ACOUSTIC-BASED TRAFFIC MONITORING WITH NEURAL NETWORK TRAINED BY MATCHING LOSS FOR RANKING FUNCTION

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 study, we propose an effective loss function for training neural networks (NNs) in acoustic-based traffic monitoring. This task involves estimating the number of vehicles from a fixed duration of acoustic input, such as one minute. Since the distribution of the number of passing vehicles depends on the road and can deviate significantly from a normal distribution, using Mean Square Error (MSE) as the loss function may not always lead to efficient learning. To address this, we introduce a matching loss for the ranking function into the loss function. This enhances learning by increasing the rank correlation between true and estimated vehicle counts. We evaluated the effectiveness of this loss function on the development dataset of the DCASE 2024 Challenge Task 10 under various input feature and network architecture conditions. The results demonstrate that the proposed loss function significantly improves Kendall's Tau Rank Correlation (KTRC) and Root Mean Square Error (RMSE), highlighting its potential for improving acoustic-based traffic monitoring systems.

***Index Terms*—** matching loss, traffic monitoring, vehicle counting, deep neural network, acoustic sensing, microphone array

1. INTRODUCTION

Measuring road traffic conditions, including traffic volume, speed, density, time occupancy, vehicle type, and direction of travel, is essential for understanding real-time traffic situations. Traffic monitoring systems that provide this information to road traffic control systems and users are also crucial for smart city development [1,2]. Various sensors can be used for traffic monitoring, including intrusive systems embedded in the road (e.g., loop coils [3], vibration sensors), non-intrusive systems mounted over or on the side of the road (e.g., radar, cameras, infrared sensors, acoustic sensors), and off-road mobile devices (e.g., aircraft, satellites) [4].

Although acoustic sensors, that is, microphones installed on the roadside are not the most common method, they offer several advantages such as low installation and maintenance costs, low energy consumption, non-intrusiveness, and insensitivity to obstructions, shadows, and lighting conditions. Approaches to acoustic-based traffic monitoring are roughly classified into two types: rule-based [5–9] and machine-learning-based [10–19]. In particular, one of the latters, [18], provides a baseline system used in DCASE 2024 Challenge, the major international competition in the field of acoustic scene and event understanding. It represents the growing interest in recent years in machine-learning-based acoustic traffic monitoring. Various data augmentation methods have been proposed [20–29], especially, [18] investigates the effectiveness of data augmentation by an open-source road acoustic simulator [30].

In this study, following the DCASE 2024 Task 10 setup, we address the problem of counting the number of vehicles per vehicle type (car or Commercial Vehicle, CV) and per direction of travel (left or right) from acoustic signals. Specifically, we focus on the loss function for training a neural network (NN). Mean Square Error (MSE) is a common loss function for this task. However, using MSE may not be optimal for counting vehicles because the distribution of passing vehicles varies depending on the road and can significantly deviate from a normal distribution. Therefore, we propose a new loss function that aims to increase the rank correlation between the true and estimated number of vehicles since the rank correlation is a nonparametric measure and does not depend on the data distribution. This loss function is derived by applying the concept of matching loss [31] to the ranking function.

We conducted an experimental evaluation for checking the effectiveness of our loss function using training, validation, and synthetic data from the DCASE 2024 Challenge Task 10 development dataset [17]. We also compared several combinations of input acoustic features and two network architectures. Evaluated using Kendall's Tau Rank Correlation (KTRC) and Root Mean Square Error (RMSE), our loss function showed improvement in both metrics. Additionally, we assessed the estimation performance with and without pre-training. The results confirmed that pre-training enhanced the estimation performance.

This paper is organized as follows. Section 2 describes the input features and network architectures considered in this study of acoustic-based traffic monitoring with NN. Section 3 introduces the proposed loss function. Section 4 presents the experimental evaluation and results. Section 5 provides the conclusion.

2. INPUT FEATURES AND NETWORK ARCHITECTURES

In acoustic-based traffic monitoring with NN, the feature extraction from the input acoustic signal and the structure of the network architecture are crucial for efficient learning. In this section, we describe several input features and network architectures, whose effectivess has been confirmed in the baseline system of the DCASE 2024 Challenge Task 10 [17,18] and our previous work [19,20]. We will compare the performance of these various combinations in the experimental evaluation section 4.

2.1. Input features

Suppose that we have multiple microphones and let $\mathbf{x}_i \in \mathbb{R}^N$ be the acoustic signal of traffic sound captured by the ith microphone.

We express the short-time Fourier transform as

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

where $\mathbf{X}_i \in \mathbb{C}^{F\times T}$ is the complex-valued spectrogram with F frequency bins and T time frames. The amplitude of $\mathbf{X}_i$ and the phase difference between $\mathbf{X}_i$ and $\mathbf{X}_j$ are known to be the basis for effective features in acoustic-based traffic monitoring as follows.

LogMelSpec: $\mathbf{X}_i^{\mathrm{LMS}} \in \mathbb{R}^{M\times T}$ is calculated by taking the logarithm of the mel-scale transform of the amplitude spectrogram of $\mathbf{X}_i$, and M specifies the number of mel frequency bands.

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 by

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

Here, f and t represent 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 [32]:

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

Here, $\mathcal{F}^{-1}_{f\to\tau}$ is the inverse Fourier transform from f to τ. The indices i,j refer to distinct channels, and G specifies the number of GCC-PHAT coefficients.

PhaseDiff: $\mathbf{X}_{i,j}^{\mathrm{PDC}} \in \mathbb{R}^{F\times T}$ and $\mathbf{X}_{i,j}^{\mathrm{PDS}} \in \mathbb{R}^{F\times T}$, where the (f,t) elements, $\mathrm{X}_{i,j}^{\mathrm{PDC}}(f,t)$ and $\mathrm{X}_{i,j}^{\mathrm{PDS}}(f,t)$, are calculated using the following equations [33]:

$$\Delta\phi_{i,j}(f,t) = \arg(\mathrm{X}_i(f,t)/\mathrm{X}_j(f,t)), \tag{4}$$
$$\mathrm{X}_{i,j}^{\mathrm{PDC}}(f,t) = \cos(\Delta\phi_{i,j}(f,t)), \tag{5}$$
$$\mathrm{X}_{i,j}^{\mathrm{PDS}}(f,t) = \sin(\Delta\phi_{i,j}(f,t)). \tag{6}$$

2.2. Network architecture

Convolutional Neural Network (CNN)-based architecture is known to be effective in acoustic-based traffic monitoring. In this study, we consider the following two types of the CNN architectures.

CRNN: The amplitude-related and phase-related input features stack in the direction of the newly added channel dimension and are passed separately through network branches with the same structure. The network branches consist of convolutional encoders and Time-Distributed Multi-Layer Perceptrons (TD-MLP), composed of multiple Conv2D layers and a fully connected (FC) layer, respectively. TD-MLP is independently applied to each time frame of their input. Features that pass through each branch are concatenated and processed by a further TD-MLP layer, followed by a Gated Recurrent Unit (GRU) and an FC layer to regress labels.

ConvMixer: Each channel's amplitude-related and phase-related input features are concatenated in the direction of the frequency dimension and passed through the embedding layer, the ConvMixer layer [34, 35], and the classifier layer. In the embedding layer, patch embedding is applied to input features by handling short time frames in the input as patches. ConvMixer layer has a repeating network structure consisting of a pointwise convolution, which mixes the features for each time frame using an FC layer, and a depthwise convolution, which mixes the features by convolving in the direction of the time frame. In the classifier layer, the output of the ConvMixer layer is finally aggregated in the time-frame direction by the average pooling layer and then transformed into a one-dimensional vector representing the label by the FC layer.

3. PROPOSED LOSS FUNCTION

Let $y_k^{(*)}$ and $\hat{y}_k^{(*)} \in \mathbb{R}$ be the true and estimated vehicle counts for data k and label $(*) \in \{\text{car-l2r}, \text{car-r2l}, \text{CV-l2r}, \text{CV-r2l}\}$[1], respectively, where $k = 1,\dots,K$ is the data index within a batch of size $K \in \mathbb{N}$. Let $\mathbf{y}^{(*)}$ and $\hat{\mathbf{y}}^{(*)} \in \mathbb{R}^K$ be the collections of true and estimated counts, respectively, for all data $k = 1,\dots,K$.

A loss function is used to evaluate the performance of a model by quantifying the closeness between the true value $y_k^{(*)}$ and its estimate $\hat{y}_k^{(*)}$. One of the most common loss functions is the MSE such as

MSE:

$$L_{\mathrm{MSE}}(\hat{\mathbf{y}}^{(*)};\mathbf{y}^{(*)}) = \frac{1}{K}\sum_{k=1}^{K}(y_k^{(*)} - \hat{y}_k^{(*)})^2. \tag{7}$$

While, the key idea of our proposed loss function is to induce the correspondence between true and estimated order relations of data, aiming to enhance learning efficiently. Here, $\varphi : \mathbb{R}^K \to \mathbb{Z}^K$ is referred to as the ranking function, defined as

$$\varphi(\mathbf{y}^{(*)}) = \sum_{k=1}^{K}\begin{bmatrix}\mathrm{sign}(\hat{y}_1^{(*)} - \hat{y}_k^{(*)})\\ \vdots \\ \mathrm{sign}(\hat{y}_K^{(*)} - \hat{y}_k^{(*)})\end{bmatrix}. \tag{8}$$

Intuitively, $[\varphi(\mathbf{y}^{(*)})]_k$ denotes the number of elements smaller than $\hat{y}_k^{(*)}$ minus the number of elements larger than $\hat{y}_k^{(*)}$. Thus, φ maps the ranking of the input vector $\mathbf{y}^{(*)}$ into the integers within $\{-K+1,\dots,K-1\}$. If we want to bring $\varphi(\hat{\mathbf{y}}^{(*)})$ and $\varphi(\mathbf{y}^{(*)})$ closer, a straightforward loss function could be $||\varphi(\hat{\mathbf{y}}^{(*)}) - \varphi(\mathbf{y}^{(*)})||_2^2$, where $||\cdot||_2$ denotes the L_2 norm of a vector. However, this function cannot be used as a loss function due to the discontinuity and zero gradient of the function φ. Instead, we propose the following loss function.

Matching:

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

[1]The number of passenger cars or commercial vehicles (CVs) moving from left to right or right to left per minute.

Note that L_{Matching} is convex with respect to its first variable $\hat{\mathbf{y}}^{(*)}$, and its subgradient of L_{Matching} is provided by

$$\nabla L_{\mathrm{Matching}}(\hat{\mathbf{y}}^{(*)}; \mathbf{y}^{(*)}) = \frac{1}{K^2}\left(\varphi(\hat{\mathbf{y}}^{(*)}) - \varphi(\mathbf{y}^{(*)})\right). \quad (10)$$

It means that minimizing L_{Matching} induces a correspondence between $\varphi(\hat{\mathbf{y}}^{(*)})$ and $\varphi(\mathbf{y}^{(*)})$, i.e., the ranking of the true and estimated data. A mathematical relationship like that between the loss function L_{Matching} and the vector-valued function φ is generally referred to as the matching loss [31]. Motivated by this concept, we incorporate the matching loss for the ranking function as described above to enhance the correspondence between the ranking of the true and estimated data.

Note that the ranking is invariant to bias, meaning that $\varphi(\hat{\mathbf{y}}^{(*)}) = \varphi(\hat{\mathbf{y}}^{(*)} + c\mathbf{1})$ for any constant c, where $\mathbf{1}$ denotes the vector each of whose element is one. This indicates that using only L_{Matching} is insufficient for estimating the correct number of vehicles without bias. Therefore, finally, we propose to combine these loss functions such as

MSE+Matching:

$$L_{\mathrm{MSE+Matching}} = (1-\lambda)L_{\mathrm{MSE}} + \lambda L_{\mathrm{Matching}}. \quad (11)$$

where λ is a weight parameter for the combination.

4. EXPERIMENTAL EVALUATIONS

In this section, we present the conditions and results of an experiment to evaluate the effectiveness of our proposed loss function. The experiments were conducted using the input features and network architectures described in Section 2.

4.1. Experimental conditions

We trained our model using the training[2], validation[3], and synthetic[4] data from the DCASE 2024 Challenge Task 10 development dataset [17] and evaluated it using the validation data[5]. We referred to the baseline system of DCASE 2024 Challenge Task 10 [17, 18], using synthetic data exclusively for pre-training and real data for fine-tuning.

Two feature patterns were used as input features: **LogMelSpec+GCC-PHAT** and **LogPowSpec+PhaseDiff**, which combine amplitude-related and phase-related features. The number of channels is four $(i, j \in \{1, 2, 3, 4\})$, and the phase-related input features are computed between pairs of channels i and j. 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.

In the **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. The first TD-MLP has two layers with 128 neurons each, the second TD-MLP has three layers with 128 neurons each, the GRU has two layers with 128 neurons each, and the last FC layer has four neurons. In the **ConvMixer**, the number of embeddings was set to $T = 1875$, the feature vector dimension was 5, the kernel size of the depthwise convolution was 5, and the number of iterations was 5.

[2]7294 1-min training samples of real data collected from 6 sites.
[3]7705 1-min validation samples of real data collected from 6 sites.
[4]1224 1-min synthetic data generated via pyroadacoustics simulator [30].
[5]Since the labels for the DCASE 2024 Challenge Task 10 evaluation dataset are not yet publicly available, validation data were used [17].

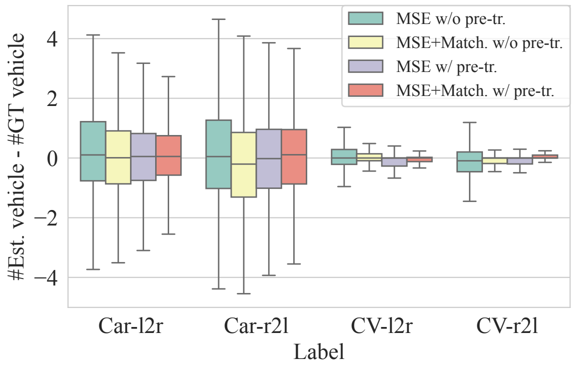

Figure 1: Error with and without **Matching** and pre-training

Two loss patterns were used as loss functions: **MSE** and **MSE+Matching**. During training, the average loss of each label was used to update the loss function, with the weighting coefficient λ of **MSE+Matching** set to 0.5, which was determined experimentally, and the batch size K set to 16. Additionally, 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 [36]. 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 labels: car-l2r, car-r2l, CV-l2r, and CV-r2l.

4.2. Experimental results

Table 1 shows the performance of vehicle counts for each location. **CRNN** performed better than **ConvMixer**, particularly in terms of CV accuracy and pre-training effectiveness. The estimation performance with pre-training, using **LogPowSpec+PhaseDiff**, **CRNN**, and **MSE+Matching**, was promising, particularly at location 6. Figure 1 shows the difference between true and estimated labels under these conditions, with and without **Matching** and pre-training. The results in the table and figure show that our proposed loss function and pre-training were confirmed to enhance estimation performance. Additionally, our proposed loss function improved not only in KTRC but also in RMSE. We submitted this best-performing condition system for the DCASE 2024 Challenge Task 10 [37].

5. CONCLUSIONS

In this study, we proposed a new loss function for acoustic-based traffic monitoring using NN. Our proposed loss function, derived from matching loss to the ranking function, aims to increase the rank correlation between the orders of true and estimated number of vehicles in each batch. We evaluated the effectiveness of our proposed loss function on the development dataset of the DCASE 2024 Challenge Task 10 and investigated good combinations of input acoustic features and network architectures. Our proposed loss function demonstrated improvements not only in KTRC but also in RMSE. As future work, our proposed loss function can be applied to other acoustics-based traffic monitoring tasks, such as traffic speed estimation.

Table 1: Performance of vehicle counts for each location

Loc.	Arc.	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	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.415	0.423	0.164	0.153	**2.619**	2.966	0.999	0.901
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.392	0.447	0.136	**0.172**	2.662	2.914	0.922	0.868
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	0.39	**0.455**	**0.182**	0.129	2.689	**2.894**	0.88	0.884
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.403	0.433	0.13	0.118	2.642	2.946	0.949	0.875
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.421	0.406	0.16	0.141	2.643	3.039	**0.837**	**0.835**
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	**0.429**	0.429	0.026	0.152	2.626	2.982	0.866	0.837
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.29	0.306	0.121	0.105	2.894	3.23	0.842	0.84
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.158	0.144	0.096	0.054	3.526	3.927	1.266	1.701
2	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.768	0.409	0.201	0.026	**1.868**	2.627	0.815	0.678
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.685	0.376	0.086	-0.002	2.466	2.832	0.862	0.715
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	0.685	0.462	-0.003	0.015	2.501	2.478	0.863	0.729
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.774**	**0.623**	0.128	**0.179**	1.9	**1.951**	0.824	**0.623**
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.415	0.361	**0.213**	-0.103	8.093	6.595	0.963	1.206
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.513	0.355	-0.118	0.045	4.384	3.429	1.766	1.003
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.44	0.318	0.174	-0.126	3.536	3.194	**0.73**	0.919
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.445	0.194	-0.193	-0.132	3.852	6.916	0.848	2.159
3	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.545	0.578	**0.197**	**0.381**	1.739	1.281	0.3	**0.199**
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.548	**0.584**	0.081	-0.008	1.73	**1.275**	0.308	0.22
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	**0.557**	**0.584**	0.191	0.226	**1.726**	1.286	**0.293**	0.224
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.548	0.582	-0.028	-0.03	1.743	1.284	0.359	0.241
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.478	0.442	0.097	0.117	1.872	1.478	0.305	0.234
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.49	0.449	0.064	0.052	1.851	1.477	0.319	0.263
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.494	0.486	0.079	0.141	1.845	1.43	0.3	0.218
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.472	0.419	-0.044	0.008	1.87	1.514	0.298	0.215
4	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.439	-0.013	-0.061	0.592	1.641	1.666	0.797	0.67
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.585	**0.467**	0.114	0.562	1.622	**0.801**	**0.501**	**0.41**
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	**0.658**	-0.189	0.251	-0.197	**1.502**	2.16	0.667	0.57
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.049	-0.013	-0.203	0.07	2.406	1.951	0.739	0.626
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.512	0.038	-0.266	-0.055	1.892	1.905	0.751	0.593
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.366	-0.164	0.301	-0.602	1.783	2.511	1.321	0.604
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.341	0.29	**0.408**	**0.602**	9.199	1.699	1.097	1.702
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.073	-0.215	0.301	0.456	2.048	1.355	0.592	0.917
5	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.428	**0.498**	0.068	0.156	**0.771**	**0.619**	0.402	**0.187**
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.303	0.091	-0.063	**0.59**	0.827	0.886	0.374	0.234
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	0.032	0.163	**0.157**	0.328	0.972	0.842	**0.352**	0.245
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.498**	0.283	-0.101	0.095	0.785	0.781	0.368	0.275
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.129	0.004	0.096	-0.134	1.013	0.889	0.372	0.407
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	-0.16	-0.149	-0.001	-0.107	1.271	0.842	0.769	0.278
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.092	0.025	-0.048	0.135	0.947	0.852	0.357	0.284
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.042	0.014	0.072	-0.034	0.947	1.006	0.372	0.28
6	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.849	0.737	0.788	0.729	1.337	1.663	0.443	0.466
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.845	0.726	0.808	0.744	1.394	1.697	0.452	0.458
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	0.827	0.713	0.753	0.681	1.507	1.748	0.519	0.511
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.854**	**0.738**	**0.821**	**0.761**	**1.288**	**1.607**	**0.433**	**0.451**
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.428	0.459	0.313	0.335	3.712	2.854	0.976	0.824
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.495	0.43	0.281	0.297	3.409	2.925	1.001	0.791
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.494	0.49	0.404	0.308	3.368	2.764	0.955	0.798
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.671	0.517	0.285	-0.069	2.584	2.707	1.002	0.857
6	CRNN	LogMelSpec+GCC-PHAT	MSE	—	**0.824**	0.709	0.78	**0.714**	**1.526**	**1.777**	0.514	**0.491**
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	—	0.816	**0.71**	**0.788**	0.706	1.596	1.814	0.516	0.533
	CRNN	LogPowSpec+PhaseDiff	MSE	—	0.773	0.668	0.651	0.502	1.829	2.012	0.649	0.656
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	—	0.803	0.693	0.786	0.71	1.633	1.864	**0.509**	0.504
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	—	0.752	0.659	0.252	0.141	1.968	1.998	1.005	0.835
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	—	0.763	0.651	0.297	0.19	2.313	2.077	1.009	0.834
	ConvMixer	LogPowSpec+PhaseDiff	MSE	—	0.774	0.65	0.343	0.118	1.945	2.38	0.966	0.847
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	—	0.744	0.601	0.249	0.148	2.403	2.474	1.036	0.852

6. REFERENCES

[1] R. Du, P. Santi, M. Xiao, A. V. Vasilakos, and C. Fischione, "The sensable city: A survey on the deployment and management for smart city monitoring," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1533–1560, 2019.

[2] M. Won, "Intelligent traffic monitoring systems for vehicle classification: A survey," *IEEE Access*, vol. 8, pp. 73 340–73 358, 2020.

[3] B. Coifman and S. Neelisetty, "Improved speed estimation from single-loop detectors with high truck flow," *Journal of Intelligent Transportation Systems*, vol. 18, no. 2, pp. 138–148, 2014.

[4] P. T. Martin, Y. Feng, and X. Wang, "Detector technology evaluation," Mountain-Plains Consortium, MPC Report 03-154, 2003.

[5] M. A. Sobreira-Seoane, A. Rodríguez Molares, and J. L. Alba Castro, "Automatic classification of traffic noise," *The Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 3823–3823, 2008.

[6] G. Szwoch and J. Kotus, "Acoustic detector of road vehicles based on sound intensity," *Sensors*, vol. 21, no. 23, 7781 (18 pages), 2021.

[7] J. Kotus and G. Szwoch, "Estimation of average speed of road vehicles by sound intensity analysis," *Sensors*, vol. 21, no. 16, 5337 (18 pages), 2021.

[8] T. Toyoda, N. Ono, S. Miyabe, T. Yamada, and S. Makino, "Traffic monitoring with ad-hoc microphone array," in *Proceedings of the 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 318–322, 2014.

[9] S. Ishida, K. Mimura, S. Liu, S. Tagashira, and A. Fukuda, "Design of simple vehicle counter using sidewalk microphones," in *Proceedings of the ITS European Congress*, 10 pages, 2016.

[10] A. Y. Nooralahiyan, M. Dougherty, D. McKeown, and H. R. Kirby, "A field trial of acoustic signature analysis for vehicle classification," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 3-4, pp. 165–177, 1997.

[11] J. George, L. Mary, and K. S. Riyas, "Vehicle detection and classification from acoustic signal using ANN and KNN," in *Proceedings of the International Conference on Control Communication and Computing (ICCC)*, pp. 436–439, 2013.

[12] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular traffic density state estimation based on cumulative road acoustics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1156–1166, 2012.

[13] J. Abeßer, S. Gourishetti, A. Kátai, T. Clauß, P. Sharma, and J. Liebetrau, "IDMT-traffic: An open benchmark dataset for acoustic traffic monitoring research," in *Proceedings of the 29th European Signal Processing Conference (EUSIPCO)*, pp. 551–555, 2021.

[14] S. Djukanović, Y. Patel, J. Matas, and T. Virtanen, "Neural network-based acoustic vehicle counting," in *Proceedings of the 29th European Signal Processing Conference (EUSIPCO)*, pp. 561–565, 2021.

[15] N. Bulatović and S. Djukanović, "Mel-spectrogram features for acoustic vehicle detection and speed estimation," in *Proceedings of the 26th International Conference on Information Technology (IT)*, 4 pages, 2022.

[16] T. Shinohara, Y. Wakabayashi, R. Scheibler, N. Ono, N. Aizawa, and H. Nakagawa, "Sound-based speed estimation using a neural network," in *Proceedings of the Spring meeting of the Acoustical Society of Japan*, pp. 385–386, 2020, (in Japanese).

[17] S. Ghaffarzadegan, L. Bondi, W.-C. Lin, A. Kumar, H.-H. Wu, H.-G. Horst, and S. Das, "Sound of traffic: A dataset for acoustic traffic identification and counting," in *Technical Report of the DCASE2024 Challenge*, 2024.

[18] 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.

[19] 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.

[20] 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.

[21] 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 the Interspeech*, pp. 2613–2617, 2019.

[22] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13 001–13 008, 2020.

[23] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proceedings of the International Conference on Learning Representations*, 13 pages, 2018.

[24] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, 2019.

[25] G. Kim, D. K. Han, and H. Ko, "SpecMix: A mixed sample data augmentation method for training with time-frequency domain features," in *Proceedings of the Interspeech*, pp. 546–550, 2021.

[26] H. Wang, Y. Zou, and W. Wang, "SpecAugment++: A hidden space data augmentation method for acoustic scene classification," in *Proceedings of the Interspeech*, pp. 551–555, 2021.

[27] H. Nam, S.-H. Kim, and Y.-H. Park, "Filteraugment: An acoustic environmental data augmentation method," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4308–4312, 2022.

[28] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3008–3017, 2020.

[29] J. Han, M. Matuszewski, O. Sikorski, H. Sung, and H. Cho, "Randmasking augment: A simple and randomized data augmentation for acoustic scene classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5 pages, 2023.

[30] 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.

[31] 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.

[32] 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.

[33] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5 pages, 2018.

[34] A. Trockman and J. Z. Kolter, "Patches are all you need?" *arXiv preprint arXiv:2201.09792*, 2022.

[35] R. Baidya and H. Jeong, "YOLOv5 with ConvMixer prediction heads for precise object detection in drone imagery," *Sensors*, vol. 22, no. 21, 8424 (17 pages), 2022.

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

[37] T. Takahashi, N. Ueno, Y. Kinoshita, Y. Wakabayashi, N. Ono, M. Sukekawa, S. Fukuma, and H. Nakagawa, "Neural network training with matching loss for ranking function," in *Technical Report of the DCASE2024 Challenge*, 2024.