" audio, CAM will assign larger weight to the feature map containing the "*fan*" spectrum structure. The SAM will locate the segments of "*fan*" on the feature map, thereby filtering out background noise. So attention module is helpful for accurately expressing the characteristics of normal sounds.

The feature map $\mathbf{X}$ ($C \times H \times W$) passes through CAM and then SAM. CAM calculates the weight ($C \times 1 \times 1$) of each channel, and multiplies the weight with the original feature map to obtain a weighted feature map. In order to obtain the weight of the channel dimension, this module calculates the average value and maximum value of each channel respectively with *avgpool* and *maxpool*, and feeds them to a common multi-layer perceptron, and then the two outputs are added together and normalized by the sigmoid function to get the final weight. CAM is defined as:

$$\begin{aligned}\mathbf{W}_C = \sigma(\mathbf{W}_2(\delta(\mathbf{W}_1 \cdot avgpool(\mathbf{X}))) \\ +\mathbf{W}_2(\delta(\mathbf{W}_1 \cdot maxpool(\mathbf{X}))))\end{aligned} \tag{1}$$

where $\mathbf{W}_1 \in \mathbb{R}^{C \times \frac{C}{r}}$, $\mathbf{W}_2 \in \mathbb{R}^{\frac{C}{r} \times C}$ represent FC layers, $\delta(\cdot)$, $\sigma(\cdot)$ represent ReLU and sigmoid function respectively, r denotes the scaling ratio.

SAM calculates the average and maximum values of different channels on the same point to obtain weights ($1 \times H \times W$) with $avgpool'$ and $maxpool'$, concatenates them along the channel dimensions, and then the weights passes a convolutional layer and sigmoid function to get the final weights. The weights is multiplied by each channel on the time-frequency domain to obtain a weighted feature map. SAM is defined as:

$$\mathbf{W}_S = \sigma\big(\mathbf{W}\big[avgpool'(\mathbf{X}); maxpool'(\mathbf{X})\big]\big) \tag{2}$$

where $\mathbf{W}$ denotes a convolutional layer. CAM and SAM are connected in a sequential manner,

$$\begin{aligned}\mathbf{Y} &= \mathbf{W}_C(\mathbf{X}) \otimes \mathbf{X} \\ \mathbf{Z} &= \mathbf{W}_S(\mathbf{Y}) \otimes \mathbf{Y}\end{aligned} \tag{3}$$

where $\otimes$ represents element-wise multiplication.

Table 1: Architecture of MobileNet

Operator	exp size	#*out*	SE	NL	*s*
conv3×3	-	32	-	HS	2
bneck3×3	64	32	-	RE	1
bneck3×3	64	32	-	RE	2
bneck3×3	64	32	-	RE	1
bneck3×3	64	32	✓	RE	2
bneck3×3	64	32	✓	RE	1
bneck3×3	128	64	✓	RE	1
bneck3×3	128	64	-	HS	2
bneck3×3	128	64	-	HS	1
bneck3×3	128	64	-	HS	1
bneck3×3	128	64	-	HS	1
bneck3×3	256	128	✓	HS	1
bneck3×3	256	128	✓	HS	1
bneck3×3	256	128	✓	HS	1
bneck3×3	256	128	✓	HS	2
bneck3×3	256	128	✓	HS	1
conv1x1	-	512	-	HS	1
GDConv32×1	-	512	-	-	1
conv1×1	-	128	-	-	1

2.1.2. Auxiliary classifiers for anomaly detection

Since we have defined the outlier datas of normal sounds as the anomalous, the outputs of the classifier are used as the anomaly scores. The network composes of multiple convolution blocks as shown in Fig.2-(a). The front stages have a larger kernel size and more pooling operations to reduce the feature dimension, while the back stages have a smaller kernel size and fewer pooling operations to maintain the resolution of the features, thereby limiting the receptive fields to capture local features [20]. In order to improve the classification ability of the network, we adopt the strategy of auxiliary classification.The Aux block is composed of two parts: the first part is global pooling and the second part is reshape. Each stage is followed by an auxiliary classifier, and a CBAM module is added in the last stage. We use multiple-level features at the same time by integrating the outputs of auxiliary classifiers according to the weights, where the back classifiers have greater weights,

$$p = (w_1 \cdot p_1 + w_2 \cdot p_2 + w_3 \cdot p_3) \tag{4}$$

where w_i, p_i $(i = 1, 2, 3)$ denote the weight and the output of the i-th classifier respectively, p denotes the final output of the network and is used as the anomaly score. We believe that the deeper the features, the stronger the expressiveness and the higher the accuracy of classification. In equation 4, $w_1 < w_2 < w_3$. The specific weight value is set based on the training set according to the trust degree.

2.2. ID classifier for multiple classification

We train an ID Classifier to recognize different machine IDs of the same machine type with recordings from all the machine IDs. The model uses the embeddings output by the hidden layer of the model to determine whether the audio is anomalous, and uses the classification confidence of the network to identify anomalies.

2.2.1. MobileNet-based Audio classification network

In this section, we introduce a model that combines the characteristics of MobileFaceNet [17] and MobileNetV3 [18]. We adopt MobileNetV3 as the main body of the

Table 2: AUC (%) and pAUC (%) for each machine

	Fan AUC(pAUC)	Pump AUC(pAUC)	Slider AUC(pAUC)	Valve AUC(pAUC)	Toy-car AUC(pAUC)	Toy-conveyor AUC(pAUC)	Average AUC(pAUC)
Baseline [4]	82.80(65.80)	82.37(64.11)	79.41(58.87)	57.37(50.79)	80.14(66.17)	85.36(66.95)	77.91(62.12)
Hayashi [7]	92.72(80.52)	90.63(73.61)	95.68(81.48)	97.43(89.69)	91.75(83.97)	**92.10**(76.76)	93.39(81.01)
Wilkinghoff [8]	93.75(80.68)	93.19(81.10)	95.71(79.45)	94.87(83.58)	94.06(86.80)	84.22(69.12)	92.63(80.12)
Durkota [9]	90.74(83.38)	88.70(75.97)	96.18(87.49)	97.48(92.46)	94.32(89.01)	64.38(53.79)	88.63(80.35)
Haunschmid [10]	91.48(74.32)	92.30(72.14)	89.74(76.43)	81.99(69.82)	81.50(67.00)	88.01(70.52)	87.50(71.71)
Giri [11]	94.54(84.30)	93.65(81.73)	97.63(89.73)	96.13(90.89)	94.34(89.73)	91.19(73.34)	94.58(84.95)
Daniluk [12]	99.13(96.40)	95.07(90.23)	98.18(91.98)	90.97(77.41)	93.52(83.87)	90.51(77.56)	94.56(86.24)
Primus [13]	96.84(95.24)	**97.76**(92.24)	97.29(88.74)	90.15(86.65)	86.37(83.83)	88.28(79.15)	92.78(87.64)
Inoue [14]	98.84(94.89)	94.37(88.27)	95.68(83.09)	**97.82**(**94.93**)	93.16(87.69)	87.41(72.03)	94.55(86.82)
Zhou [15]	99.79(98.92)	95.79(**92.60**)	99.84(99.17)	91.83(84.74)	**95.60**(**91.30**)	73.61(64.06)	92.74(88.47)
Outlier classifier	97.53(95.64)	97.34(91.54)	99.04(95.14)	92.00(89.05)	88.11(86.53)	89.80(**80.61**)	93.97(89.75)
ID classifier	99.94(99.80)	95.01(90.89)	99.09(95.91)	95.82(93.58)	91.33(86.57)	71.32(60.09)	92.09(87.81)
ensemble	**99.96**(**99.84**)	97.35(91.58)	**99.97**(**99.83**)	95.82(93.58)	92.02(88.50)	89.80(**80.61**)	**95.82**(**92.32**)

network structure and modify the network parameters as shown in Table 1. *#out* refers to the number of out channels, SE refers to Squeeze-And-Excite block, HS refers to h-swish, RE refers to ReLU, and *s* refers to stride.

The model inherits the advantages of MobileNetV3. Depthwise separable convolutions contain spatial filtering and feature generation, which has fewer parameters and lower computational cost compared with conventional convolution. The linear bottleneck and inverted residual structure map features into high-dimensional space to increase the expressiveness of the network. The squeeze and excitation is integrated as attention module. We use h-swish or ReLU as the non-linearity. We also use global depthwise convolution (GDConv) to replace global pooling like MobileFaceNet.

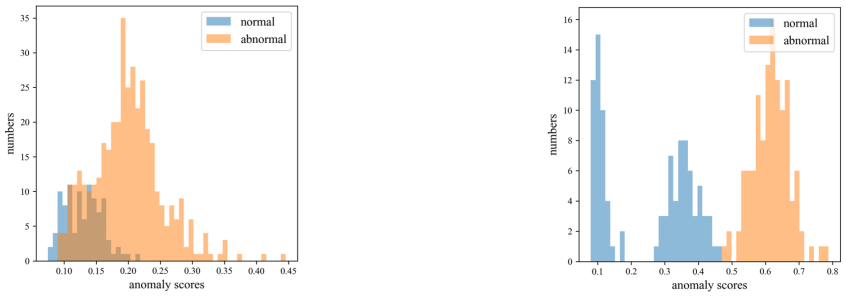
2.2.2. Anomaly detection in multiple ways

For the ID classifier, we use two methods for anomaly detection as shown in Fig.2-(b). The first method is to use the embeddings output by the hidden layer of the network to calculate the similarity between the unseen sound and the normal sound, and the similarity is calculated in two ways: angle (Cosine similarity) or distance (OneClassSVM). Another method uses the softmax probability output by the network as the probability that the sample belongs to the corresponding machine ID, and its opposite number is used as the anomaly score. We apply different methods on different machines.

3. Results

The two trained models have different definitions for anomaly detection. The Outlier classifier is trained for distinguishing anomalies from normal sounds, so the outputs of the model are directly used as the anomaly scores. We also apply the same supervised settings shown in Fig.1-(c) as the Outlier classifier to the network in Fig.2-(b), but it doesn't perform well. Different from the Outlier classifier, we train the ID Classifier to recognize different IDs of the same machine type and learn the hidden characteristics of the normal sounds. We calculate the similarity between the embeddings of unseen sounds and corresponding normal sounds for anomaly detection in two ways: angle (Consine similarity) and distance (OneClassSVM), and the final anomaly score is calculated as "1-similarit". It is worth noting that OneClassSVM is suitable for anomaly detection of the machine "*ToyCar*", Consine similarity is suitable for other machine types according to our experiments.

The comparison of our methods against other advanced approaches on the evaluation set of DCASE 2020 task 2. We can find our methods performs well on different



(a) "*Valve*" ID : 5 (b) "*Valve*" ID : 1

Figure 3. Distribution of anomaly scores of the machine "*Valve*". (b) shows that the model completely distinguishes normal and anomalous sounds, but (a) does not.

machines, the Outlier classifier achieves the average AUC of 93.97% and average pAUC of 89.75% and the ID classifier achieves the average AUC of 92.09% and average pAUC of 87.81%. We summarize the advanced systems on DCASE 2020 task 2 into two categories: the classification-based models [11-15] and unsupervised-based models [7-10]. The unsupervised-based models like autoencoder, PCA, KNN and normalizing flow only use normal sounds of the target machine as the training set. The classification-based models add another data sets to create a training set includeing multiple categories, and convert unsupervised anomaly detection to supervised or semi-supervised tasks. We can see that the classification-based models outperform the unsupervised-based models by a large margin, outlier samples can greatly help the model to recognize anomalous sounds. The experimental results confirm that the machine ID information is benefit to accurately determine the classification boundary of the classifier and extract more distinguishing hidden features.

Table 2 shows that even the best models cannot perform best on all machine types and the performance of different machines of the same type varies greatly as shown in Fig.3. So we apply the model ensemble strategy. For the target machine, we choose the model with better performance on development data set. Our ensemble method achieves the highest average AUC of 95.82% and average pAUC of 92.32%, even outperforms all other methods on some machine types such as "*fan*".

4. Conclusions

In this paper, we introduce two classification-based models for the anomaly detection and conduct experiments in task 2 of DCASE 2020 challenge. Both models are trained with only normal sounds to learn distribution characteristics of the normal sounds like most unsupervised methods, and then the unseen sounds are identified as the anomalous when they are outliers of normal sounds. Different from the unsupervised methods, we also use samples from other machine IDs to train the models in a supervised manner, so that the classification-based method can be used to find the decision boundary between the normal and outliers. The use of machine ID information helps to determine the decision boundary accurately and improve the ASD performance. Table 2 demonstrates that the classification-based models outperform the unsupervised-based models significantly across all machine types, and our models outperform the state-of-the-art models, achieving an averaged AUC of 95.82% and an averaged pAUC of 92.32% with an ensemble strategy.

References

1. P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio Surveillance of Roads: A System for Detecting Anomalous Sounds," *IEEE Trans. ITS*, pp.279–288, 2016.
2. S. Ntalampiras, I. Potamitis, and N. Fakotakis, "Probabilistic Novelty Detection for Acoustic Surveillance Under Real-World Conditions," *IEEE Trans. on Multimedia*, pp.713–719, 2011.
3. Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, "Optimizing Acoustic Feature Extractor for Anomalous Sound Detection Based on NeymanPearson Lemma," in Proc. of EUSIPCO, 2017.

4. 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, 1–4. June 2020.
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), 308–312. November 2019.
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), 209–213. November 2019.
7. T. Hayashi, T. Toshimura, and Y. Adachi, "Conformer-based ID-aware Autoencoder for Unsupervised Anomalous Sound Detection," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
8. H. Wilkinghoff, "Anomalous sound detection with look, listen, and learn embeddings," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
9. K. Durkota, M. Linda, M. Ludvik, J. Tozicka, "Euron-Net: siamese network for anomaly detection," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
10. V. Haunschmid, P. Praher, "Anomalous sound detection with masked autoregressive flows and machine type dependent postprocessing," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
11. R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, A. Krishnaswamy, "Unsupervised anomalous sound detection using self-supervised classification and group masked autoencoder for density estimation," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
12. P. Daniluk, M. Goździewski, S. Kapka, and M. Kośmider, "Ensemble of Auto-encoder based and WaveNet like Systems for Unsupervised Anomaly Detection," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
13. P. Primus, "Reframing Unsupervised Machine Condition Monitoring as a Supervised Classification Task with OutlierExposed Classifiers," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
14. T. Inoue, P. Vinayavekhin, S. Morikuni, S. Wang, T. H. Trong, D. Wood, M. Tatsubori, and R. Tachibana, "Detection of Anomalous Sounds for Machine Condition Monitoring using Classification Confidence," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
15. Q. Zhou, "ArcFace based Sound MobileNets for DCASE 2020 Task 2," *Tech. report in DCASE2020 Challenge Task 2*, 2020.
16. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), Las Vegas, NV, pp. 770-778, 2016,
17. S. Chen, Y. Liu, X. Gao, and Z. Han, "MobileFaceNets: Efficient CNNs for accurate realTime face verification on mobile devices," in Proceedings of the 13th Chinese Conference, CCBR 2018, Urumqi, China, August 11-12, 2018.
18. A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le and H. Adam, "Searching for MobileNetV3," *International Conf. on Computer Vision 2019* (ICCV2019), Seoul, Koren, 2019.
19. S. Woo, J. Park, J. Y. Lee and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, pp. 3-19, September 8-14, 2018.
20. K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptivefield-regularized CNN variants for acoustic scene classification," CoRR, vol. abs/1909.02859, 2019.
21. B. McFee, C. Raffel, D. Liang, D. P.W. Ellis, M. McVicar, E. Battenberg and O. 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.
22. I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with restarts," CoRR, abs/1608.03983 (2016).