# Feature Alignment for Robust Acoustic Scene Classification across Devices

Jingqiao Zhao, Qiuqiang Kong, Xiaoning Song*, *Member, IEEE*, Zhenhua Feng*, *Member, IEEE* and Xiaojun Wu, *Member, IEEE*

***Abstract*—This letter presents a feature alignment method for domain adaptive Acoustic Scene Classification (ASC) across recording devices. First, we design a two-stream network, in which each stream processes two features, *i.e.*, Log-Mel spectrogram and delta-deltas, using two sub-networks. Second, we investigate different loss functions for feature alignment between the feature maps obtained by the source and target domains. Last, we present an alternate training strategy to deal with the data imbalance problem between paired and unpaired samples. The experimental results obtained on the DCASE benchmarks demonstrate the effectiveness and superiority of the proposed method. The source code of the proposed method is available at https://github.com/Jingqiao-Zhao/FAASC.**

***Index Terms*—Acoustic Scene Classification, domain adaption, feature alignment.**

## I. Introduction

Acoustic Scene Classification (ASC) is a popular research topic in computational auditory scene analysis. Given an audio sequence, ASC aims to detect and classify the environment in which the audio was recorded. ASC has many practical applications, such as IoT services [1] and surveillance systems [2].

During the last decades, a wide spectrum of ASC methods have been proposed. Classical ASC approaches usually use hand-crafted features, such as Mel Frequency Cepstral Coefficients (MFCCs), and shallow classifiers, *e.g.,* Hidden Markov Models (HMMs) [3], Support Vector Machine (SVM) [4] and decision trees [5]. In recent years, deep learning has become the mainstream in ASC due to its promising performance in various benchmarks and competitions. In general, an audio sequence is first converted to a 2D image using a time-frequency analysis method, such as wavelet transform, short-time Fourier transform and Log-Mel spectrogram, as the input of a deep Convolutional Neural Network (CNN). Then various networks, such as VGG [6] and ResNet [7], can be used to extract deep features for ASC [8]–[10]. Ren *et al.* demonstrated that the use of large receptive fields is more conducive for the task [11]. Bai *et al.* improved the performance of CNN by selecting relevant acoustic scene segments and using a two-stage attention strategy [12].

This work was supported by the National Natural Science Foundation of China (61876072, 61902153). The authors also would like to thank Mr Yang Hua for his contribution to the revision of the manuscript.

* Corresponding authors.

J. Zhao, X. Wu and X. Song are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China (e-mail: author@boulder.nist.gov).

Q. Kong and Z. Feng are with the School of Computer Science and Electronic Engineering, University of Surrey, Guildford GU2 7XH, UK (e-mail: {q.kong; z.feng}@surrey.ac.uk).

Most of the existing methods focus on ASC for a single device. However, in practice, different audio sequences are usually recorded by a variety of devices, leading to inconsistency across the captured signals. In this case, the performance of ASC methods may degrade significantly. To address this issue, one solution is to use ensemble modeling [13], [14]. However, this strategy is expensive, which increases the computational complexity significantly thus not suitable for real-time ASC tasks. The use of ensemble methods is not encouraged by many competitions such as DCASE2021. Another popular solution is domain adaption. Suppose we have two different datasets with different distributions (source and target domains), domain adaption aims to transfer the knowledge learned from the source domain to the target one, hence improving the generalization capability of a trained model across different data distributions.

The existing domain adaption methods can be divided into three categories [15]. The first one is instance adaptation that applies weighted re-sampling to the samples of the source domain for the approximation of the distribution of the target domain [16]. For example, a weighted local domain adaptive method was used to establish a connection between the source and target domains [17]. The second category is feature adaptation [18], [19]. This method aims to project the data of the source and target domains into a common feature space. For example, we can use multiple features obtained from the source domain to extract target domain features by establishing saddle points to improve the classification accuracy [20]. The last one is model adaptation, which modifies the loss function in the source domain to match the loss in the target domain [21].

This letter addresses the domain adaptation problem for ASC using feature alignment. Specifically, we achieve domain adaptation by investigating different alignment positions and different alignment loss functions for the proposed two-stream network. Furthermore, the numbers of samples collected by different devices may vary significantly, leading to a serious data imbalance issue. To mitigate this problem, we propose an alternate training strategy. The main contributions of the proposed method include:

- We design a two-stream network that performs ASC using different features in the source and target domains.
- We investigate different feature alignment loss functions and alignment positions for the proposed network.
- We deal with the training data imbalance problem using an alternate training strategy.

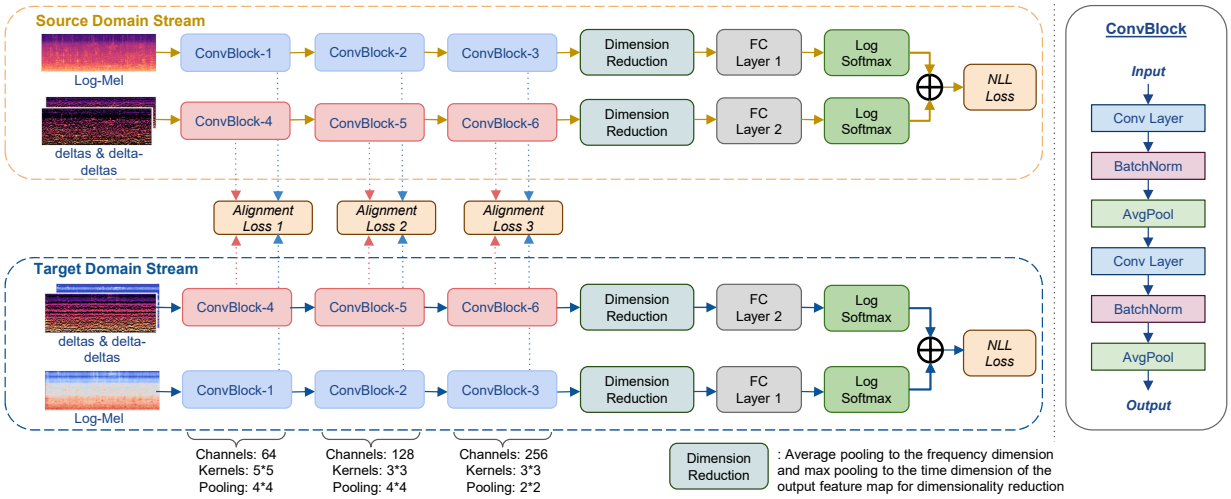



Fig. 1: The overall architecture of the proposed domain adaptation method using feature alignment.

## II. The Proposed Method

This section first introduces the architecture of the proposed network. Then we present the proposed domain adaptation approach, including training data partition, feature alignment and alternative network training.

### A. Network Architecture

The proposed network architecture is shown in Fig. 1. It has two main streams for the source and target domains, respectively. Note that the parameters of the two streams are shared. For each stream, there are two sub-networks for acoustic scene classification. The reason is that different acoustic features have different characteristics thus should be processed by different network configurations [22]. Specifically, we use three types of acoustic features. One branch uses a two-channel input with the deltas and delta-deltas features as input, and the other one uses the Log-Mel spectrogram as input.

Each branch in a stream has three convolutional blocks, consisting of two convolutional layers followed by the batch normalization and average pooling layers. After the convolutional blocks, we apply average pooling to the frequency dimension and max pooling to the time dimension of the output feature map for dimensionality reduction, by which a $C \times H \times W$ tensor is reduced to $C \times 1 \times 1$. Then we use the log-softmax layer for nonlinear activation and the Negative Log-Likelihood (NLL) loss for network training:

$$l\left(p, y\right) = -\sum_{k=1}^{K} y_k * p_k / K, \tag{1}$$

where $y = (y_1, ..., y_K) \in \{0, 1\}^K$ is the class label and $K$ is the number of classes. The prediction $p = (p_1, ..., p_K) \in [0, 1]^K$ is the predicted probability of sound classes.

Besides, several feature alignment loss functions are used to perform domain adaptation across different devices, as introduced in Section II-C.

### B. Training Data Partition

To perform domain adaptive network training, the source stream only processes the samples captured by one device, and the target stream processes all the samples captured by the other devices. Note that we intercept the same location clips recorded by different devices to obtain the samples for network training. Although there might be a slight difference between the arrival times of an event at the microphones of two devices, the recorded audios can be considered identical in time.

It should be noted that, to perform feature alignment across domains, a sample in the target stream should have the corresponding sample in the source stream. To meet this requirement, we split the training samples into paired and unpaired ones. We denote a source domain training sample as $\{X_s, Y_s\}$ and a target domain training sample as $\{X_t, Y_t\}$, where $X$ is the input data and $Y$ is the class label.

The paired data includes the samples captured by more than one device. The device with the maximal number of samples is used as the inputs of the source stream and the others are used as the inputs of the target stream. The unpaired samples are captured by only one device. We only use the unpaired samples for the training of the source stream network. We use the paired samples for the training of the source and target streams to achieve feature alignment across recording devices.

### C. Feature Alignment

To perform feature alignment in domain adaptive network training, the most widely used loss functions are distribution losses, such as the Maximum Mean Discrepancy (MMD) and Kullback-Leibler (KL) divergence [23]. However, the differences among different acoustic recording devices may not be represented by the variations in the distributions of different domains. The differences can be represented in the spectral energy value of the recording equipment. In this case, the use of distribution loss functions may not perform well. To mitigate this issue, this letter investigates different loss functions, including MMD, KL, Mean Squared Error (MSE), and L1 loss functions for feature alignment. To be specific, we denote $\mathcal{F} \in \mathbb{R}^{W \times H \times D} \sim P$ and $\tilde{\mathcal{F}} \in \mathbb{R}^{W \times H \times D} \sim Q$ as

the feature maps obtained by the source and target streams, respectively. The above loss functions are defined as below:

$$MMD(P,Q) = ||\gamma(\mathcal{F}) - \gamma(\tilde{\mathcal{F}})||_{\mathcal{H}}, \tag{2}$$

$$KL(P||Q) = \sum_{i=1}^{M} p_i log(\frac{p_i}{q_i}), \tag{3}$$

$$MSE(\mathcal{F},\tilde{\mathcal{F}}) = \sum_{i=1}^{M}(\mathcal{F}_i - \tilde{\mathcal{F}}_i)^2/M, \tag{4}$$

$$L1(\mathcal{F},\tilde{\mathcal{F}}) = \sum_{i=1}^{M}|\mathcal{F}_i - \tilde{\mathcal{F}}_i|/M, \tag{5}$$

where $\gamma(\cdot)$ is a function that maps the data to the reproducing kernel Hilbert space $\mathcal{H}$ with Gaussian kernel as the default setting, $M = W \times H \times D$ is the total number of elements in each feature map.

### D. Alternative Network Training

To fully exploit the paired and unpaired samples, we propose a two-stage alternative training strategy. We use $X$ and $\tilde{X}$ for source and target domain inputs. In each training batch, the first stage uses unpaired samples to train the two sub-networks of both streams. Note that the network parameters of the two streams are shared. One sub-network uses Log-Mel spectrogram $X_m$ and the other one uses deltas and delta-deltas information $X_d$. The training loss for the source domain is:

$$L_s = l(\phi_m(X_m), Y) + l(\phi_d(X_d), Y), \tag{6}$$

where $l(\cdot,\cdot)$ is the NLL loss and $Y$ is the label. The functions $\phi_m()$ and $\phi_d()$ are the convolutional layers of the mel and delta branches.

In the second stage, the networks are trained using the target domain classification loss and feature alignment loss jointly. It is worth noting that in the process of feature alignment, when the data of the source domain and the target domain pass through the network. There is no dimensionality reduction and subsequent operations. Still, the feature alignment is performed after the convolution block. This part of the network is represented by $\zeta_m()$ and $\zeta_d()$.

$$L_f = \varepsilon(\zeta_m(X_{s,m}), \zeta_m(X_{t,m})) + \varepsilon(\zeta_d(X_{s,d}), \zeta_d(X_{t,d})), \tag{7}$$

$$L_t = \xi(\phi_m(X_{t,m}), Y) + \xi(\phi_d(X_{t,d}), Y) + \lambda * L_f, \tag{8}$$

where $L_f$ is the feature alignment loss as defined in the last section between the source and target domain feature maps, $\varepsilon()$ is the feature alignment loss, $\lambda$ is the hyper-parameter for balancing the total loss function.

As the number of unpaired training samples is much larger than that of the paired training samples, we propose an alternate training strategy for effective network training. The samples of each batch are selected by using the same number of unpaired and paired samples without putting them back. When all the paired samples are selected, there are still many unpaired samples left. In this case, we restore the paired data and select them with the left unpaired samples until all the unpaired samples are selected. In each iteration, we calculate $L_s$ and $L_t$ for back propagation separately.

TABLE I: Effects of data augmentation and feature alignment

| System | Data Augmentation | Alignment loss | Acc.(%) |
|---|---|---|---|
| Baseline | NO / YES | NO | 54.1 / 56.2 |
| Two-Stage | NO / YES | MMD | 69.6 / 71.1 |
| Two-Stage | NO / YES | KL | 68.6 / 69.8 |
| Two-Stage | NO / YES | L1 | 70.8 / 71.2 |
| Two-Stage | NO / YES | MSE | 70.2 / 72.2 |

## III. Experimental Results

### A. Datasets and Experimental Settings

We evaluate the proposed method on the DCASE2019 Task1b [24] and DCASE2020 Task1a [25] benchmarks. DCASE2020 contains 15480 samples captured by 9 devices: $14400 \times 3$ samples of 3 real devices (A, B, C) and $1080 \times 6$ samples of 6 simulated devices (S1-S6). Note that the samples of S4-S6 do not appear in the training set. DCASE2019 contains 16560 segments, including 14400/1080/1080 samples from the devices A/B/C, respectively.

The audios are re-sampled to 32kHz. Then the Log-Mel spectrogram of each audio signal is extracted with the window size of 2048 (25% hop size). The number of frequency bands is assigned by 64. The Log-Mel features are normalized over each frequency bin such that the training data values have zero mean and are in the range of [-1, 1].

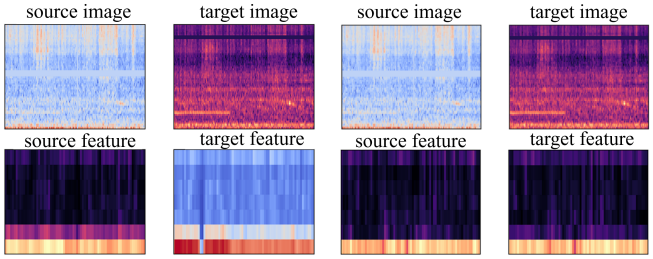
We implemented our method using PyTorch. We train our network for 200 epochs on one RTX 3090 card with the AdamW optimizer and the batch size of 64. The optimizer has the learning rate of 0.001, betas of (0.9, 0.999), eps of 1e-08, and weight decay of 0.01. The learning rate decreases for every 200 iterations exponentially with the rate of 0.1.

We applied data augmentation methods, including mix-up [26] (alpha: 0.5), SpecAugment [27] (frequency mask: 10; time mask: 10), and random crop (40 frames along the time dimension) to the proposed method.

### B. Effect of Domain Adaption

We use the model proposed in [28] as the baseline method and test different feature alignment loss functions. Meanwhile, the difference raised by not using domain adaption is also analyzed. As shown in Table I, unlike other domain adaption tasks, the MMD loss and KL loss achieve worse results. Instead, the use of MSE and L1 loss functions performs better. Obviously, by adding feature alignment loss for domain adaptation, the features obtained from the high-dimensional source domain can be effectively aligned with the target domain, as shown in Fig. 2.

We select three categories from sound sources of a variety of different devices, including large-sample data (Device A) as source domain, small-sample data (Device B) as target domain and the data that does not appear in the training set (Device S4) as unseen domain from the DCASE 2020 dataset. The t-SNE method is used for visualization. Fig. 4 shows the method using feature alignment (second row) outperforms those without feature alignment (first row). We can see that the method can better distinguish the features from different categories by performing domain adaptation with feature alignment.




(a) Without feature alignment (b) With feature alignment

Fig. 2: Visualization of feature alignment. The original Log-Mel spectrogram images and extracted features are very different without feature alignment. In contrast, the extracted features become very similar by applying feature alignment.

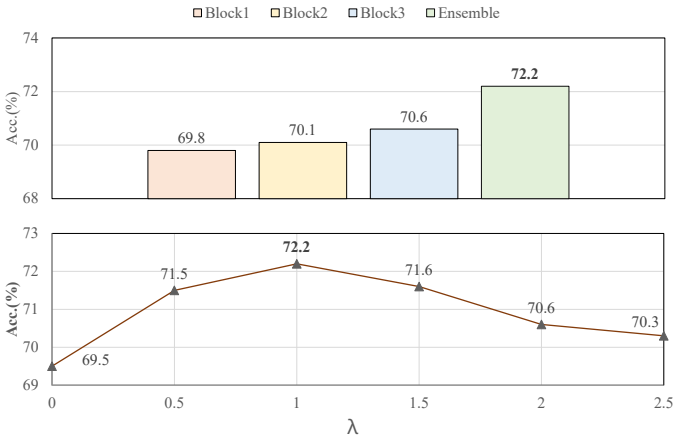
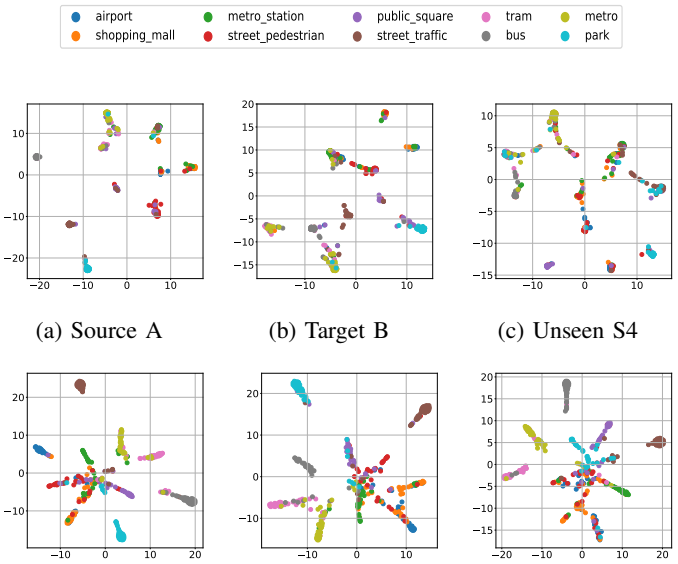



Fig. 3: The effect of feature alignment at different positions. The 'ensemble' one performs alignment at all the 3 positions.

## C. Ablation Study

We further conduct a series of ablation experiments to study the influence on domain adaptation by using feature alignment in different positions. Meanwhile, the effect of $\lambda$ is also investigated. As we know, CNNs can extract general features of an image in shallow layers. In contrast, we can obtain high-level abstract features from a deeper layer. We first fixed the value of $\lambda$ to 1. Fig. 3 indicates the effectiveness and robustness of the network as the degree of feature alignment deepens in different layers. With this model, we can achieve the best performance by using the three aligned convolution modules after feature extraction.

The parameter, $\lambda$, represents the ratio of the classification loss of the target domain to the feature alignment loss in the second round of training. In the ablation experiment, we use the MSE loss to perform feature alignment using the ensemble strategy that integrates the feature alignment results obtained from the previous convolution blocks. As shown in Fig. 3, the effectiveness of using feature alignment is better than that without feature alignment. The highest accuracy, 72.2%, is obtained when we set $\lambda$ to 1.




(a) Source A (b) Target B (c) Unseen S4

(d) Source A (e) Target B (f) Unseen S4

Fig. 4: The t-SNE plots of the models trained on DCASE2020 task 1-A.

TABLE II: A comparison with the SOTA methods.

| DCASE 2020 | | | DCASE 2019 | | |
|---|---|---|---|---|---|
| **Method** | **Ensemble** | **2020 Acc.(%)** | **Method** | **Ensemble** | **Acc.(%)** |
| Baseline | w/o | 51.6 | Baseline | w/o | 41.4 |
| CNNensem | 3 | 68.8 | hitsplab-2 | w/o | 41.4 |
| VilEnsemb3 | 4 | 70.3 | IITKGP-MFDWC19 | w/o | 52.3 |
| Cp-res | w/o | 71.8 | cvssp-cnn9 | w/o | 52.7 |
| UOS-totens | 3 | 71.9 | Rui-task1b | w/o | 54.8 |
| B3-all-mix | 16 | 71.9 | Randomforest-16 | 16 | 62.2 |
| **Ours** | **w/o** | **72.2** | UniSA-1b3 | w/o | 64.2 |
| Gao-UNISA | 16 | 72.5 | CPR-Ensemble | 8 | 65.1 |
| Liu-SHNU | 3 | 73.1 | **Ours** | w/o | **66.5** |
| Suh-ETRI | 3 | 74.2 | Kent | 2 | 72.9 |

## D. A Comparison with SOTA Methods

We compare the proposed method with the recently published state-of-the-art approaches on the DCASE2019 and 2020 datasets in Table II. In the table, many approaches use ensemble methods to improve their performance, resulting in high computational complexity. However, this strategy is not encouraged by DCASE since 2021. In contrast, the proposed method does not use any additional data or ensemble methods. It has much fewer model parameters as compared with the ensemble methods in the table. According to the table, the proposed method beats all the SOTA methods that do not use ensemble modeling. As compared with the ensemble methods, our method still achieves competitive (or even better) results.

# IV. Conclusion

In this letter, we proposed a novel method to solve the device mismatch problem in acoustic scene classification. We used a two-stream CNN to extract the features from the Log-Mel spectrogram and deltas dimensions. Different feature alignment losses were investigated for feature alignment. We also addressed the data imbalance problem via an alternative training strategy. As evaluated on the DCASE2019 and 2020 benchmarks, the proposed method achieves promising results as compared with the state of the art approaches.

## REFERENCES


[1] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," *World Congress on Services*, pp. 21–28, 2015.

[2] F. Dufaux and T. Ebrahimi, "Scrambling for Privacy Protection in Video Surveillance Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1168–1174, 2008.

[3] N. Jakovljević and T. Lončar-Turukalo, "Hidden markov model based respiratory sound classification," in *International Conference on Biomedical and Health Informatics*. Springer, 2017, pp. 39–43.

[4] J. T. Geiger, B. W. Schuller, and G. Rigoll, "Large-scale audio feature extraction and svm for acoustic scene classification," *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4, 2013.

[5] H. Phan, L. Hertel, M. Maaß, P. Koch, and A. Mertins, "Label tree embeddings for acoustic scene classification," *Proceedings of the 24th ACM international conference on Multimedia*, 2016.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] 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.

[8] O. Mariotti, M. Cord, and O. Schwander, "Exploring deep vision models for acoustic scene classification," *Proc. DCASE*, pp. 103–107, 2018.

[9] J. Huang, H. Lu, P. Lopez Meyer, H. Cordourier, and J. Del Hoyo Ontiveros, "Acoustic scene classification using deep learning-based ensemble averaging," 2019.

[10] L. Zhang, J. Han, and Z. Shi, "Learning temporal relations from semantic neighbors for acoustic scene classification," *IEEE Signal Processing Letters*, vol. 27, pp. 950–954, 2020.

[11] Z. Ren, Q. Kong, J. Han, M. D. Plumbley, and B. W. Schuller, "Attention-based atrous convolutional neural networks: Visualisation and understanding perspectives of acoustic scenes," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 56–60.

[12] X. Bai, J. Du, J. Pan, H.-s. Zhou, Y.-H. Tu, and C.-H. Lee, "High-resolution attention network with acoustic segment model for acoustic scene classification," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 656–660.

[13] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, *et al.*, "A two-stage approach to device-robust acoustic scene classification," *arXiv preprint arXiv:2011.01447*, 2020.

[14] H. Wang, Y. Zou, and D. Chong, "Acoustic scene classification with spectrogram processing strategies," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 210–214.

[15] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," *arXiv preprint arXiv:1707.01217*, 2017.

[16] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in neural information processing systems*, 2007, pp. 601–608.

[17] N. He and J. Zhu, "A weighted partial domain adaptation for acoustic scene classification and its application in fiber optic security system," *IEEE Access*, 2020.

[18] J. Wang, J. Chen, J. Lin, L. Sigal, and C. W. de Silva, "Discriminative feature alignment: Improvingtransferability of unsupervised domainadaptation by gaussian-guided latentalignment," *arXiv preprint arXiv:2006.12770*, 2020.

[19] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[20] R. Wang, M. Wang, X.-L. Zhang, and S. Rahardja, "Domain adaptation neural network for acoustic scene classification in mismatched conditions," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1501–1505.

[21] P. Primus, H. Eghbal-zadeh, D. Eitelsebner, K. Koutini, A. Arzt, and G. Widmer, "Exploiting parallel audio recordings to enforce device invariance in cnn-based acoustic scene classification," *arXiv preprint arXiv:1909.02869*, 2019.

[22] Y. Wu and T. Lee, "Time-frequency feature decomposition based on sound duration for acoustic scene classification," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 716–720.

[23] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[24] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in dcase 2019 challenge: Closed and open set classification and data mismatch setups," 2019.

[25] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," *arXiv preprint arXiv:2005.14623*, 2020.

[26] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[27] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[28] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: Dcase 2019 baseline systems," *arXiv preprint arXiv:1904.03476*, 2019.