Missing data recovery using autoencoder for multi-channel acoustic scene classification

Yuki Shiroma
Tokyo Metropolitan University
Tokyo, Japan

Yuma Kinoshita
Tokyo Metropolitan University
Tokyo, Japan

Keisuke Imoto
Doshisha University
Kyoto, Japan

Sayaka Shiota
Tokyo Metropolitan University
Tokyo, Japan

Nobutaka Ono
Tokyo Metropolitan University
Tokyo, Japan

Hitoshi Kiya
Tokyo Metropolitan University
Tokyo, Japan

***Abstract*—In this paper, we propose a method of missing data recovery using an autoencoder for multi-channel signals. Recently, many deep neural network-based classification methods using multi-channel signals have been proposed. The advantage of using multi-channel signals is that both frequency and spatial information can be used. However, systems using such signals are vulnerable to missing data because of a mismatch between the training and testing data. To minimize the mismatch, some techniques that include simulated missing data to the training data have been proposed. However, it is difficult to prepare missing data covering all possible mismatch situations. Therefore, we focus on using an autoencoder to recover the missing data without any assumptions of missing situations. In the case of multi-channel data inputted into an autoencoder, channel relationships are compressed into a low-dimensional hidden layer. Then, the autoencoder outputs data to reconstruct the input data from the layer. Therefore, when multi-channel input into the autoencoder has some missing channels, the output of the autoencoder is expected to recover some missing channel information by using the hidden layer. Since the autoencoder is regarded as a preprocessing of acoustic tasks, we evaluated the proposed method using an acoustic classification task. From the experimental results, we confirmed that the proposed method can recover missing data and improve the classification performance.**

***Index Terms*—Missing data recovery, Autoencoder, Multi-channel signals**

I. Introduction

Recently, as an evolution of machine learning, many neural network-based methods have been proposed for audio signal processing. These methods can handle complicated tasks better than conventional algorithm-based methods because neural networks can effectively extract information automatically. Therefore, these methods can achieve high performance in various acoustic classification tasks such as scene classification, and speech recognition [1], [2].

In particular, many neural network-based methods have been proposed for acoustic classification tasks using multi-channel signals [3]–[6] because both frequency and spatial information can be extracted from multi-channel signals [7]. However, it has also been reported that in the case of multi-channel input data including missing data, the performance of classification systems deteriorates significantly [8] because of a mismatch between the training and testing data. To minimize the the mismatch, some techniques such as data augmentation by training with simulated missing data have been proposed [9], [10]. However, it is difficult to prepare missing data covering all possible mismatch situations for constructing a robust model.

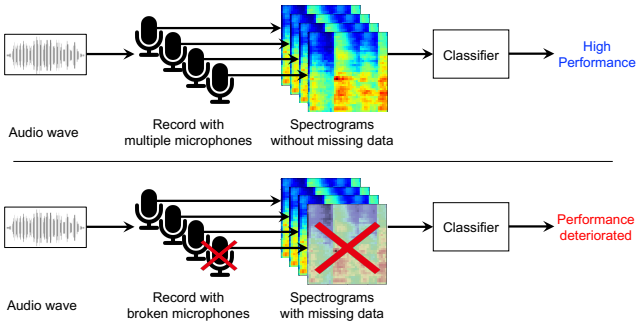

Fig. 1. Performance differences between acoustic tasks with and without missing channels

Therefore, we propose a method to recover missing data in multi-channel signals using an autoencoder. Generally, an autoencoder is used to extract important information in a low-dimensional hidden layer. In the case of multi-channel data inputted into an autoencoder, channel relationships are compressed into the hidden layer. Then, the autoencoder outputs data to reconstruct the input data from the hidden layer. Therefore, when multi-channel input into the autoencoder has some missing channels, the output of the autoencoder is expected to recover some missing channel information by using the trained channel relationships in the hidden layer. Additionally, since the missing data are different from the training data, data recovery using the proposed method only once is not sufficient. Therefore, by repeating this process, missing data is gradually recovered. This process is similar to the Griffin-Lim algorithm [11]. Since the recovered data contribute to improving the classification performance, in this paper, the proposed method is regarded as a preprocess of classification tasks. To evaluate the effectiveness of our proposed method,

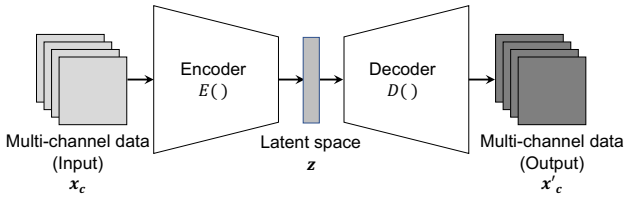

Fig. 2. Network architecture of the autoencoder used in the proposed method

we performed the acoustic scene classification (ASC) task. We compared classification performances between the task with missing data and the task with recovered data. Moreover, we investigated the effects of repeating autoencoder inputting. From the experimental results, we found that the classification performance of the task with recovered data is improved by 7.85 points in macro F-score compared with that with missing data.

II. Acoustic Task with Multi-Channel Signals

As an evolution of machine learning, the spatial and frequency information extracted from multi-channel signals is utilized for various acoustic tasks. A common example is ASC, which is a task that classifies sounds into predefined categories such as "cooking" "vacuuming" and "watching TV" or situations such as "being on the bus" "being in a park" and "meeting." Since 2015, an international competition called the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge has been held every year. In this competition, many deep neural network-based methods have been proposed and they showed high performances.

In the case of multi-channel data, classification methods obtain higher performances by using both frequency and spatial information. However, as shown in Fig. 1, it has also been reported that in the case of multi-channel input data including missing data, the performance of the method declines significantly [8] because of a mismatch between the training and testing data. The more microphones are used, the greater the possibility of microphone malfunction or network problems, and so on. Actually, in the DCASE challenge, one of the microphones was not used due to a recording error. Since it is difficult to avoid failures completely, a method to recover missing data is required to deal with this problem.

III. Proposed Method

In this section, the proposed method of missing signal recovery using an autoencoder is described.

A. Autoencoder with multi-channel signals

An autoencoder consists of two components, an encoder and a decoder. As shown in Equation (1), the autoencoder is trained to reconstruct input data by minimizing the mean squared error (MSE) of input and output as follows,

$$\min\Big(\frac{1}{n}\sum_{i=1}^{n}\big(\boldsymbol{x}-D\big(E\big(\boldsymbol{x}\big)\big)\big)^2\Big), \tag{1}$$

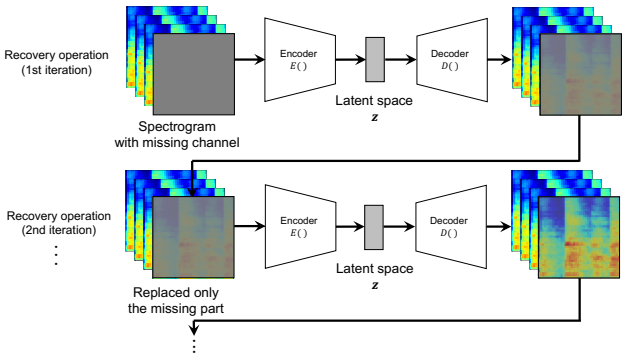

Fig. 3. Block diagram of proposed missing signal recovery method using autoencoder. Only a missing channel was replaced. Normal channels were not changed.

where n is the number of samples, and $\boldsymbol{x}$ and $\boldsymbol{x'}$ represent input and output spectrograms, respectively. $E()$ and $D()$ are the encoder and decoder functions of an autoencoder, respectively.

An autoencoder is trained to reconstruct input data. Since important information is extracted and compressed in the layer, an autoencoder can recover input data from a low-dimensional hidden layer. In our proposed method, as shown in Fig. 2, a multi-channel autoencoder was used. In a multi-channel autoencoder, channel relationships are compressed to latent variables. When multi-channel input into the autoencoder has some missing channels, the output of the autoencoder is expected to recover some missing channel information by using the layer.

B. Missing signal recovery using autoencoder

In this section, we describe the proposed method of missing data recovery. As a similar task, there is a virtual microphone technique but different from the virtual microphone, any channels need recovering in the missing data recovery. We focus on the fact that feature extraction of multi-channel signals using an autoencoder can recover missing data, as described in Section 3.A. The procedure for recovering missing data is illustrated in the top of Fig. 3. Since only the 1st iteration is not enough to recover missing data because of the large difference between the training and testing data, by repeating this procedure, missing data are gradually recovered. This algorithm is similar to the Griffin-Lim algorithm.

The specific steps are as follows. In the training step, an autoencoder is trained with multi-channel signals without missing data. In the testing step, to recover missing data, multi-channel signals with missing data are inputted into the autoencoder. Then, only the missing channel data of the autoencoder output is replaced with the input data. As shown in Fig. 3 the replaced data are regarded as new input data including missing data, and the replaced data is repeatedly input to the autoencoder. With this procedure, the missing data are gradually recovered and can be treated as the original data without any missing data.

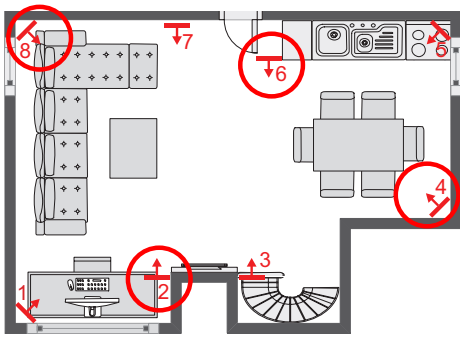

Fig. 4. 2D floorplan of combined kitchen and living room [6] with selected microphone numbers that were circled

TABLE I
TYPE OF CLASS AND NUMBER OF 10 SECONDS SEGMENTS OF ARRANGED DATASET USED IN THE EXPERIMENT

Class	Training	Validation	Testing	Sum
Absence	3018	754	943	4715
Cooking	819	205	257	1281
Dishwashing	227	57	72	356
Eating	369	92	116	577
Other	330	82	103	515
Social activity	790	198	248	1236
Vacuum cleaner	155	39	49	243
Watching TV	2982	746	933	4661
Working	2982	746	933	4661
Total	11672	2919	3654	18245

IV. EXPERIMENT

A. Dataset

In our experiments, the SINS dataset [12] was used as a multi-channel acoustic scene dataset. The SINS database contains continuous audio recordings of one person living in a vacation home over a period of one week. To simplify, as shown in Fig. 4, 16kHz four-channel audio signals were generated using microphones Nos. 2, 4, 6, and 8. The nine classes "Absence," "Cooking," "Dishwashing," "Eating," "Other," "Social activity," "Vacuum cleaner," "Watching TV," and "Working" were used. The class "Social activity" was made by concatenating "Calling" with "Visit." Because SINS is a long dataset, the dataset was divided into 10 seconds segments. Finally, the dataset was divided into train, validation, and testing datasets as shown in Table I.

B. Experimental setup

As an input feature, a four-channel logmel amplitude spectrogram was used. It was extracted by calculating the short-time fourier transform where the window length was 1024, the window shift was 320, and the mel filter bank dimension was 40. As an autoencoder, a 19-layer convolutional autoencoder was used. Table II shows the network structure of the autoencoder. It was based on a model that was proposed for the anomaly sound detection task [13]. The training conditions for the autoencoder were 1000 epochs using the Adam optimizer [14], where the parameters of the optimizer were set at a learning rate of 0.0001, $\beta 1$ = 0.9, and $\beta 2$ = 0.999. It was trained with training and validation datasets without missing data. In this experiment, to simplify the conditions, we assume that the information of a missing channel was given data, and we also regard the missing channel as fixed. We assume that a missing channel can be estimated by cross-correlation easily. An initial value of missing data is the average value of other channels. The autoencoder trains differences between channels, which are represented by small changes in values. Other initial values such as zeros make a huge difference so these values failed to recover signals. The number of iterations of the proposed method was set to 30.

TABLE II
NETWORK STRUCTURE OF AUTOENCODER USED IN PROPOSED METHOD

Layer	Output size
Input	40 × 496 × 4
Convolution (3 × 3, 16) + ReLU	40 × 496 × 16
Convolution (3 × 3, 16) + ReLU	40 × 496 × 16
MaxPooling (2 x 2)	20 × 248 × 16
Convolution (3 × 3, 32) + ReLU	20 × 248 × 32
MaxPooling (2 x 2)	10 × 124 × 32
Convolution (3 × 3, 64) + ReLU	10 × 124 × 64
MaxPooling (2 x 2)	5 × 62 × 64
Convolution (3 × 3, 128) + ReLU	5 × 62 × 128
Convolution (3 × 3, 256) + ReLU	5 × 62 × 256
Convolution (3 × 3, 128) + ReLU	5 × 62 × 128
Convolution (3 × 3, 64) + ReLU	5 × 62 × 64
UpSampling (2 x 2)	10 × 124 × 64
Convolution (3 × 3, 32) + ReLU	10 × 124 × 32
UpSampling (2 x 2)	20 × 248 × 32
Convolution (3 × 3, 16) + ReLU	20 × 248 × 16
UpSampling (2 x 2)	40 × 496 × 16
Convolution (3 × 3, 4) + ReLU	40 × 496 × 4
output	40 × 496 × 4

To evaluate the proposed method, we performed the ASC task. As a classification model, Convmixer [15] was used. The training conditions for the autoencoder were 100 epochs using the Adam optimizer [14], where the parameters of the optimizer were set at a learning rate of 0.0001, $\beta 1$ = 0.9, and $\beta 2$ = 0.999. It was trained with training and validation datasets without missing data as shown in Table I. To evaluate the performance of the proposed method, we compared three conditions using testing data. The first condition is "Without missing data." It was only just a logmel amplitude spectrogram calculated using four-channel signals without missing data. This result means one of the upper bounds of the proposed method. The second condition is "With missing data." It was multi-channel data with missing data. The third condition is "Proposed." It means the input data to the ASC system was recovered using the proposed method. The number of iterations was set to 30 in our proposed method. As a metric, a macro F-score [6] was used because macro F-score considers the weights of the data amount for each class.

C. Experimental results

Fig. 5 shows the macro F-score of the ASC task and MSE of the autoencoder input and output on each iteration. MSE was calculated from only one channel that had missing data. As shown in Fig. 5, the macro F-score increased until the

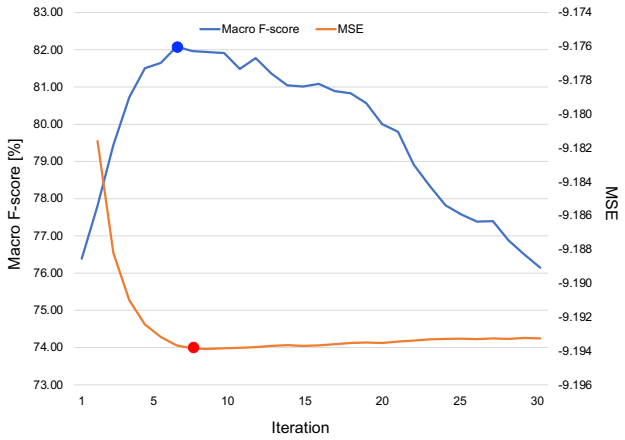

Fig. 5. Macro F-score and MSE of autoencoder input and output on each iteration

TABLE III
EXPERIMENTAL RESULTS OF RECOVERING MISSING DATA IN ACOUSTIC SCENE CLASSIFICATION

Conditions		Macro F-score [%]
Without missing data		86.63
With missing data		74.24
Proposed	Min. MSE (8th iteration)	81.96
	Best score (7th iteration)	**82.09**

7th iteration. This result indicates that our proposed method can gradually recover the missing channels and thus improve the classification performance. However, the macro F-score decreased after the 7th iteration. This indicates that there was an appropriate number of iterations. The reason for this score degradation is that the proposed method used the autoencoder without any assumptions for the classification system. It led to the over-recovery of missing channel data. Regarding the assessments on when to stop the iteration of the proposed method, the trajectory of the MSE was also plotted. From the MSEs, on the 8th iteration, the MSE indicated the lowest score. Since the macro F-score was almost similar to that with the lowest macro F-score (7th iteration), it can be regarded as one of the assessment parameters for when to stop the iteration in the proposed method. From these results, the proposed method could recover the missing data, and the iterations were stopped on the basis of MSE.

Table III shows the macro F-score for ASC under each condition. First, by comparing "Without missing data" and "With missing data," we found that the score of "With missing" was lower than that of "Without missing data." This indicates that channel missing data degraded the ASC performance. Second, by comparing "With missing data" with "Proposed Best score," we found that the score of "Proposed best score" was higher than that of "With missing data." This indicates that our proposed method can recover the missing channels with an appropriate number of iterations, leading to the improvement of the classification task.

Fig. 6 shows the logmel spectrograms of each condition.

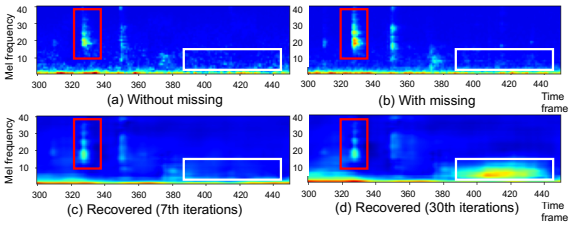

Fig. 6. Logmel spectrogram of each condition

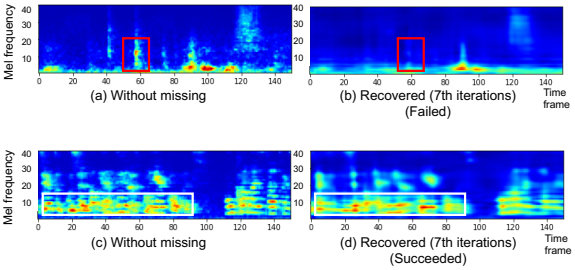

Fig. 7. Logmel spectrogram of each condition in the failed case of the ASC task

First, by comparing "Proposed (30th iteration)" with others, we found that "Proposed (30th iteration)" has a markedly different value at the white box. This indicates that when the number of iterations exceeds the appropriate number, over-convertion occurred, which badly affected the classification performance. Second, by comparing "Proposed (7th iteration)" with others, we found that "Proposed (7th iteration)" was more similar to "Without missing data." Moreover, as shown in Table III "Proposed (7th iteration)" showed a higher score than "With missing data." These results indicates that our proposed method can recover the missing channels and thus improve the ASC performance. Fig. 7 shows the two conditions of "Proposed (7th iteration)." The first condition is "Failed to classify," the other is "Classified successfully." By focusing on a red box, we found that an acoustic feature vanished from "Failed to classify." This indicates that the classifier rarely failed to classify because sometimes acoustic features vanished owing to the failure of recovery.

Figs. 8 and 9 show the confusion matrices of "With missing data" and "Proposed Best score." Compared with the proposed method with the missing situation, the accuracies of six classes were improved (Cooking, Dishwashing, Eating, Social activity, Watching TV, and Working). In particular, the Diswashing and Working classes obtained improvements markedly. It can be seen that these classes are required for all channels for the ASC task.

V. CONCLUSION

In this research, we proposed a method of missing data recovery using an autoencoder for multi-channel signals as a preprocess of acoustic scene classification. In the case of multi-channel input data, an autoencoder pretrained with

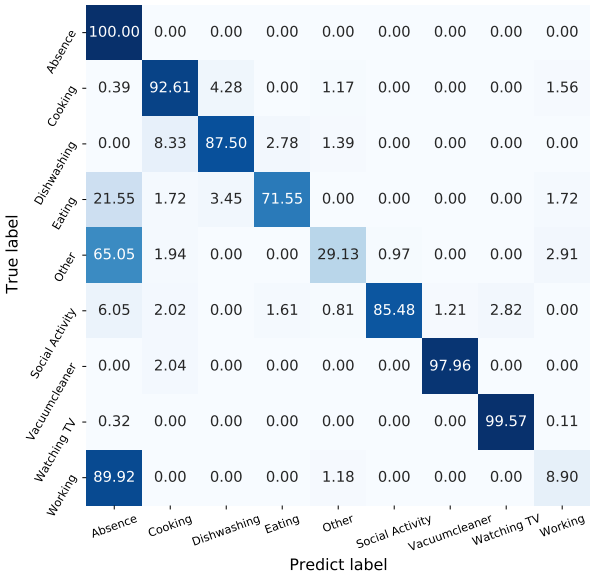

Fig. 8. Confusion matrices under the condition "With missing data" [%]

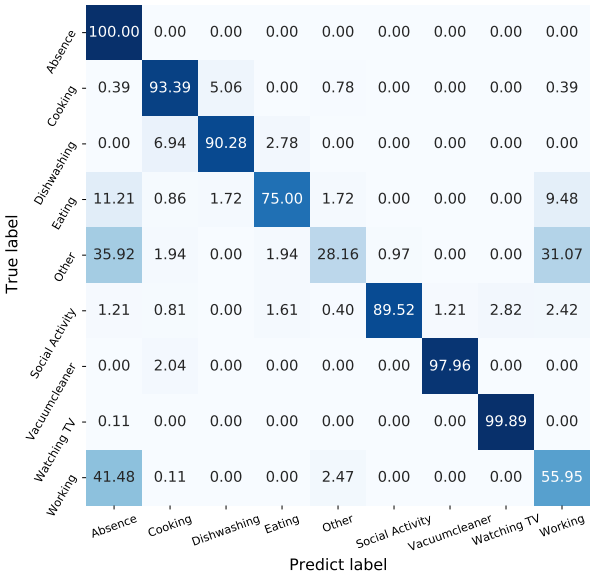

Fig. 9. Confusion matrices under the condition "Proposed 7th iteration" [%]

complete multi-channel data can recover missing data. By repeating the proposed method, missing data are gradually recovered. To evaluate the effectiveness of our proposed method, we performed the ASC task. From the experimental results, we confirmed that the proposed method can recover missing data and improve the classification performance. As our future work, we will perform other tasks using the proposed method and also evaluate more complex conditions.

Acknowledgement

This work was supported in part by JSPS KAKENHI Grant numbers JP19K20271, JP20H00613, and ROIS DS-JOINT (030RP2021) to S. Shiota.

References

[1] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: analysis of dcase 2021 challenge systems," *arXiv:2105.13734*, 2021.

[2] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association*, 2018.

[3] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, 2017.

[4] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5745–5749, 2016.

[5] T. Ochiai, S. Watanabe, T. Hori, and J. R. Hershey, "Multichannel end-to-end speech recognition," in *Proceedings of the 34th International Conference on Machine Learning (ICML) - vol. 70*, p. 2632–2641, 2017.

[6] G. Dekkers, L. Vuegen, T. van Waterschoot, B. Vanrumste, and P. Karsmakers, "DCASE 2018 Challenge - Task 5: Monitoring of domestic activities based on multi-channel acoustics," *arXiv:1807.11246*, 2018.

[7] K. Imoto and N. Ono, "Spatial cepstrum as a spatial feature using a distributed microphone array for acoustic scene analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1335–1343, 2017.

[8] Y. Shiroma, K. Imoto, S. Shiota, N. Ono, and H. Kiya, "Investigation on spatial and frequency-based features for asynchronous acoustic scene analysis," *APSIPA Annual Summit and Conference*, 2021.

[9] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of Interspeech 2019*, pp. 2613–2617, 2019.

[10] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220–5224, 2017.

[11] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[12] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," pp. 32–36, November 2017.

[13] T. B. Duman, B. Bayram, and G. İnce, "Acoustic anomaly detection using convolutional autoencoders in industrial processes," in *14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO)*, pp. 432–442, 2020.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[15] A. Trockman and J. Z. Kolter, "Patches are all you need?," *arXiv preprint*, 2022.