ON THE IMPACT OF NORMALIZATION STRATEGIES IN UNSUPERVISED ADVERSARIAL DOMAIN ADAPTATION FOR ACOUSTIC SCENE CLASSIFICATION

Michel Olvera[1]*, Emmanuel Vincent*[1]*, Gilles Gasso*[2]

[1] Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
[2] LITIS EA 4108, Université & INSA Rouen Normandie, 76800 Saint-Étienne du Rouvray, France
michel.olvera@inria.fr

ABSTRACT

Acoustic scene classification systems face performance degradation when trained and tested on data recorded by different devices. Unsupervised domain adaptation methods have been studied to reduce the impact of this mismatch. While they do not assume the availability of labels at test time, they often exploit parallel data recorded by both devices, and thus are not fully blind to the target domain. In this paper, we address a more practical scenario where parallel data are not available. We thoroughly analyze the impact of normalization and moment matching strategies to compensate for the linear distortion introduced by the recording device and propose their integration with adversarial domain adaptation to handle the remaining non-linear distortion. Experiments on the DCASE Challenge 2018 Task 1B dataset show that the proposed integrated approach considerably reduces domain mismatch, reaching an accuracy in the target domain close to that obtained in the source domain.

Index Terms— Acoustic scene classification, adversarial domain adaptation, feature normalization, moment matching

1. INTRODUCTION

Acoustic Scene Classification (ASC) consists of identifying the acoustic environment in which an audio signal was captured [1]. The growing interest for ASC in recent years has led it to be the core task of acoustic monitoring applications.

When the acoustic conditions at test time differ from those considered during model training, ASC systems may exhibit performance degradation due to a shift between the data distributions. A well-studied cause of mismatch is the use of different data acquisition hardware, which prevents generalization to data captured with unseen recording devices.

The ASC task with mismatched recording devices has been widely popularized by the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge Task 1 series [2–4], which provides a dataset with a large number of recordings from a source device but only a limited amount of data from target devices. The primary goal has been to improve generalization on the underrepresented devices. Supervised machine learning algorithms have been proposed to account for the data imbalance problem and are often combined with data augmentation, regularization and fine tuning approaches [5–7]. As the dataset contains recordings captured simultaneously by the source and target devices, some methods leverage these parallel data to compensate for the effects of the frequency responses of the devices [8–10].

A few works have also used this dataset to investigate the more practical scenario where recordings of the source and target devices are available, but only the source recordings are labeled. To improve generalization on the target devices, they have applied unsupervised domain adaptation (UDA) methods [11, 12]. In particular, adversarial domain adaptation [13] has proven to be effective [14, 15]. Despite its effectiveness, it requires a large number of recordings from the target device to carry out the adaptation process. Furthermore, the adaptation process could implicitly benefit from the parallel data present in the dataset, or explicitly use these data to ease generalization [16, 17]. To overcome these limitations, band-wise statistical matching (BSWM) was introduced as a simple, linear UDA method for ASC that does not require any adaptation stage [18, 19]. The integration of this method with non-linear, learning-based UDA methods has not been explored.

In this paper, we thoroughly analyze the impact of various feature normalization and moment matching strategies to compensate for the domain shift due to mismatched recording devices (among which BWSM is a particular case), without assuming the availability of parallel source and target device data. Using the development set of the DCASE Challenge 2018 Task 1B, we show experimentally the individual scopes and limitations of such techniques, as well as their integration with adversarial domain adaptation strategies to further improve generalization in the target domain.

2. PROBLEM FORMULATION

2.1. Linear distortion model

Let us denote by x_{nmk} the log-magnitude short-time Fourier transform (STFT) coefficients in time frame m and frequency bin k of the actual (undistorted) signal of some acoustic scene indexed by n, and by x_{dnmk} the log-magnitude STFT coefficients of the same signal captured by some recording device d with time-invariant linear magnitude frequency response h_{dk}. Assuming that there is no other distortion, the captured acoustic scene can be expressed as $x_{dnmk} = x_{nmk} + \log h_{dk}$. The undistorted signal x_{nmk} can therefore be recovered as

$$\hat{x}_{dnmk} = x_{dnmk} - \log h_{dk}. \tag{1}$$

We keep index d in the notation $\hat{x}_{dnmk}$ to emphasize the fact that this estimate was obtained from device d. In practice h_{dk} is unknown, hence $\hat{x}_{dnmk}$ must be obtained from x_{dnmk} only.

This work was made with the support of the French National Research Agency, in the framework of the project LEAUDS "Learning to understand audio scenes" (ANR-18-CE23-0020). Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (`https://www.grid5000.fr`).

2.2. Moment normalization

Moment normalization consists of applying a domain-dependent linear transform to the data in the source and target domains so that their first- and second-order moments are fixed. Mean normalization is the simplest form of this technique. It consists of subtracting the average log-magnitude spectrum

$$\mu_{dk} = \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} x_{dnmk} \tag{2}$$

of the data recorded by each device d from the original data:

$$\hat{x}^{\mathrm{MN}}_{dnmk} = x_{dnmk} - \mu_{dk}. \tag{3}$$

This is equivalent to (1), where μ_{dk} can be seen as an estimate of $\log h_{dk}$ up to an arbitrary frequency response which is common to all devices.

Mean and variance normalization (a.k.a. standardization) additionally requires the computation of the sample standard deviation

$$\sigma_{dk} = \sqrt{\frac{1}{NM-1} \sum_{n=1}^{N} \sum_{m=1}^{M} (x_{dnmk} - \mu_{dk})^2} \tag{4}$$

of the data for each device d. Normalization is achieved by

$$\hat{x}^{\mathrm{MVN}}_{dnmk} = \frac{x_{dnmk} - \mu_{dk}}{\sigma_{dk}}. \tag{5}$$

Although not addressing any specific physical distortion, this is a common preprocessing step in machine learning [20,21].

2.3. Moment matching

Unlike moment normalization, moment matching does not eliminate the distortion due to the recording device d from the source data. Instead, it transforms the target data recorded by some other device d' so that its first- and second-order moments match those of the source data.

Matching the means of x_{dnmk} and $x_{d'nmk}$ can be achieved by removing from $x_{d'nmk}$ the distortion due to device d' as in (3), and then introducing the distortion due to device d:

$$\hat{x}^{\mathrm{MM}}_{d'nmk} = x_{d'nmk} - \mu_{d'k} + \mu_{dk}. \tag{6}$$

After moment matching, x_{dnmk} and $\hat{x}^{\mathrm{MM}}_{d'nmk}$ share the same sample mean, which according to (1) should suffice to transfer the distortion from one device to another. In practice, further robustness may be obtained by also matching the variances. This is achieved by first normalizing the mean and variance of $x_{d'nmk}$ as in (5), and then scaling and shifting the standardized frequency bands by σ_{dk} and μ_{dk}:

$$\hat{x}^{\mathrm{MVM}}_{d'nmk} = \frac{(x_{d'nmk} - \mu_{d'k})\sigma_{dk}}{\sigma_{d'k}} + \mu_{dk}. \tag{7}$$

2.4. Adversarial domain adaptation (ADA)

The four linear moment normalization or moment matching techniques above improve robustness to linear filtering of the acoustic scene, however they fail to compensate for non-linear mismatches, e.g., reverberation or phase distortion.

To mitigate these non-linear mismatches, we follow the unsupervised domain adaptation method proposed for ASC in [15]. The general framework is a two-step adversarial domain adaptation process based on the Wasserstein generative adversarial networks (WGAN) formulation [22]. It relies on three deep neural network-based models: a feature extractor g, a classifier f which outputs the vector of posterior probabilities of all ASC classes, and a discriminator h which outputs the posterior probability that the input data is from the target (as opposed to the source) domain. We regard as source domain data $\mathbf{X}^{\mathrm{s}} = \{x_{dnmk}\}_{n=1}^{N_{\mathrm{s}}}$ the acoustic scenes from device d, with one-hot labels $\mathbf{y}^{\mathrm{s}}$ of the considered classes. We regard as target domain data $\mathbf{X}^{\mathrm{t}} = \{x_{d'nmk}\}_{n=1}^{N_{\mathrm{t}}}$ the acoustic scenes recorded by some other device d', without class labels. Starting from a pre-trained feature extractor g^* and a classifier f trained on source data, the goal is to regularize the feature extractor g using the discriminator h so that it produces features $g(\mathbf{X}^{\mathrm{s}})$ and $g(\mathbf{X}^{\mathrm{t}})$ which exhibit the same distribution across domains.

Pretraining: In the first step, we obtain the pre-trained feature extractor g^* and label classifier f from the source domain data by minimizing

$$\mathcal{L}_s = -\sum_{n=1}^{N_{\mathrm{s}}} \mathbf{y}^{\mathrm{s}}_n \bullet \log(f(g^*(\mathbf{X}^{\mathrm{s}}_n)) \tag{8}$$

where $\bullet$ denotes the dot product.

Adaptation: In the second step, the feature extractor g is initialized as the pretrained model g^*, and g and h are jointly optimized on source and target domain data by minimizing

$$\mathcal{L}_h = \sum_{n=1}^{N_{\mathrm{s}}} h(g^*(\mathbf{X}^{\mathrm{s}}_n)) - \sum_{n=1}^{N_{\mathrm{t}}} h(g(\mathbf{X}^{\mathrm{t}}_n)) \tag{9}$$

$$\mathcal{L}_g = \sum_{n=1}^{N_{\mathrm{t}}} h(g(\mathbf{X}^{\mathrm{t}}_n)) - \sum_{n=1}^{N_{\mathrm{s}}} \mathbf{y}^{\mathrm{s}}_n \bullet \log(f(g(\mathbf{X}^{\mathrm{s}}_n)) \tag{10}$$

to enforce domain-invariant distributions. The second term in (10) is a classification loss that prevents g from losing performance on the source domain data. Following [15], $\mathcal{L}_h$ and $\mathcal{L}_g$ are iteratively minimized by updating h according to the gradient of (9) w.r.t. h with g fixed, and updating g according to the gradient of (10) w.r.t. g with h fixed.

Inference: After adaptation, g and f are used to classify acoustic scenes from both source and target devices.

2.5. Conditional adversarial domain adaptation (CADA)

The above adversarial domain adaptation strategy aligns the marginal distributions of the source- and target-domain features, but not their class-conditional distributions. Following [23], an alternative adversarial domain adaptation formulation that enforces the joint distribution alignment of features and ASC classes is to condition the domain discriminator h on the class-posteriors from f with the joint variable $w(\mathbf{X}) = g(\mathbf{X}) \otimes f(g(\mathbf{X}))$ which aims to capture the multimodal information of g and f. Introducing the multilinear mapping through $w(\mathbf{X})$, the losses in (9) and (10) become

$$\mathcal{L}_h = \sum_{n=1}^{N_{\mathrm{s}}} h(w^*(\mathbf{X}^{\mathrm{s}}_n)) - \sum_{n=1}^{N_{\mathrm{t}}} h(w(\mathbf{X}^{\mathrm{t}}_n)) \tag{11}$$

$$\mathcal{L}_g = \sum_{n=1}^{N_{\mathrm{t}}} h(w(\mathbf{X}^{\mathrm{t}}_n)) - \sum_{n=1}^{N_{\mathrm{s}}} \mathbf{y}^{\mathrm{s}}_n \bullet \log(f(g(\mathbf{X}^{\mathrm{s}}_n)). \tag{12}$$

3. EVALUATION SETUP

3.1. Dataset

In order to assess the impact of moment normalization and moment matching, as well as their integration with adversarial domain adaptation, we perform experiments on the development dataset of the DCASE Challenge 2018 Task 1B. The dataset comprises 10 s acoustic scenes recorded in six European cities using three different recording devices, namely devices A, B and C. From each acoustic scene, we extracted 64-dimensional log-Mel spectra, using a Hamming window of 2048 samples (46 ms) and a hop size of 1024 samples (23 ms), leading to an overlap of 50% across frames. Each acoustic scene is categorized by one of the following labels: *airport*, *bus*, *metro*, *metro station*, *park*, *public square*, *shopping mall*, *street pedestrian*, *street traffic*, and *tram*. The dataset contains a total of 28 hours of audio out of which 24 hours are from device A, 2 hours from device B, and 2 hours from device C. We follow the same setup as in [14, 15, 19], except that we discard the subset of recordings from device A which are parallel to recordings from devices B and C. In the original setup, 8.8% of parallel data is randomly distributed in the training and validation sets of device A. These parallel recordings raise two issues in the context of UDA: first, in the adaptation step, the model could discriminate distortions between parallel recordings more easily; second, in the inference step, the parallel recordings from the target devices could be considered by the model as transformed examples of acoustic scenes already seen during training, thus making their classification easier. All previous works on UDA for ASC using this setup suffer from these issues.[1]

To address UDA in the fully blind setting where a pretrained ASC system must be deployed on devices with unknown microphone responses, the assumption of the availability of parallel recordings is not realistic. Accordingly, our setup without parallel recordings comprises 5,024 training audio scenes from the source device A, and 486 for each target device B and C. The validation set comprises 558 acoustic scenes from the source device A, and 54 acoustic scenes from each target device B and C. The test set is composed of 2,518 acoustic scenes from device A, and 180 acoustic scenes from each target device B and C.

3.2. Model and training

We employ the model architecture referred to as “Kaggle” in [14, 15, 19]. The feature extractor g consists of five convolutional neural network (CNN) layers, with square kernel shapes of widths 11, 5, 3, 3, 3, and 48, 128, 192, 192, 128 channels. The stride is (2, 3) for the first two layers and (1, 1) for the rest. All layers are followed by rectified linear unit (ReLU) activation, and the first two and last layers use batch normalization and max pooling, with square kernels of width 2 and a stride of (1,2), (2,2),(1,2). The label classifier f consists of two linear layers with ReLU activations followed by a linear layer with softmax activation. The domain discriminator h consists of a linear layer with ReLU activation followed by a linear layer without activation. The RMSProp optimizer is used with a learning rate of 5×10^{-5}. We use a batch size of 16 and the feature classifier g was trained for 300 epochs.

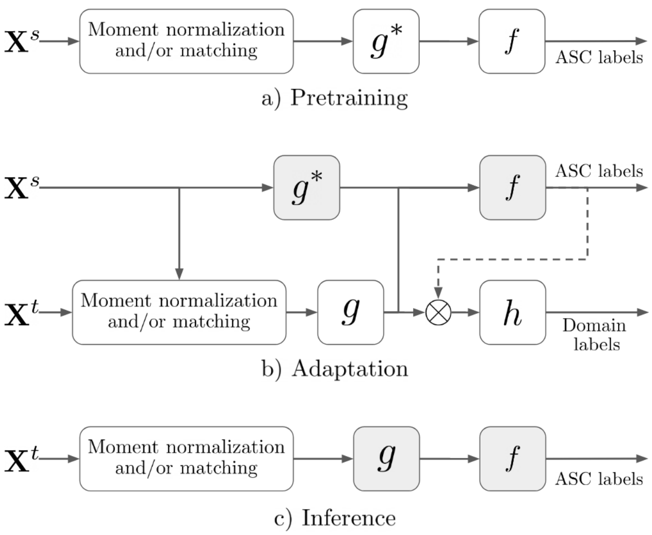

Fig. 1. Proposed integration of moment normalization and/or matching with adversarial domain adaptation methods. a) Pretraining. b) Adaptation: the dashed line allows conditional adversarial domain adaptation (CADA). If removed, the strategy corresponds to adversarial domain adaptation (ADA). c) Inference. Gray boxes indicate the elements which are not optimized in the corresponding step.

3.3. Experiments

We carry out experiments to analyze the impact of moment normalization and moment matching used alone or in combination with adversarial domain adaptation (ADA) or conditional adversarial domain adaptation (CADA). More specifically, we transform the source device data by mean normalization (MN) or mean and variance normalization (MVN) in the pretraining step. At inference time, we transform the target device data by the same MN or MVN strategy when it was applied in the pretraining step, or by mean matching (MM) or mean and variance matching (MVM) when no normalization was applied in the pretraining step. In addition, a hybrid mean normalization and variance matching (MNVM) strategy is also tested, where mean normalization is applied to the source and target data and the variances are subsequently matched. In the adaptation step, the above strategies are used to transform the target data prior to ADA or CADA. We also evaluate ADA and CADA alone for comparison.[2] Figure 1 illustrates the proposed integration of moment normalization and/or matching with adversarial domain adaptation methods at each of these three steps.

For ADA/CADA, we consider devices B and C as one domain, because the amount of training data from each device is small [15]. By contrast, we perform moment normalization and matching in two settings: *device-independent*, in which we regard devices B and C as one domain and transform the data using the average sample statistics of the two devices, and *device-dependent*, in which we regard B and C as two distinct domains and transform the data using their respective statistics. When no moment normalization/matching is performed, the system is categorized as device-independent.

[1] In practice, experiments with parallel data (not shown here) resulted in a higher mean accuracy for 72 out of the 90 results reported in Table 1, out of which 8 were statistically significant.

[2] Applying MVM alone at inference time is equivalent to BWSM in [19]. The results differ from [15] and [19] due to discarding parallel data and not standardizing the data using the average statistics of devices A, B and C.

Table 1. Average ASC accuracy (%) and standard deviation (in parentheses) on the test set achieved over 20 training runs. Bold numbers show the best statistically significant (p-value < 0.05) results in the target domain.

	Method		Device indep.		Device dep.	
Pretrain.	Adapt.	Infer.	source	target	source	target
-	-	-	59.3(5.1)	13.6(2.3)	N/S	N/S
-	ADA [15]	-	62.3(2.0)	36.2(2.6)	N/S	N/S
-	CADA	-	61.0(2.7)	39.4(2.7)	N/S	N/S
MN	-	MN	62.5(2.0)	43.5(2.3)	62.1(2.1)	51.1(2.4)
MN	MN-ADA	MN	64.7(1.0)	50.8(1.6)	64.0(2.8)	58.5(1.2)
MN	MN-CADA	MN	64.5(1.1)	51.6(1.3)	64.2(1.3)	59.0(1.6)
MVN	-	MVN	62.3(1.9)	41.8(1.0)	62.5(2.1)	51.0(2.0)
MVN	MVN-ADA	MVN	64.8(1.4)	**52.0(1.2)**	64.7(1.3)	**59.3(1.4)**
MVN	MVN-CADA	MVN	64.3(1.3)	**52.7(1.6)**	65.0(1.4)	**60.0(1.2)**
-	-	MM	59.3(5.1)	37.3(5.5)	59.3(5.1)	50.0(3.5)
-	ADA	MM	62.3(2.0)	41.3(2.0)	62.3(2.0)	52.0(2.7)
-	MM-ADA	MM	62.1(2.8)	47.0(2.3)	62.2(2.0)	56.2(1.8)
-	CADA	MM	61.0(6.5)	40.3(2.0)	61.0(6.5)	51.9(2.6)
-	MM-CADA	MM	62.5(1.5)	48.9(2.2)	62.7(1.2)	57.7(1.7)
-	-	MVM [19]	59.3(5.1)	38.8(2.9)	59.3(5.1)	50.1(3.5)
-	ADA	MVM	62.3(2.0)	38.2(2.2)	62.3(2.0)	51.0(2.3)
-	MVM-ADA	MVM	63.7(1.1)	47.4(1.9)	62.6(1.5)	56.5(1.7)
-	CADA	MVM	62.4(2.8)	38.1(1.8)	62.4(2.8)	51.2(2.1)
-	MVM-CADA	MVM	63.0(1.3)	50.4(1.2)	63.2(1.6)	58.5(2.0)
MN	-	MNVM	64.2(2.4)	42.3(4.3)	63.1(1.8)	51.8(2.2)
MN	MN-ADA	MNVM	64.7(1.0)	48.6(1.9)	64.0(2.22)	56.0(1.7)
MN	MN-CADA	MNVM	64.0(1.6)	49.4(1.2)	64.0(1.6)	57.2(1.3)
MN	MNVM-ADA	MNVM	64.2(1.8)	50.6(1.1)	64.2(1.0)	**59.9(1.6)**
MN	MNVM-CADA	MNVM	64.0(1.9)	51.7(1.1)	64.6(1.8)	**60.1(1.8)**

4. RESULTS AND DISCUSSION

The results are reported in Table 1. The accuracy of the system pretrained on unnormalized data reaches 59.3% on unnormalized source domain data, but drops to 13.6% on unnormalized target domain data (1st row).

Adversarial domain adaptation methods with unnormalized data (2nd and 3rd rows) boost accuracy by up to 26% absolute in the target domain and by up to 3% absolute in the source domain. CADA obtains significantly higher average accuracies than ADA (p-value < 0.05) for the conditions in the 3rd, 14th, 19th and 24th rows in the device independent setting, and for those in the 14th, 19th and 22nd rows in the device-dependent setting.

Adapting the system pretrained on unnormalized data through moment matching (MM or MVM) increases the accuracy in the target domain by up to 25% absolute in the device-independent setting and by 36% absolute in the device-dependent setting (10th and 15th rows). These results show that moment matching effectively transfers distortions from one device to another. In both settings, MM and MVM are not statistically different.

Pretraining the system on normalized data (MN or MVN) and applying the same normalization during inference is sufficient to largely correct the mismatch between the source and target distributions by removing linear distortions introduced by the recording device. The system's performance in the target domain increases by up to 30% absolute in the device-independent setting and by 37% absolute in the device-dependent setting (4th and 7th rows). In the former setting MN outperforms MVN, while in the latter normalizing by MN or MVN is similarly effective.

By integrating moment normalization or moment matching with adversarial methods, the gap between the source and target domains is further reduced. The best accuracy achieved in the target domain is 53% in the device-independent setting and 60% in the device-dependent one. Such large mismatch correction is obtained by standardizing the source and target data during adaptation regardless of the adaptation method (MVN-ADA or MVN-CADA, 8th and 9th rows). An equally good performance is obtained by normalizing the means and matching the variances of the data during adaptation (MNVM-ADA or MNVM-CADA, 23rd and 24th rows) in the device-dependent setting. This shows that second-order moment normalization or matching help further improve the performance in the target domain compared to methods that use first-order statistics only.

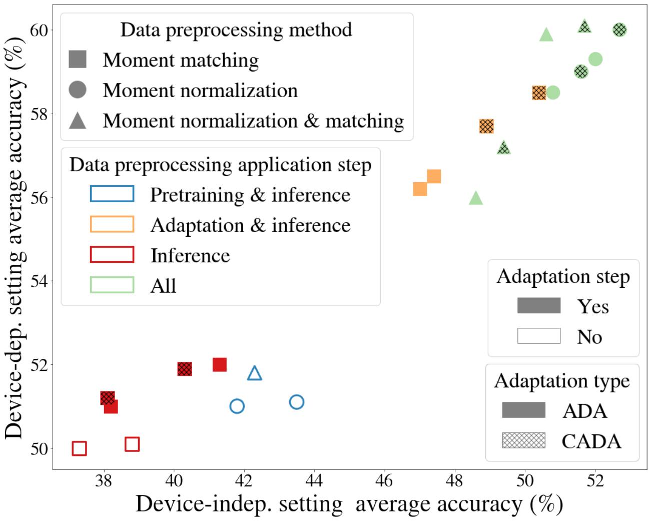

Fig. 2. Target domain average accuracy in the device-independent setting and corresponding average accuracy in the device-dependent setting for the adaptation strategies in Table 1. Best viewed in color.

Figure 2 shows the average accuracy in the target domain obtained by the adaptation strategies in Table 1. For each strategy, its accuracies in the device-independent and device-dependent settings are shown. Adaptation strategies combining moment normalization and/or matching with adversarial domain adaptation perform better than those lacking adversarial domain adaptation or that do not combine moment normalization and/or matching with adversarial domain adaptation. Among the best performing adaptation strategies, those that apply moment normalization and/or matching in all steps tend to outperform moment matching strategies applied in the adaptation and inference steps.

5. CONCLUSION

We experimentally assessed the impact of moment normalization and moment matching strategies as well as their integration with adversarial domain adaptation methods for acoustic scene classification. We showed that normalization strategies are particular instances of a linear distortion model that improve robustness to mismatched recording devices. The combination of moment normalization and/or matching strategies with adversarial domain adaptation methods reduces remaining mismatch due to non-linear effects. Results indicate that such integration achieves a performance in the target domain close to that obtained in the source domain.

6. REFERENCES

[1] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[2] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Acoustic scene classification: an overview of DCASE 2017 Challenge entries," in *Proc. IWAENC*, 2018, pp. 411–415.

[3] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Acoustic scene classification in DCASE 2019 Challenge: closed and open set classification and data mismatch setups," in *Proc. DCASE*, 2019, pp. 164–168.

[4] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. DCASE*, 2018, pp. 9–13.

[5] Truc Nguyen and Franz Pernkopf, "Acoustic scene classification with mismatched devices using cliquenets and mixup data augmentation," in *Proc. Interspeech*, 2019, pp. 2330–2334.

[6] Hu Hu, Chao-Han Huck Yang, Xianjun Xia, Xue Bai, Xin Tang, Yajian Wang, Shutong Niu, Li Chai, Juanjuan Li, Hongning Zhu, Feng Bao, Yuanjun Zhao, Sabato Marco Siniscalchi, Yannan Wang, Jun Du, and Chin-Hui Lee, "A two-stage approach to device-robust acoustic scene classification," in *Proc. ICASSP*, 2021, pp. 845–849.

[7] Saori Takeyama, Tatsuya Komatsu, Koichi Miyazaki, Masahito Togami, and Shunsuke Ono, "Robust acoustic scene classification to multiple devices using maximum classifier discrepancy and knowledge distillation," in *Proc. EUSIPCO*, 2021, pp. 36–40.

[8] Michał Kośmider, "Spectrum correction: acoustic scene classification with mismatched recording devices," in *Proc. Interspeech*, 2020, pp. 4641–4645.

[9] Truc Nguyen, Franz Pernkopf, and Michał Kośmider, "Acoustic scene classification for mismatched recording devices using heated-up softmax and spectrum correction," in *Proc. ICASSP*, 2020, pp. 126–130.

[10] Paul Primus, Hamid Eghbal-zadeh, David Eitelsebner, Khaled Koutini, Andreas Arzt, and Gerhard Widmer, "Exploiting parallel audio recordings to enforce device invariance in CNN-based acoustic scene classification," in *Proc. DCASE*, 2019, pp. 204–208.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.

[12] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, 2017.

[13] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *Proc. CVPR*, 2017, pp. 7167–7176.

[14] Shayan Gharib, Konstantinos Drossos, Emre Cakir, Dmitriy Serdyuk, and Tuomas Virtanen, "Unsupervised adversarial domain adaptation for acoustic scene classification," in *Proc DCASE*, 2018, pp. 138–142.

[15] Konstantinos Drossos, Paul Magron, and Tuomas Virtanen, "Unsupervised adversarial domain adaptation based on the Wasserstein distance for acoustic scene classification," in *Proc. WASPAA*, 2019, pp. 259–263.

[16] Seongkyu Mun and Suwon Shon, "Domain mismatch robust acoustic scene classification using channel information conversion," in *Proc. ICASSP*, 2019, pp. 845–849.

[17] Dongchao Yang, Helin Wang, and Yuexian Zou, "Unsupervised multi-target domain adaptation for acoustic scene classification," in *Proc. Interspeech*, 2021, pp. 1159–1163.

[18] Alessandro Ilic Mezza, Emanuël A. P. Habets, Meinard Müller, and Augusto Sarti, "Feature projection-based unsupervised domain adaptation for acoustic scene classification," in *Proc. MLSP*, 2020, pp. 1–6.

[19] Alessandro Ilic Mezza, Emanuël A. P. Habets, Meinard Müller, and Augusto Sarti, "Unsupervised domain adaptation for acoustic scene classification using band-wise statistics matching," in *Proc. EUSIPCO*, 2021, pp. 11–15.

[20] Jerome H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2017.

[21] Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[22] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein generative adversarial networks," in *Proc. ICML*, 2017, pp. 214–223.

[23] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan, "Conditional adversarial domain adaptation," in *Proc. NIPS*, 2018, pp. 1647–1657.