NEURAL NETWORK TRAINING WITH MATCHING LOSS FOR RANKING FUNCTION

Technical Report

Tomohiro Takahashi[1], *Natsuki Ueno*[1,2], *Yuma Kinoshita*[3], *Yukoh Wakabayashi*[4], *Nobutaka Ono*[1],
Makiho Sukekawa[5], *Seishi Fukuma*[5], *Hiroshi Nakagawa*[5]

[1] Tokyo Metropolitan University, Tokyo, Japan
[2] Kumamoto University, Kumamoto, Japan
[3] Tokai University, Tokyo, Japan
[4] Toyohashi University of Technology, Aichi, Japan
[5] NEXCO-EAST ENGINEERING Company Limited, Tokyo, Japan

ABSTRACT

In this report, we summarize our approach for DCASE 2024 Challenge Task 10, acoustic-based traffic monitoring. Our approach consists of two improvements from the baseline system. One is the introduction of the matching loss for the ranking function to the loss function of the Convolutional Recurrent Neural Network (CRNN), which aims to improve the Kendall's Tau Rank Correlation (KTRC). The results indicate that it is also effective in improving the Root Mean Square Error (RMSE). The other improvement is a change in the input features. We also report the estimation performance for the development datasets.

Index Terms— traffic monitoring, vehicle detection, deep neural network, acoustic sensing, microphone array

1. INTRODUCTION

This report provides the description of our submitted system for the DCASE 2024 Challenge Task 10, which focuses on acoustic-based traffic monitoring. The goal of this task is to count the number of vehicles per vehicle type (car or Commercial Vehicle, CV) and per direction of travel (left or right) [1]. Additionally, this task will investigate the effectiveness of data augmentation by an open-source road acoustic simulator [2].

In this task, the performance is evaluated using two metrics: Root Mean Square Error (RMSE) and Kendall's Tau Rank Correlation (KTRC). Although RMSE is directly utilized as the loss function for training the neural network, it is not straightforward to determine the appropriate loss function for optimizing KTRC. Our proposed method aims to improve KTRC by incorporating a matching loss [3] for the ranking function into the loss function. We also compared several combinations of input features and loss functions for acoustic-based traffic monitoring. Additionally, we evaluated the estimation performance of our system with and without pre-training.

We conduct an experimental evaluation of our method using the training, validation, and synthetic data from the development dataset of DCASE 2024 Challenge Task 10. As a result, our approach of introducing matching loss showed some improvements not only in KTRC but also in RMSE. In addition, we confirmed that pre-training improves the estimation performance.

This report is organized as follows. In Section 2, we describe our acoustic-based traffic monitoring method. In Section 3, we show the experimental evaluations and the results. In Section 4, we summarize this report.

2. PROPOSED METHOD

We investigated input features and loss functions that are more effective than those used in the baseline system. Specifically, we compared several combinations of input features and loss functions and submitted the best-performing combination as our proposed method. This section describes the input features and loss functions used in the experiments.

2.1. Input feature

We considered the following input features in our experiments in Section 3. The idea of considering these input features came from the baseline system and our previous work [4, 5]. Here, $\mathbf{x}_i$ is the input acoustic signal of the ith channel ($i = 1, 2, 3, 4$), and the short-time Fourier transform spectrogram $\mathbf{X}_i$ of the $\mathbf{x}_i$ is calculated using the following equation:

$$\mathbf{X}_i = \mathrm{STFT}(\mathbf{x}_i). \tag{1}$$

Here, $\mathrm{STFT} : \mathbb{R}^N \to \mathbb{C}^{F \times T}$ in (1) represents the short-time Fourier transform that transforms an N sample time signal into a spectrogram with a frequency bin number $F\times$ time frame number T.

LogMelSpec: $\mathbf{X}_i^{\mathrm{LMS}} \in \mathbb{R}^{M \times T}$ is obtained by taking the logarithm of the short-time power spectrogram's magnitude after remapping $\mathbf{X}_i$'s frequency to the mel scale (logarithmic transformation of the frequency bands). Here, M specifies the number of frequency bands in the **LogMelSpec**.

LogPowSpec: $\mathbf{X}_i^{\mathrm{LPS}} \in \mathbb{R}^{F \times T}$, where the (f, t) element $\mathrm{X}_i^{\mathrm{LPS}}(f, t)$ is calculated using the following equation:

$$\mathrm{X}_i^{\mathrm{LPS}}(f, t) = 10 \log_{10}(|\mathrm{X}_i(f, t)|^2). \tag{2}$$

Here, f and t are indices in the frequency and time directions, respectively, and $X_i(f, t)$ denotes the (f, t) element of $\mathbf{X}_i$ (the same notation applies to other matrices).

GCC-PHAT: $\mathbf{X}_{i,j}^{\mathrm{GCC}} \in \mathbb{R}^{G \times T}$, where the (τ, t) element $\mathrm{X}_{i,j}^{\mathrm{GCC}}(\tau, t)$ is calculated using the following equation:

$$X_{i,j}^{\mathrm{GCC}}(\tau, t) = \mathcal{F}_{f \to \tau}^{-1} \frac{X_i(f,t) X_j^*(f,t)}{|X_i(f,t)||X_j(f,t)|}. \tag{3}$$

Here, $\mathcal{F}_{f \to \tau}^{-1}$ is the inverse Fourier transform from f to τ. The j is the number of a different channel from i $(j = 1, 2, 3, 4)$. G specifies the number of **GCC-PHAT** coefficients. The time difference of arrival (TDOA), i.e., the lag time between ith and jth channels, can be estimated by finding the maximum peak of $X_{i,j}^{\mathrm{GCC}}(\tau, t)$ [6].

PhaseDiff: $\mathbf{X}_{i,j}^{\mathrm{PD}} \in \mathbb{R}^{2 \times F \times T}$ is calculated using the following equations:

$$\Delta\phi_{i,j}(f,t) = \arg(\mathrm{X}_i(f,t)/\mathrm{X}_j(f,t)), \tag{4}$$

$$\mathrm{X}_{i,j}^{\mathrm{PDC}}(f,t) = \cos(\Delta\phi_{i,j}(f,t)), \tag{5}$$

$$\mathrm{X}_{i,j}^{\mathrm{PDS}}(f,t) = \sin(\Delta\phi_{i,j}(f,t)), \tag{6}$$

$$\mathbf{X}_{i,j}^{\mathrm{PD}} = \mathrm{Stack}(\mathbf{X}_{i,j}^{\mathrm{PDC}}, \mathbf{X}_{i,j}^{\mathrm{PDS}}). \tag{7}$$

As for the phase difference in (4), the effect of its periodicity on the input features is ignored by computing cos and sin as in (5) and (6). The Stack in (7) stacks the arrays $\mathbf{X}_{i,j}^{\mathrm{PDC}}$ and $\mathbf{X}_{i,j}^{\mathrm{PDS}}$ in the direction of the newly added channel dimension.

2.2. Loss function

We considered the following two loss functions, **MSE** and **Matching**, in our experiments in Section 3. The idea of considering **MSE** came from the baseline system, and that for **Matching** came from the motivation to improve KTRC. Here, for each data index $k = 1, \ldots, K$ with the batch size $K \in \mathbb{N}$, $y_k^{(*)}, \hat{y}_k^{(*)} \in \mathbb{R}$ denotes respectively the true and estimated vehicle counts corresponding to the label $(*) \in \{\text{car-l2r}^1, \text{car-r2l}^2, \text{CV-l2r}^3, \text{CV-r2l}^4\}$, and $\mathbf{y}^{(*)}, \hat{\mathbf{y}}^{(*)} \in \mathbb{R}^K$ denotes respectively the collection of true and estimated data for all index $k = 1, \ldots, K$.

MSE: L_{MSE} is calculated using the following equation:

$$L_{\mathrm{MSE}}(\hat{\mathbf{y}}^{(*)}; \mathbf{y}^{(*)}) = \frac{1}{K} \sum_{k=1}^{K} (y_k^{(*)} - \hat{y}_k^{(*)})^2. \tag{8}$$

Matching: L_{Matching} is calculated using the following equation:

$$L_{\mathrm{Matching}}(\hat{\mathbf{y}}^{(*)}; \mathbf{y}^{(*)}) = \frac{1}{K^2} \left(\frac{1}{2} \sum_{k=1}^{K} \sum_{l=1}^{K} |\hat{y}_k^{(*)} - \hat{y}_l^{(*)}| - \sum_{k=1}^{K} [\varphi(\mathbf{y}^{(*)})]_k \hat{y}_k^{(*)} \right). \tag{9}$$

Here, $\varphi : \mathbb{R}^K \to \mathbb{Z}^K$ is referred to as the ranking function, defined as

$$\varphi(\mathbf{y}^{(*)}) = \sum_{k=1}^{K} \begin{bmatrix} \mathrm{sign}(\hat{y}_1^{(*)} - \hat{y}_k^{(*)}) \\ \vdots \\ \mathrm{sign}(\hat{y}_K^{(*)} - \hat{y}_k^{(*)}) \end{bmatrix}. \tag{10}$$

Intuitively, $[\varphi(\mathbf{y}^{(*)})]_k$ denotes the number of elements smaller than $\hat{y}_k^{(*)}$ subtracted by the number of elements larger than $\hat{y}_k^{(*)}$. Therefore, φ maps the ranking of the input vector $\mathbf{y}^{(*)}$ into the integers within $\{-K+1, \ldots, K-1\}$. Note that L_{Matching} is convex with respect to the first variable $\hat{\mathbf{y}}^{(*)}$, and the subgradient of L_{Matching} is given by

$$\nabla L_{\mathrm{Matching}}(\hat{\mathbf{y}}^{(*)}; \mathbf{y}^{(*)}) = \frac{1}{K^2} \left(\varphi(\hat{\mathbf{y}}^{(*)}) - \varphi(\mathbf{y}^{(*)}) \right). \tag{11}$$

It means that the minimization of L_{Matching} induces the correspondence between $\varphi(\hat{\mathbf{y}}^{(*)})$ and $\varphi(\mathbf{y}^{(*)})$, that is, the ranking of the true and estimated data. A mathematical relationship such as that between the loss function L_{Matching} and the vector-valued function φ is generally referred to as the matching loss [3]. Motivated by this concept, we incorporated the matching loss for the ranking function as described above to improve the correspondence between the ranking of the true and estimated data.

3. EXPERIMENTAL EVALUATIONS

In this section, we report the results of our experimental evaluation for the development dataset of the DCASE challenge task 10 using the input features and loss functions described in Section 2.

3.1. Experimental conditions

Our system builds upon the baseline system provided by the organizers, incorporating several modifications. We trained our model using the training, validation, and synthetic data from the development dataset and evaluated it using the validation data. The evaluation was conducted across all locations, and we maintained the baseline system's method for generating synthetic data.

Two feature patterns, **LogMelSpec+GCC-PHAT** and **LogPowSpec+PhaseDiff**, combining amplitude-related and phase-related features, were used as input features. The sampling frequency F_s was 16 kHz, the STFT frame length N was 1024 points (64 ms), and the frameshift was 160 points (10 ms) for a 1-min signal. When the signal length $L = 60$ s, $F = \lfloor N/2 \rfloor + 1 = 513$ and $T = \lceil L \cdot F_s/(N/2) \rceil = 1875$. The number of frequency bands M of **LogMelSpec** was set to 48 and the number of coefficients G of **GCC-PHAT** was set to 96.

As for our neural network architecture, we employed a slightly modified version based on the baseline system CRNN. We used six Conv2D layers of convolutional encoders, each with filters 32-32-64-64-128-128 and a kernel size of (5, 5) and a stride of 2 in both dimensions. When using **PhaseDiff** as the input feature, the shape of the input feature is formed into $\mathbb{R}^{12 \times F \times T}$ by stacking a total of six patterns of combinations of the ith and jth channels along the channel dimension direction stacked in (7).

Two loss patterns, **MSE** and **MSE+Matching**, were used as loss functions, no weights were given between estimated labels and between losses, and the batch size K was set to 16. During training, the learning rate for pre-training and without pre-training learning was set to 0.00005, and the learning rate for fine-tuning was set to 0.0005, optimized by Adam [7]. We trained the model for 100 epochs and selected the best checkpoint based on validation loss. KTRC and RMSE were used as evaluation metrics and were evaluated for four estimated labels: car-l2r, car-r2l, CV-l2r, and CV-r2l.

[1] Number of passenger vehicles going left to right per minute.

[2] Number of passenger vehicles going right to left per minute.

[3] Number of commercial vehicles going left to right per minute.

[4] Number of commercial vehicles going right to left per minute.

3.2. Experimental results

Tables 1 to 6 show the performance of vehicle counts for each location with pre-training. These results show that estimation performance varies significantly from location to location and that the best combination of input features and loss functions also varies. When the MSE+Matching was introduced as the loss function and the LogPowSpec+PhaseDiff as input features, the estimation performance was promising, particularly for locations 2 and 6. Therefore, we have chosen this as our proposed method and submitted the proposed system to DCASE 2024 Challenge Task 10. Our proposed **MSE+Matching** loss showed some improvements not only in KTRC but also in RMSE.

Table 7 show the performance of vehicle counts for location 6 without pre-training. Tables 6 and 7 confirm that pre-training improves estimation performance.

4. CONCLUSIONS

This report summarizes our approach for DCASE 2024 Challenge Task 10, acoustic-based traffic monitoring. Our approach of introducing matching loss for the ranking function into the CRNN's loss function showed some improvements not only in KTRC but also in RMSE. Our proposed method, which introduces matching loss as the loss function, along with logarithmic power spectrogram and the cosine and sine of the phase difference as input features, performed well on the development dataset, particularly for locations 2 and 6. In addition, experimental evaluation confirmed that pre-training improves estimation performance. Improving estimation accuracy in locations with low estimation accuracy is a future work.

5. REFERENCES

[1] S. Damiano, L. Bondi, S. Ghaffarzadegan, A. Guntoro, and T. van Waterschoot, "Can synthetic data boost the training of deep acoustic vehicle counting networks?" in *Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 631–635, 2024.

[2] S. Damiano and T. van Waterschoot, "Pyroadacoustics: a road acoustics simulator based on variable length delay lines," in *Proceedings of the 25th International Conference on Digital Audio Effects (DAFx22)*, pp. 216–223, 2022.

[3] D. Helmbold, J. Kivinen, and M. Warmuth, "Relative loss bounds for single neurons," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1291–1304, 1999.

[4] T. Takahashi, Y. Kinoshita, Y. Wakabayashi, N. Ono, J. Honda, S. Fukuma, A. Kitamori, and H. Nakagawa, "Acoustic traffic monitoring based on deep neural network trained by stereo-recorded sound and sensor data," in *Proceedings of the 31st European Signal Processing Conference (EUSIPCO)*, pp. 935–939, 2023.

[5] T. Takahashi, Y. Kinoshita, N. Ueno, Y. Wakabayashi, N. Ono, J. Honda, S. Fukuma, A. Kitamori, and H. Nakagawa, "Augmentation of various speed data by controlling frame overlap for acoustic traffic monitoring," in *Proceedings of Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 2068–2072, 2023.

[6] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 15 pages, 2015.

Table 1: Performance of vehicle counts for location 1 with pre-training

Loc.	Input	Loss	Pre-tr.	↑ Kendall's Tau Rank Corr				↓ RMSE			
				car-l2r	car-r2l	CV-l2r	CV-r2l	car-l2r	car-r2l	CV-l2r	CV-r2l
1	LogMelSpec+GCC-PHAT	MSE	✓	**0.415**	0.423	0.164	0.153	**2.619**	2.966	0.999	0.901
1	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.392	0.447	0.136	**0.172**	2.662	2.914	0.922	**0.868**
1	LogPowSpec+PhaseDiff	MSE	✓	0.39	**0.455**	**0.182**	0.129	2.689	**2.894**	**0.88**	0.884
1	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.403	0.433	0.13	0.118	2.642	2.946	0.949	0.875

Table 2: Performance of vehicle counts for location 2 with pre-training

Loc.	Input	Loss	Pre-tr.	↑ Kendall's Tau Rank Corr				↓ RMSE			
				car-l2r	car-r2l	CV-l2r	CV-r2l	car-l2r	car-r2l	CV-l2r	CV-r2l
2	LogMelSpec+GCC-PHAT	MSE	✓	0.768	0.409	**0.201**	0.026	**1.868**	2.627	**0.815**	0.678
2	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.685	0.376	0.086	-0.002	2.466	2.832	0.862	0.715
2	LogPowSpec+PhaseDiff	MSE	✓	0.685	0.462	-0.003	0.015	2.501	2.478	0.863	0.729
2	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.774**	**0.623**	0.128	**0.179**	1.9	**1.951**	0.824	**0.623**

Table 3: Performance of vehicle counts for location 3 with pre-training

Loc.	Input	Loss	Pre-tr.	↑ Kendall's Tau Rank Corr				↓ RMSE			
				car-l2r	car-r2l	CV-l2r	CV-r2l	car-l2r	car-r2l	CV-l2r	CV-r2l
3	LogMelSpec+GCC-PHAT	MSE	✓	0.545	0.578	**0.197**	**0.381**	1.739	1.281	0.3	**0.199**
3	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.548	**0.584**	0.081	-0.008	1.73	**1.275**	0.308	0.22
3	LogPowSpec+PhaseDiff	MSE	✓	**0.557**	**0.584**	0.191	0.226	**1.726**	1.286	**0.293**	0.224
3	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.548	0.582	-0.028	-0.03	1.743	1.284	0.359	0.241

Table 4: Performance of vehicle counts for location 4 with pre-training

Loc.	Input	Loss	Pre-tr.	↑ Kendall's Tau Rank Corr				↓ RMSE			
				car-l2r	car-r2l	CV-l2r	CV-r2l	car-l2r	car-r2l	CV-l2r	CV-r2l
4	LogMelSpec+GCC-PHAT	MSE	✓	0.439	-0.013	-0.061	**0.592**	1.641	1.666	0.797	0.67
4	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.585	**0.467**	0.114	0.562	1.622	**0.801**	**0.501**	**0.41**
4	LogPowSpec+PhaseDiff	MSE	✓	**0.658**	-0.189	**0.251**	-0.197	**1.502**	2.16	0.667	0.57
4	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.049	-0.013	-0.203	0.07	2.406	1.951	0.739	0.626

Table 5: Performance of vehicle counts for location 5 with pre-training

Loc.	Input	Loss	Pre-tr.	↑ Kendall's Tau Rank Corr				↓ RMSE			
				car-l2r	car-r2l	CV-l2r	CV-r2l	car-l2r	car-r2l	CV-l2r	CV-r2l
5	LogMelSpec+GCC-PHAT	MSE	✓	0.428	**0.498**	0.068	0.156	**0.771**	**0.619**	0.402	**0.187**
5	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.303	0.091	-0.063	**0.59**	0.827	0.886	0.374	0.234
5	LogPowSpec+PhaseDiff	MSE	✓	0.032	0.163	**0.157**	0.328	0.972	0.842	**0.352**	0.245
5	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.498**	0.283	-0.101	0.095	0.785	0.781	0.368	0.275

Table 6: Performance of vehicle counts for location 6 with pre-training

Loc.	Input	Loss	Pre-tr.	↑ Kendall's Tau Rank Corr				↓ RMSE			
				car-l2r	car-r2l	CV-l2r	CV-r2l	car-l2r	car-r2l	CV-l2r	CV-r2l
6	LogMelSpec+GCC-PHAT	MSE	✓	0.849	0.737	0.788	0.729	1.337	1.663	0.443	0.466
6	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.845	0.726	0.808	0.744	1.394	1.697	0.452	0.458
6	LogPowSpec+PhaseDiff	MSE	✓	0.827	0.713	0.753	0.681	1.507	1.748	0.519	0.511
6	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.854**	**0.738**	**0.821**	**0.761**	**1.288**	**1.607**	**0.433**	**0.451**

Table 7: Performance of vehicle counts for location 6 without pre-training

Loc.	Input	Loss	Pre-tr.	↑ Kendall's Tau Rank Corr				↓ RMSE			
				car-l2r	car-r2l	CV-l2r	CV-r2l	car-l2r	car-r2l	CV-l2r	CV-r2l
6	LogMelSpec+GCC-PHAT	MSE	—	**0.824**	0.709	0.78	**0.714**	**1.526**	**1.777**	0.514	**0.491**
6	LogMelSpec+GCC-PHAT	MSE+Matching	—	0.816	**0.71**	**0.788**	0.706	1.596	1.814	0.516	0.533
6	LogPowSpec+PhaseDiff	MSE	—	0.773	0.668	0.651	0.502	1.829	2.012	0.649	0.656
6	LogPowSpec+PhaseDiff	MSE+Matching	—	0.803	0.693	0.786	0.71	1.633	1.864	**0.509**	0.504