UNSUPERVISED ANOMALOUS SOUND DETECTION USING SELF-SUPERVISED CLASSIFICATION AND GROUP MASKED AUTOENCODER FOR DENSITY ESTIMATION

Technical Report

Ritwik Giri[*]*, Srikanth V. Tenneti*[*]*, Fangzhou Cheng, Karim Helwani,*
Umut Isik, Arvindh Krishnaswamy

Amazon Web Services, Palo Alto, CA, USA

ABSTRACT

This technical report outlines our solutions to Task 2 of the DCASE 2020 challenge, *Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring*. The objective is to detect audio recordings containing anomalous machine sounds in a test set, when the training dataset itself does not contain any examples of anomalies. Our approaches are based on an ensemble of a novel density estimation based anomaly detector (Group Masked Autoencoder for Density Estimation (GMADE)) and self-supervised classification based anomaly detector.

***Index Terms*—** Unsupervised anomaly detection, machine condition monitoring, self-supervision.

1. INTRODUCTION

The IEEE Audio and Acoustic Signal Processing Society's 2020 Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge features anomalous sound detection in machines as one of the tasks [1]. Given a training set containing audio recordings solely from healthy machines, the task is to identify recordings from defective machines in the test set.

The challenge dataset consists of two recent machine audio datasets, ToyADMOS [2] and MIMII [3]. Data from six types of machines, namely toy-car, toy-conveyor, fan, pump, slider and valve, have been provided. The former two are from toy machines, while the rest are from real machines. For each machine type, data from 7 to 8 machine IDs has been provided. Defects of various kinds are introduced in the machines to record the anomalous sounds in the test set.

Our submission includes ensemble of two major components/approaches for anomaly detection.

First approach is based on a neural density estimator model, Group-Masked Autoencoder. This density estimator has been used to estimate the probability distribution that models the normal audio recordings during training time. During inference we use the negative log likelihood of the test point as an anomaly score to detect anomalies.

Our second approach leverages the idea of self-supervised classification to extract representations of the data. Specifically, for each machine type, we train classifiers based on several popular architectures from image classification literature, such as: MobileNetV2 [4], and ResNet [5], on collated data from all the machine IDs, towards the following tasks:

1. Being able to identify the machine ID a sample came from,
2. Being able to distinguish a sample from a set of synthetically perturbed versions of itself.

[*] Equal contribution.

2. PROPOSED APPROACH

2.1. Group Masked Autoencoder (Group-MADE)

Our density estimation method builds on previous work on Masked Autoencoder for Distribution Estimation (MADE). We provide a brief description of MADE below. More details about MADE can be found in original publication [6].

In [6], the authors propose a simple way of adapting an autoencoder architecture to develop a competitive and tractable neural density estimator. The key idea lies in masking the weighted connections between layers of a standard autoencoder to convert it into a tractable density estimator. Authors show that by designing appropriate masks, the output of the autoencoder can be interpreted in an autoregressive (AR) manner for a given ordering of inputs, i.e., each input dimension is reconstructed solely from the dimensions preceding it in the ordering. Multiple layers with non-linearity can be added in this structure, which will result in a highly capable neural density estimator.

By using MADE, probability density of the vectorized input data, $\mathbf{x}$ is calculated by means of the decomposition according to the probability chain rule. In an autoregressive setting this will be,

$$p(\mathbf{x}) = \prod_{d=1}^{D} p(x_d|\mathbf{x}_{<d}) \tag{1}$$

Hence, in the autoencoder output, each dimension can be interpreted as one of the D conditional probability distributions as shown above, and each output unit $\hat{x}_d$ only depends on the previous input units, $\mathbf{x}_{<d}$, and not the other units, $\mathbf{x}_{\geq d} = [x_d, ..., x_D]^T$. In our work, we parameterize each conditional distribution as a mixture of C Gaussians, i.e., the autoencoder outputs mean, variance and the mixture component probabilities. E.g., for a D dimensional input, the number of the model outputs will be, $D \times C \times 3$. For all our experiments, we set $C = 10$. This model is trained by minimizing the negative log likelihood for all training data points,

$$Cost = -\log p(\mathbf{x}) = \sum_{d=1}^{D} -\log p(x_d|\mathbf{x}_{<d}) \tag{2}$$

For the problem in hand, following the baseline model as provided by the challenge organizers, 5 Mel spectrum frames have been concatenated to produce $5 \times 128 = 640$ dimensional input vector, where 128 Mel bands have been used. Since for this task, we are

interested in the autoregressive ordering across frames (not across each dimension of the input), we design a Group MADE architecture, where the joint distribution can be decomposed as conditionals over groups/frames, instead of probability distributions over the individual dimensions. Also, note that in our architecture, the Mel bins in one frame are conditionally independent when conditioned on all previous frames.

Let us assume, that one input sample can be represented as $\mathbf{t} = [\mathbf{t_{i+1}}, \mathbf{t_{i+2}}, ..., \mathbf{t_{i+5}}]^T \in \mathbb{R}^{640\times 1}$, where i^{th} frame is $\mathbf{t_i} \in \mathbb{R}^{128\times 1}$. Hence the joint density will be decomposed as,

$$p(\mathbf{t}) = \prod_{i=1}^{5} p(\mathbf{t_i}|\mathbf{t}_{<i}) = \prod_{i=1}^{5}\prod_{j=1}^{128} p(t_{ij}|\mathbf{t}_{<i}) \tag{3}$$

Hence, all the Mel bins in an output frame $\mathbf{t}_i$ depends on all the Mel bins from previous frames but not on other units, i.e., not on Mel bins of the i-th frame, or on the Mel bins of the future frames. Because of this group masking nature, we name our approach as Group Masked Autoencoder for Density Estimation (Group MADE).

So far, we have assumed that the conditionals modeled by Group MADE were consistent with the causal frame ordering, but in our submission we use, the following ordering mimicking deterministic approach proposed in IDNN [7]. In this case we predict the middle frame conditioned on 4 other frames, i.e.,

$$p(\mathbf{t}) = p(\mathbf{t}_3|\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_4, \mathbf{t}_5)p(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_4, \mathbf{t}_5) \tag{4}$$

The proposed Group MADE model is trained using negative log likelihood as cost function, using all the normal training data across all IDs for a specific machine. During inference we use the negative log likelihood as anomaly score for each test sample, and report AUC numbers. We use a fully connected network as the architecture where the number of hidden layers and the corresponding hidden units in each layer follows this structure: $[128, 128, 128, 128, 32, 128, 128, 128, 128]$. Finally, the output layer has $640 \times 10 \times 3 = 19200$ units. We use Adam optimizer with 0.001 learning rate for training.

Following the baseline model, each input 10s file is split into frames of length 64ms, with hop length of 32ms between frames. 1024-FFT and 128 Mel bins are used to featurize each frame. 5 frames are concatenated, resulting in $5 \times 128 = 640$ dimensional input.

2.2. Self-Supervised Classification

Within the framework of unsupervised representational learning, self-supervision involves withholding certain aspects of the data, and tasking a network to predict it. The features learned by such a network are then used for further downstream tasks.

Self-supervision using classification tasks has been previously used for detecting anomalies in [8, 9, 10]. In these works, the learning task involves networks to discriminate between multiple geometric transformations, including rotations, flipping and translations, applied to images. Another approach is presented in [11], where data is transformed onto a finite number of subspaces, before learning a feature mapping that maximizes the difference between inter-class and intra-class separations. We employ a different strategy here. We leverage machine ID metadata, combined with different types of audio-inspired data augmentations to set up classification tasks. Specifically, for each machine type, we train two popular architectures from image classification domain on normal data from all the machine IDs to:

1. Identify the machine ID of an audio sample. Apart from the provided samples, we also consider randomized linear combinations of the existing machine IDs to simulate new synthetic machine IDs.
2. Distinguish a sample from a set of synthetically perturbed versions of it. In particular, we use resampling of the time signals of existing machine IDs, before computing the log-Mel spectra.

For the above tasks, the neural architecture is appended with a softmax layer, and cross-entropy loss is used. The softmax classification score of a test sample, measured at the output corresponding to its true machine ID, is taken as a measure of a sample's "inlier" score. Its negative is taken as the anomaly score.

2.2.1. Classifier Architectures

For the classification task, we employ two different architectures; MobileNetV2 and ResNet-50. MobileNetV2 is introduced in [4] as a computationally efficient improvisation of convolutional neural networks for visual recognition tasks such as object detection, classification and semantic segmentation. We use off-the-shelf Keras implementation of MobileNetV2, with the width multiplier parameter set to 0.5. A summary of the architecture is given in Table 2. The ResNet-50 [5] (Residual Network) model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. For ResNet-50, we also use an off-the-shelf Keras implementation.

2.2.2. Inputs

The inputs to the classifiers are 64×128 images, which are the log-Mel spectrograms computed using the following parameters:

1. Each input 10s file is split into frames of length 64ms, with hop length of 32ms between frames.
2. 1024-FFT and 128 Mel bins are used to featurize each frame.
3. 64 featurized frames are stacked to form a 64×128 image.
4. The successive 64×128 images have an overlap of 56 frames.

2.2.3. Label Augmentation

For the label augmentation, we have applied a combination of the following two main techniques:

- **Linear combination augmentations:** The provided machine IDs are combined in pairs using randomized linear combinations, and the network is trained to learn to identify the mixing proportions. For example, for an input sample that is a mixture of $(0.4 * x_1) + (0.6 * x_2)$, where x_1 and x_2 are samples from IDs 1 and 2, the network is trained to output $[0.4, 0.6, 0, 0, \ldots]$. KL divergence is used as the loss for this task. We consider linear combinations both before and after taking the log, on Mel-spectrograms.
- **Spectral warping augmentations:** We perturb samples from existing machine IDs to create new machine IDs, using image warping. Specifically, we apply a polynomial warping using opencv's geometrical transformation functions.

Table 1: DCASE 2020 Task 2 Experimental Results over Dev Data

Algorithm	Crit.	Toy Car	Toy Conveyor	Fan	Pump	Slider	Valve
Baseline	-	78.77 (67.58)	72.53 (60.43)	65.83 (52.45)	72.89 (59.99)	84.76 (66.53)	66.28 (50.98)
System 1	mean	94.97 (90.03)	**81.46 (66.62)**	82.39 (78.23)	87.64 (82.37)	97.09 (88.03)	90.52 (88.06)
	max	**95.57 (91.54)**	79.74 (64.88)	82.23 (77.87)	**87.87 (82.38)**	96.84 (87.72)	98.46 (94.87)
System 2	mean	95.04 (90.39)	80.67 (65.90)	82.33 (78.97)	86.94 (79.60)	97.28 (89.54)	97.38 (91.21)
	max	95.15 (90.86)	79.22(64.25)	82.73 (78.76)	87.05 (79.26)	97.16 **(90.34)**	98.53 (93.40)
System 3	mean	94.64 (89.48)	80.53 (65.58)	82.75 **(79.72)**	86.73 (79.60)	**97.62** (89.70)	95.00 (90.32)
	max	95.27 (90.23)	79.10 (64.02)	**83.06** (79.55)	87.04 (79.52)	97.43 (88.91)	**99.07 (96.20)**
System 4	-	80.51 (71.89)	76.03 (60.70)	70.10 (53.62)	75.68 (68.97)	93.29 (83.46)	89.68 (70.95)

Operation	t	c	n	s
Conv2D	-	16	1	2
Bottleneck	1	8	1	1
Bottleneck	6	16	2	2
Bottleneck	6	16	3	2
Bottleneck	6	32	4	2
Bottleneck	6	48	3	1
Bottleneck	6	80	3	2
Bottleneck	6	160	1	1
Conv2D	-	1280	1	1
Avg Pool	-	1280	1	-
Dense	-	num classes	1	-

Table 2: MobileNetV2 architecture used in this report. Each line describes a sequence of 1 or more identical (modulo stride) layers, repeated n times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s and all others use stride 1. All spatial convolutions use 3 × 3 kernels. The expansion factor t is always applied to the input size as described in [4]

Depending on the specific machine type, different combinations of the aforementioned augmentations are found to give the best results on the development test set. We implement ensembling over such different combinations in our submission.

2.3. Ensembling

We submit 4 systems to the challenge, which are essentially ensembles of different variants of the above described two approaches.

To ensemble across multiple anomaly detection models, we transform the anomaly scores of each model into a standardized scale, before combining them. The standardization transformation for any given model is applied in a per-machine ID fashion, by computing the mean and variance of its anomaly scores over the training data for that machine ID. The anomaly scores are then transformed to have zero mean and unit variance over the training data of that machine ID. Standardized anomaly scores across different models are then combined using mean or max ensembling.

3. DATASET

The data used for this task comprises parts of ToyADMOS [12] and the MIMII [13] Dataset 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 that includes both a target machine's operating sound and environmental noise. The following six types of toy/real machines are used in this task:

- Toy-car (ToyADMOS)
- Toy-conveyor (ToyADMOS)
- Valve (MIMII Dataset)
- Pump (MIMII Dataset)
- Fan (MIMII Dataset)
- Slide rail (MIMII Dataset)

4. RESULTS

As instructed by the challenge organizers, in this section we only report results using the development set. In Table 1, we present AUC results and pAUC in parentheses for both the challenge baseline autoencoder model, and our 4 submissions for all 6 machines averaged across IDs.

The systems 1-4 are implemented as follows:

- **System 1 (M-G-R-IDcl):** An ensemble of a MobileNetV2, a ResNet-50 both trained in a self supervised manner and a Group MADE network.
- **System 2 (M-G-IDcl):** An ensemble of a MobileNetV2 trained in a self supervised manner and a Group MADE network.
- **System 3 (M-A-G-IDcl):** An ensemble of a MobileNetV2 trained in a self supervised manner, and another self-supervised MobileNetV2 with Additive angular margin (ArcFace) [14] as a loss function, and a Group MADE network.
- **System 4 (Group MADE):** A Group MADE network as described in Sect. 2.1.

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

[2] 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.

[3] 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.

[4] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "MADE: masked autoencoder for distribution estimation," 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. 881–889.

[7] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 271–275.

[8] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 15 663–15 674.

[9] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 9758–9769.

[10] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations*, 2018.

[11] B. Liron and H. Yedid, "Classification-based anomaly detection for general data," *arXiv preprint arXiv:2005.02359*, 2020.

[12] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "Toyadmos: A dataset of miniature-machine operating sounds for anomalous sound detection," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2019, New Paltz, NY, USA, October 20-23, 2019*. IEEE, 2019, pp. 313–317.

[13] H. Purohit, R. Tanabe, K. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," *CoRR*, vol. abs/1909.09347, 2019.

[14] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.