# Anomaly Detection in Driving Urban Soundscape

Giovanni Muhammad Raditya
B4 Takeda Laboratory

## I. Introduction

As a human drivers, we are able to perceive both sound and vision in urban environment using our ears and eyes. In cases where vision are obstructed our main source of info are from our hearing, even then in some cases drivers do not have a clear perception of signals coming from outside. This is due to high soundproofing in passenger compartments for modern cars or other distraction. Hence to improve those missing information, sound recorded by microphones would be able to give a reliable source of information, as there are vast numbers of cases where sound information is important. Such as, emergency vehicle siren, horn, accident noise, a vehicle approaching from a sharp corner [1], and obstructed visibility [2].

In this research we are going to focus more on detecting emergency vehicle with a minimal anomaly driving sound dataset. For this experiment, we captured recordings onboard a vehicle equipped with 8 microphones, 6 on the outside and 2 on the inside of the passenger compartment. The recording has revealed the difficulty of capturing and labeling audio data containing anomaly sounds and the influence of sensor location to the signal acquisition.

To overcame the scarce availability and difficulty of collecting a large dataset of anomaly sound events during driving, we use the more available normal driving sound data and feed the autoencoder framework. Two autoencoder framework will be used based on DCASE 2021 baseline dense autoencoder [3] and U-net type network. Mean Square Error (MSE) with an additional max pooling on time axis and average pooling on frequency axis will be used as an evaluation and anomaly score of the network.

## II. Related Research

Anomalous sound detection (ASD) have been used as automatic detection of mechanical failure as in DCASE 2020 [4], DCASE 2021 [3], and DCASE 2022 [5]. Where in real-world scenario, actual anomalous sounds rarely occur, this means that we must detect unknown anomalous sounds that were not in the given training data. Thus it can be correlated with detecting siren in driving scenario, where siren does rarely occurs and big siren dataset in driving scenario is unavailable, anomaly sound detection can be one of the choice to detect siren as an anomaly in urban driving scenario.

We further focus on the other DN models for anomaly detection: Dense Linear AE and U-Net [6]. Dense Linear AE have been used as a baseline in multiple DCASE challenge for anomaly detection. U-Net is also useful in the image processing field, and acoustic anomaly detection is sometimes held using sound spectrogram images. Some research have been conducted using a mask as the reconstruction focus [7] and it shows an improvement compared to linear Dense AE. Thus by incorporating U-Net skip connection to focus on reconstructing the mask rather than the whole spectrogram, we believe these models are also helpful in this task. We build these models only using normal data, and compute an error between given and reconstructed data as an anomaly score. Since the models cannot well reconstruct anomaly data which are not used for model training, higher error scores are observed for the anomaly data.

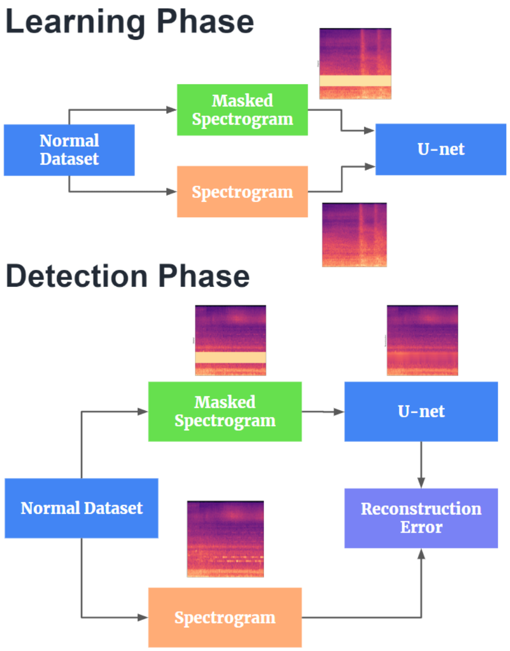

Fig. 1. Overview of the masked method

## III. Experimentation Details

### A. Data Preparation

To evaluate the performance of our model, we collected 2 hours of data by driving around Nagoya University and Japanese Red Cross Nagoya Daini Hospital at different traffic conditions. The data was gathered using 8x Sony ECM-FT5B omnidirectional microphones mounted on multiple places as illustrated in Figure 1 and a Roland OCTA-CAPTURE 8-channel audio interface. The data was recorded at a sampling frequency $f_s$ of 44100 Hz and saved in wav format.

Recording containing siren events is collected with total of 60 second of recording. In total we generate more than 5.4K samples of 10s frames with each microphone generate 682 samples. To overcame the lack of anomaly data, we augmented

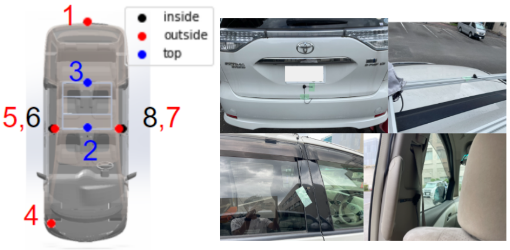

Fig. 2. (Left) Layout of the microphone positions: positions 1 near the car plate number, positions 2-3 at the top of the car, positions 5,7 located outside the car, and positions 6,8 located inside the car. (right) Details of the microphones, from top-left to bottom-right: (top-left), at position 1; (top-right), in the top positions 2; (bottom-left), outside the car position 5; (bottom-right), inside the car position 8.

6 siren sound with 23 normal samples using noise factor in the range of (0.5, 1). Thus in total, we have 825 samples for each microphone. A more detailed description of the dataset employed is given in Table I.

TABLE I
DATASET USED FOR EVALUATION

| | Training | Test | |
|---|---|---|---|
| Type | Normal | Normal | Anomaly |
| Each Microphone | 609 | 67 | 149 |

## B. Preprocessing

*1) Dense Linear AutoEncoder:* Before the data is feed into the model, the audio is transformed into mel-spectrogram using short-Time Fourier Transform (STFT) and mel-filter bank with the parameter as follow: Hamming window; length of windowed signal is 1024; sampling rate is 22.05 kHz; hop length of 512; and 128 mel-bins. Thus we have (427,128) as the shape of mel-spectrogram. To add more feature during training, we concatenating frame of the mel-spectrogram 5 times. Thus, in total we have (427, 640) features for training.

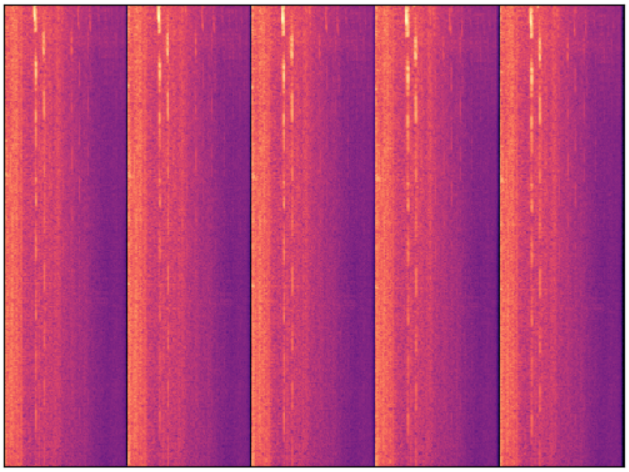

Fig. 3. Mel-spectrogram feature vector

*2) U-Net:* Before the data is feed into the model, the audio is transformed into mel-spectrogram using short-Time Fourier Transform (STFT) and mel-filter bank with the parameter as follow: Hamming window; length of windowed signal is 1024; sampling rate is 22.05 kHz; hop length of 413; and 128 mel-bins. Thus we have (512,128) as the shape of mel-spectrogram. Next, the spectrogram is masked horizontally by 2 frequency mask with value 0. The masking position is changed to a random position with 0.15 mask percentage. Figure 4 shows the original spectrogram (A) and masked spectrograms (B, C).

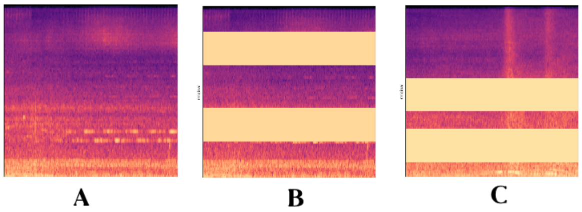

Fig. 4. A: Original spectrogram, B, C: Spectrogram masked by 3/20 width in horizontal direction

## C. Model training

*1) Dense Linear AutoEncoder:* In our experiments, we utilized a laptop with 16 GB RAM, an Intel Core i5-10300H CPU (8 cores @2.50 GHz), and NVIDIA GeForce GTX 1650 GPU 4 GB. The laptop was running windows 10, and we used the TensorFlow deep learning framework to implement the network designs. The baseline setup for the network hyper-parameters are listed as follows: 150 training epochs; the initial learning rate is 0.001; 512 batch size; and we trained the models with Adam optimizer [8]. We evaluated the models using reconstruction error using the mean square error calculation. In addition, we also add pooling during the mean square error calculation, with max-pooling(2,2) for the time axis and average-pooling(2,2) on frequency axis as it is believed to be able to leverage weakly labeled data by using different pooling to aggregate dynamic predictions. [9]. To obtain the offset-onset anomaly time, we also implement addition MSE scoring by having a 2 s overlap and 0.5 s hop length on the MSE.

*2) U-Net:* For our second experiments, we utilized a machine with 30 GB RAM, and Quadro M4000 GPU 8 GB. We used the TensorFlow deep learning framework to implement the network designs. The baseline setup for the network hyper-parameters are listed as follows: 150 training epochs; the initial learning rate is 1; 32 batch size; and we trained the models with Adam optimizer [8]. Different with the previous experiment we evaluated the models using reconstruction error of the masked spectrogram reconstruction with additional pooling during the mean square error calculation, with max-pooling(2,2) for the time axis as it is believed to be able to leverage weakly labeled data by using different pooling to aggregate dynamic predictions. [9].

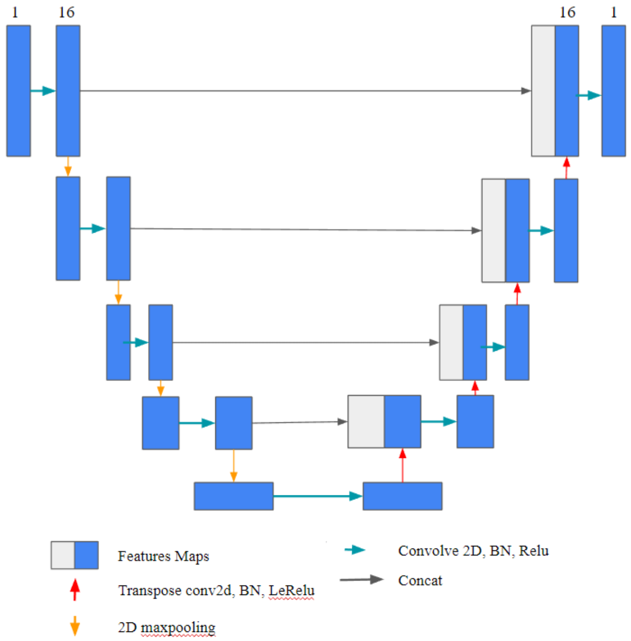

Fig. 5. U-Net Framework

## IV. Experimental Evaluation

### A. Comparison of AUC for Different Pooling

The method to add max-pooling and avg-pooling to time axis and frequency axis respectively, have different result in AUC for Linear Dense AutoEncoder as shown in Table II. For some cases such as microphone 1, microphone 6, and microphone 8, max-pooling on time axis give better result in AUC and pAUC score compared to both no-pooling and avg-pooling on frequency axis. Whereas in the case of microphone 2 and microphone 3 avg-pooling on frequency axis give better result, with the rest (microphone 4, microphone 5, and microphone 7) have naive MSE as their preferred method. For overall location, microphone inside the driving compartment ( microphone 1, microphone 6, and microphone 8) have relatively higher AUC score especially when max-pooled on time axis. With the worst AUC score held by microphone 4 which is located in front of the car.

### B. Comparison of AUC for Different model

When comparing the accuracy between U-Net model and Linear Dense AutoEncoder model, from Table II it is shown that the Linear Dense AutoEncoder still outperform U-Net in 5 out of 8 microphone. Nevertheless it still have the same overall average AUC on both U-Net and Linear Dense AutoEncoder with 0.87 where as Linear Dense AutoEncoder still outperform U-Net in term of pAUC. There is not any pattern regarding correlation between mic location to the model used that can be concluded at the moment, thus further testing are still needed for the comparison.

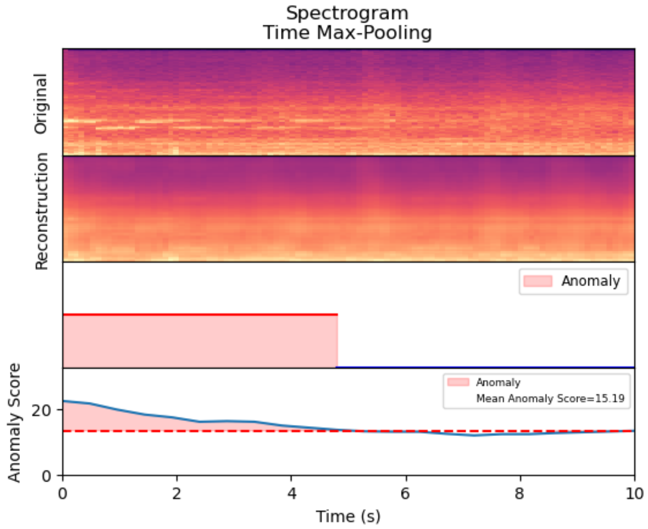

Fig. 6. Anomaly Offset-onset timing

## V. Considerations

The model is trained using limited number of normal driving sound. Various anomalous sound that are not recorded such as horn and accident noise can be detected as an anomaly in this model. Thus, a further classification of sound after anomaly is detected are certainly needed. Implementation of masking on each epoch are needed during U-Net model training to produce a more reliable model. Recording more normal driving dataset can also be considered to increase the model training in reconstructing the sound. Further preprocessing using gammatone filter can also be considered, as it shown on past research [10] that it improve the accuracy of the model.

## VI. Summary and Future Issues

The microphone with less noise (microphone 1, microphone 6, and microphone 8) have higher AUC rating when using max-pooling on time-axis, where as microphone with the most noise (microphone 2 and microphone 3) have higher AUC rating when using avg-pooling on frequency-axis. Thus, pooling method can be used depends on the noise level of the recording. For overall AUC score, microphone 1 (located behind the car) have the highest AUC and pAUC score with score 0.98 and 0.94 respectively.

Comparing the Linear Dense AutoEncoder model with U-net model, the overall AUC of AutoEncoder are still considerably higher on some microphone compared to U-Net. Though this can change with the implementation of masking on each epoch and more training dataset are acquired [6] [7]. Noise removal preprocessing should also be considered to further improve the model as high noise can decrease the AUC as shown by the AUC of microphone 4. In addition to that, as we record the sound using symmetrical microphone 5,7 and 6,8 we should be able to calculate the direction of arrival using 2 microphone on each side[11].

TABLE II
ANOMALY DETECTION PERFORMANCE

| | | Linear Dense AutoEncoder | | U-Net | |
|---|---|---|---|---|---|
| | Pooling | AUC | pAUC | AUC | pAUC |
| Microphone 1 | Without Pooling | 0.96 | 0.84 | 0.98 | 0.90 |
| | Freq avg-pooling | 0.96 | 0.85 | - | - |
| | Time max-pooling | 0.97 | 0.86 | **0.98** | 0.94 |
| Microphone 2 | Without Pooling | 0.86 | 0.83 | 0.87 | 0.74 |
| | Freq avg-pooling | **0.97** | 0.89 | - | - |
| | Time max-pooling | 0.83 | 0.80 | 0.76 | 0.67 |
| Microphone 3 | Without Pooling | 0.89 | 0.79 | 0.86 | 0.77 |
| | Freq avg-pooling | **0.97** | 0.89 | - | - |
| | Time max-pooling | 0.83 | 0.75 | 0.83 | 0.74 |
| Microphone 4 | Without Pooling | **0.76** | 0.72 | 0.72 | 0.70 |
| | Freq avg-pooling | 0.75 | 0.71 | - | - |
| | Time max-pooling | 0.75 | 0.71 | 0.69 | 0.67 |
| Microphone 5 | Without Pooling | 0.85 | 0.80 | 0.91 | 0.82 |
| | Freq avg-pooling | 0.84 | 0.80 | - | - |
| | Time max-pooling | 0.84 | 0.80 | **0.91** | 0.85 |
| Microphone 6 | Without Pooling | 0.82 | 0.77 | 0.85 | 0.82 |
| | Freq avg-pooling | 0.81 | 0.77 | - | - |
| | Time max-pooling | **0.96** | 0.84 | 0.77 | 0.77 |
| Microphone 7 | Without Pooling | 0.89 | 0.78 | **0.90** | 0.88 |
| | Freq avg-pooling | 0.88 | 0.77 | - | - |
| | Time max-pooling | 0.85 | 0.76 | 0.88 | 0.86 |
| Microphone 8 | Without Pooling | 0.94 | 0.83 | 0.88 | 0.83 |
| | Freq avg-pooling | 0.93 | 0.82 | - | - |
| | Time max-pooling | **0.95** | 0.87 | 0.88 | 0.84 |
| Mean | Without Pooling | **0.87** | 0.80 | 0.87 | 0.71 |
| | Freq avg-pooling | 0.87 | 0.79 | - | - |
| | Time max-pooling | **0.87** | 0.80 | 0.72 | 0.79 |

## REFERENCES

[1] Y. Schulz *et al.*, "Hearing what you cannot see: Acoustic detection around corners," *CoRR*, vol. abs/2007.15739, 2020. [Online]. Available: https://arxiv.org/abs/2007.15739

[2] F. R. Valverde, J. V. Hurtado, and A. Valada, "There is more than meets the eye: Self-supervised multi-object detection and tracking with sound by distilling multimodal knowledge," *CoRR*, vol. abs/2103.01353, 2021. [Online]. Available: https://arxiv.org/abs/2103.01353

[3] Y. Kawaguchi *et al.*, "Description and discussion on DCASE 2021 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring under domain shifted conditions," *In arXiv e-prints: 2106.04492, 1–5*, 2021.

[4] Y. Koizumi, Y. Kawaguchi, and K. Imoto, "Description and discussion on dcase2020 challenge task2: unsupervised anomalous sound detection for machine condition monitoring," DCASE2020 Challenge, Tech. Rep., July 2020.

[5] K. Dohi *et al.*, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," *In arXiv e-prints: 2206.05876*, 2022.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[7] A. Matsui *et al.*, "Anomaly detection in mechanical vibration using combination of signal processing and autoencoder," *Journal of Signal Processing*, vol. 24, no. 4, pp. 203–206, 2020.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[9] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *CoRR*, vol. abs/1804.10070, 2018. [Online]. Available: http://arxiv.org/abs/1804.10070

[10] L. Marchegiani and I. Posner, "Leveraging the urban soundscape: Auditory perception for smart vehicles," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6547–6554.

[11] L. Marchegiani and P. Newman, "Listening for sirens: Locating and classifying acoustic alarms in city scenes," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2022.