INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Personalized Acoustic Scene Classification in Ultra-low Power Embedded Devices Using Privacy-preserving Data Augmentation

Timm Koppelmann, Semih Ağcaer, and Rainer Martin

Institute of Communication Acoustics, Ruhr-Universität Bochum, Bochum, Germany
{timm.koppelmann,semih.agcaer,rainer.martin}@rub.de

Abstract

In this work we present an adaptation method for personalized acoustic scene classification in ultra-low power embedded devices (EDs). The computational limitation of EDs and a large variety of acoustic scenes may lead to poor performance of the embedded classifier in specific real-world user environments. We propose a semi-supervised scheme that estimates the audio feature distribution at ED level and then samples this statistical model to generate artificial data points which emulate user-specific audio features. Then, a second, cloud-based classifier assigns pseudo labels to samples, which are merged with existing labeled data for retraining the embedded classifier. The proposed method leads to significant performance improvements on user-specific data sets and does neither require a persistent connection to a cloud service nor the transmission of raw audio or audio features. It thus results in low data rates, high utility, and privacy-preservation.

Index Terms: acoustic scene classification, embedded devices, semi-supervised learning, data augmentation

1. Introduction

Data-driven and machine-learning (ML) based audio classification systems are widely used in stationary and mobile embedded devices (EDs) like smart-home assistants or smartphones. These classifiers assign audio recordings to certain, predefined classes related to acoustic scenes and environments [1, 2, 3]. They have to cope with a large variety of acoustic environments which range, for instance, from living room activities (listening to radio, watching TV), to driving to work (car, street, train), to sports (jogging outside in the park), and many more. The focus of this work is on *acoustic scene classification* (ASC) in *ultra-low power* EDs such as hearing aids or cochlear implants. Here, scarce computational resources have to be shared between many (real-time) tasks, limiting the computational complexity available for ASC to low-dimensional feature vectors and less than 10^3 parameters in the embedded classifier.

In this scenario, the vast amount of real-life acoustic conditions leads to potentially poor classification performance. As shown in Fig 1, the intersection of a large and diverse training set and frequently-seen user-specific acoustic environments can be small. Since the embedded classifier will not perform at its best when it is overloaded with a large variety of acoustic conditions, it is beneficial to use a reduced training set which just contains the most relevant acoustic data of typical ED users. However, when the training data is tailored to typical ED users, it works well in matched conditions but may degrade severely in unknown acoustic environments.

The main goal of the proposed method is to adapt a baseline classification system to personal user conditions, while adhering to the following constraints: (1) very low data rates and no persistent connection from the ED to a back-end device, (2) no user feedback on the performance of the classifier in the ED, and (3) no transmission of raw audio or high-resolution audio features. Constraints (1) and (2) result in a high utility of the proposed method and limit the power consumption of the ED while (3) preserves the user's privacy.

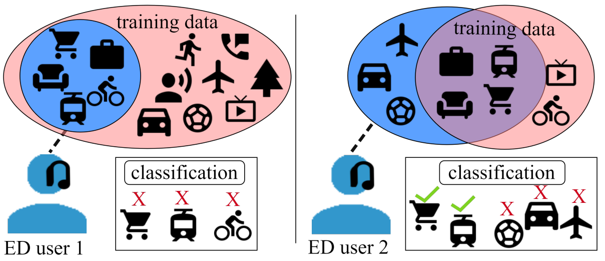


Figure 1: *Left: When the intersection of the training data (red) and the acoustic scenes of an ED user (blue) is rather small, it may lead to sub-optimal classification performance ('X') in some cases. Right: As a consequence, a reduced baseline set is used for training. Now, the acoustic scenes of a second user, which are not covered by the baseline set, are detected poorly.*

The proposed adaptation scheme improves the performance in user-specific scenes via data aggregation in the ED, data augmentation, and re-training in a back-end device (a.k.a. "cloud"). In particular, we aggregate audio features at ED level and formulate a Gaussian mixture model (GMM) that represents the underlying feature distribution related to the specific user. Then, at certain points in time (e.g., once every night), the acquired statistical parameters are transmitted to the more powerful back-end device. Here, the received GMM is sampled to generate artificial data points, which emulate user-specific audio features. Therefore, the back-end device also uses a second powerful DNN classifier to assign pseudo labels to the generated samples. After adding these samples to the baseline data, the embedded classifier is then retrained and thus provides improved personalization of the device. In what follows, we describe the components of this approach and their interactions in detail and analyze its performance w.r.t. the accuracy of the statistical model.

2. Related work

The proposed procedure is closely related to *semi-supervised* learning (SSL), where usually a larger amount of unlabeled and a limited amount of labeled data are jointly used in ML tasks [4]. SSL is applied in many applications such as image classification [5, 6] or text classification [7, 8], and is also

10.21437/Interspeech.2023-932

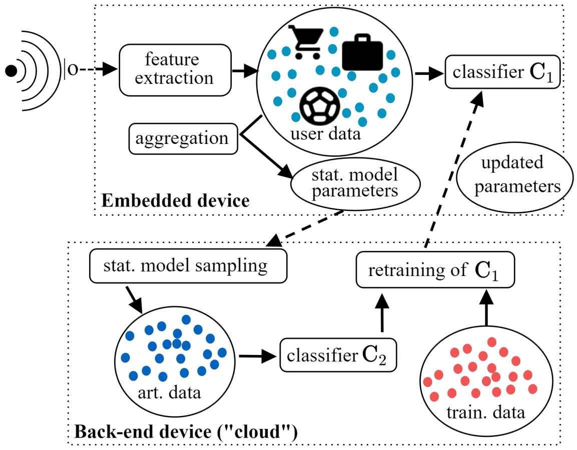


Figure 2: *Overview of the core elements in our SSL training scheme. The ED-based classifier is retrained on a combination of labeled training data and pseudo-labeled artificial samples. These samples are generated based on statistical moments and represent the acoustic environments of the ED user.*

increasingly applied to ASC [9, 10]. Mainly, SSL methods which are directly comparable to our work are *wrapper methods*, where one or multiple classifiers assign *pseudo labels* to unlabeled data, which can then be used for supervised training [11]. Two well-known methods are *self-training* [12, 13], using only one classifier that is pre-trained with labeled data and then retrained after generating pseudo-labeled data, and *co-training* [14, 15], where two classifiers are iteratively trained via an exchange of pseudo-labeled data. Furthermore, there are methods where un- or semi-supervised clustering techniques are used to assist the training process [16, 17].

While some parts of the proposed method are related to known SSL methods, the main novelties of our work are (1) the local aggregation of low-dimensional audio features which results in a statistical model describing the user's acoustic environment and (2) the use of this statistical model to generate artificial audio features which are then labeled and used for retraining the classifier in the ED. To the best of our knowledge, no other works have considered such a combination of statistical data acquisition, data augmentation, and SSL. In addition, we demonstrate that a statistical model of moderate complexity is sufficient to achieve notable improvements.

Note that low-complexity ASC solutions were also investigated within the DCASE [18, 19]. However, the footprint of DCASE solutions is at least one magnitude larger than with our approach. Therefore, ASC in assistive hearing devices usually discriminates only between a few major signal classes [20].

3. Method description

The main goal of the proposed method is to improve the user-specific ASC performance of a computationally constrained embedded classifier. We consider the framework in Fig. 2, which mainly consists of the embedded classifier $\mathbf{C}_1$ and an assisting powerful, DNN-based classifier $\mathbf{C}_2$ that could, for instance, be implemented in a cloud-based server application or a mobile device. More information on the structure of the classifiers and the related training is provided in Sec. 4.3, and a detailed description of the proposed framework is given below.

3.1. Feature extraction and aggregation in the ED

We employ amplitude-modulation-spectrum (AMS) based low-level audio features [21], which had been specifically designed for ultra-low power devices. Efficient implementation of these features is achieved through the use of 2 banks of recursive filters with optimized filter bandwidths. We extract one nine-dimensional AMS feature vector $\boldsymbol{z}$ from two seconds of audio and concatenate 5 consecutive feature vectors into a 45-dimensional input vector $\boldsymbol{x}$ for our classifiers, thus performing one classification per 10 s. The set of audio feature sequences $\boldsymbol{x}$ in the ED of a specific user is then denoted as user data $\mathcal{U}$:

$$\mathcal{U} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots \boldsymbol{x}_M\}, \tag{1}$$

with

$$\boldsymbol{x}_t = \left(\boldsymbol{z}_{t1}^{\mathrm{T}}, \boldsymbol{z}_{t2}^{\mathrm{T}}, \boldsymbol{z}_{t3}^{\mathrm{T}}, \boldsymbol{z}_{t4}^{\mathrm{T}}, \boldsymbol{z}_{t5}^{\mathrm{T}}\right)^{\mathrm{T}}. \tag{2}$$

The number of sequences M in $\mathcal{U}$ is not fixed but rather related to the user-specific time span between initialization of the ED (”baseline”) and the aspired re-configuration of $\mathbf{C}_1$, which could, e.g., cover one day ($M = 8640$) or one week ($M = 60480$). As we do not like to transmit audio features $\boldsymbol{z}$ directly for further processing, we aggregate the sequences $\boldsymbol{x}$ in $\mathcal{U}$ and derive statistical information about the underlying data distribution. However, in this first proof-of-concept work, we aim to study the required accuracy of the statistical model and therefore employ a GMM at ED level. Thus, we use the whole set $\mathcal{U}$ which, in principle, could be stored in the ED's memory since the feature vectors $\boldsymbol{z}$ are low-dimensional. Other, less complex procedures are possible as well and will be discussed in Section 6. The GMM is defined as

$$p(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \mathcal{N}\left(\boldsymbol{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right), \tag{3}$$

with mean vectors $\boldsymbol{\mu}_i$, covariance matrices $\boldsymbol{\Sigma}_i$, and weights α_i. Depending on the number of components N, the GMM is expected to provide a more or less accurate and versatile statistical representation of $\mathcal{U}$. Due to the mentioned restrictions, we only consider diagonal covariance matrices in this work.

3.2. Data augmentation and classifier re-configuration

As indicated earlier, our method includes a back-end application that assists the ED and provides updates for classifier $\mathbf{C}_1$. After receiving the model parameters that have been computed at ED-level, the GMM is statistically sampled in order to generate A artificial data points

$$\mathcal{U}_{\text{art}} = \{\boldsymbol{x}_1^{\text{art}}, \boldsymbol{x}_2^{\text{art}}, \boldsymbol{x}_3^{\text{art}}, \ldots, \boldsymbol{x}_A^{\text{art}}\}, \tag{4}$$

which emulate sequences of audio features $\boldsymbol{x}$ based on the acoustic environments of the ED user. As shown in the lower part of Fig. 2, a DNN-based classifier $\mathbf{C}_2$ is used to assign pseudo-labels to the artificial data samples $\mathcal{U}_{\text{art}}$. Only samples that are classified with at least 75% certainty are kept until the desired number A is reached. Afterwards, $\mathcal{U}_{\text{art}}$ is combined with the baseline training data and the resulting set is used to retrain $\mathbf{C}_1$. This process adapts the embedded classifier to user-specific, personalized acoustic conditions. Finally, the updated parameters of $\mathbf{C}_1$ will be transmitted to the ED.

4. Experimental setup

4.1. Database

In line with the mentioned restrictions on computational complexity, we consider four target classes: ”noise”, ”music”,

Table 1: *Definition of data sets. Brackets indicate the included amount of classes/genres from AudioSet [22]. Abbreviations: ”D” = domestic sounds, ”O” = outside/natural sounds, ”P” = pop, ”R” = rock, ”H” = hip hop, ”E” = electronic, ”C” = classical, ”J” = jazz.*

Data set	Noise types	Music genres	Files
Compreh.	various [46 cl.]	various [11 genr.]	10314
Baseline	D [13 cl.]	P, R, H, E	3023
U1	D [13 cl.]	P, R, H, E	2825
U2	mostly D [14 cl.]	P, E	1902
U3	mostly O [11 cl.]	P, R	1864
U4	O [9 cl.]	J, C	1991

”speech”, and ”speech plus noise” (SpN). Nevertheless, in order to cover a broad variety of acoustic environments, we make use of the Google AudioSet [22] which contains over two million snippets from YouTube videos of 10 s length. Hereby, parts from the ”balanced train”, ”unbalanced train”, and ”evaluation” sets are used at a sampling rate of 16 kHz. For downloading files from the AudioSet, parts of the code provided by [23] are employed. For classes ”noise” and ”music”, the AudioSet files are used directly. As AudioSet occasionally provides ambiguous labels (a specific label only indicates that a certain sound type occurs *somewhere* in the 10 s snippet), we use non-overlapping parts of 10 s length from the MUSAN database [24] speech files and mix them with noise files from AudioSet for the classes ”speech” (mixed at 25 to 40 dB SNR) and ”SpN” (mixed at -10 to 10 dB SNR). Additionally, we define the ratio of maximum energy across signal segments $s[l,k]$ and average signal energy of signal $s[k]$ as λ_s:

$$\lambda_\text{s} = \frac{\max\limits_{l} \sum_{k=1}^{K} s^2[l,k]}{\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K} s^2[l,k]}, \tag{5}$$

with sample index k, segment index l, segment length $K = 4000$, and number of segments $L = 40$. We sort out all files from AudioSet with $\lambda_\text{s} > 5$ as this indicates files with only short energy peaks and thus implies a label assignment that is not representative for the full file.

4.2. Data sets for training, evaluation, and aggregation

We define several data sets with different contents for specific purposes. For this work we select a fraction of the 632 available audio event classes from the Google AudioSet (49 different noise types and 11 popular music genres, which are not listed in detail due to space limitations) and provide a brief overview of the contents in Tab. 1. The ”comprehensive” set contains all considered sound types and music genres and roughly 2500 files for each of the four target classes. Note that the assisting classifier $\mathbf{C}_2$, which is used for labeling the artificial user data, is always trained with the comprehensive data set. The ”baseline” set is a consolidated version of the comprehensive set, only containing domestic environments and four of the most popular music genres. It emulates the acoustic conditions of a typical ED user. Both sets (comprehensive and baseline) are split into train (50%), validation (10%), and evaluation (40%) subsets. For the investigation of user-specific acoustic scenarios we define four user profiles (”U1”-”U4”) with unique characteristics and gradually increasing differences w.r.t. the baseline set. All user sets consist of an evaluation subset and aggregation data (50/50-split) which is used for training the user-specific GMM at ED level.

Table 2: *Architecture of the lightweight ED-based classifier* $\boldsymbol{C}_1$.

Input	Operator	Nodes	Act.	Param.
1×45	BatchNorm	-	-	90
1×45	Dense	9	ReLU	414
1×9	Dense	4	Softmax	40

Table 3: *Architecture of the powerful classifier* $\mathbf{C}_2$*, mainly consisting of a preprocessing block (the first three operators) and the MobileNet V2 [25], and a softmax layer.*

Input	Operator	Nodes	Act.	Param.
5×9	Dense	224	ReLU	3590
224×5	Dense	224	ReLU	
224×224×1	Conv2d	3	ReLU	
224×224×3	Mob.Net	-	-	≈3.5 M
1×1000	Dense	4	Softmax	4004

4.3. Classifier architectures and training schedules

The embedded classifier $\mathbf{C}_1$ consists of two small dense layers with only 544 parameters in total as shown in Tab. 2. For the powerful, DNN-based classifier $\mathbf{C}_2$ we use the MobileNet V2 [25] as core element, complemented by a preprocessing block and a final softmax layer (cf. Tab. 3). Note that $\mathbf{C}_2$ processes the input sequences $\boldsymbol{x}$ as a 5x9-dimensional matrix rather than the 45-dimensional vector representation. Obviously $\mathbf{C}_2$ is much larger than $\mathbf{C}_1$ with more than 3.5 million parameters - while still being small enough so that the back-end application could be implemented in a mobile device. $\mathbf{C}_1$ is trained over 250 epochs using an Adam optimizer [26] with an empirically determined learning rate of 0.008 with a dropout of 0.05 being applied at the first layer. For $\mathbf{C}_2$, we start with the implementation from PyTorch [27] and fine-tune with our data over 250 epochs using the Adam optimizer and a learning rate of 0.001. Here, we apply a mix-up training schedule [28] in order to prevent overfitting and to achieve better robustness. The GMMs are trained using an expectation-maximization (EM) algorithm, treating each feature sequence as 45-dimensional vector. After generating the same number of artificial data samples as in the train/validation subsets of the baseline data, we reshape the 45-dimensional samples back into 5x9-dimensional sequence matrices so that $\mathbf{C}_2$ can assign pseudo-labels. For the retraining process of $\mathbf{C}_1$, we repeat the initial training schedule with a combination of baseline data and artificial samples.

4.4. Experiments

We investigate the classification performance of the embedded classifier $\mathbf{C}_1$ evaluated on the user-specific data sets that are introduced in Sec. 4.2. First, we consider users U1 and U2 in order to demonstrate the benefit of using our baseline system rather than a classifier that has been trained with the comprehensive training set. Subsequently, based on the findings in Sec. 5.1 we consider our baseline system as a starting point and personalize the classifier for U3 and U4. Hereby, we consider GMMs of increasing complexity in terms of underlying components N and compare the performance on the evaluation subsets of the user data. In addition to the overall classification accuracy we specifically evaluate the class ”noise”, which is considered to be most representative for the users’ acoustic environments.

We display the statistical distribution of our evaluation results gained from ten training repetitions on the same data as box plots where the whiskers cover 100% of the results. Additionally, we indicate the performance of $\mathbf{C}_2$ on the respective user data as a blue dashed line (cf. Fig. 4).

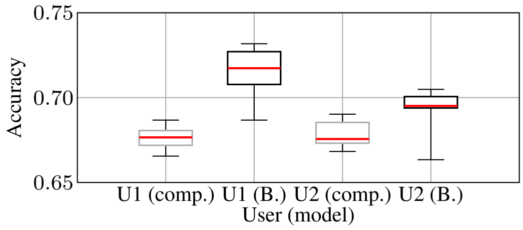


Figure 3: *Effect of using a baseline system ("B.") with reduced training set over a system trained with a comprehensive data set ("comp."), evaluated for users U1 and U2.*

5. Results and discussion

5.1. Baseline classification performance

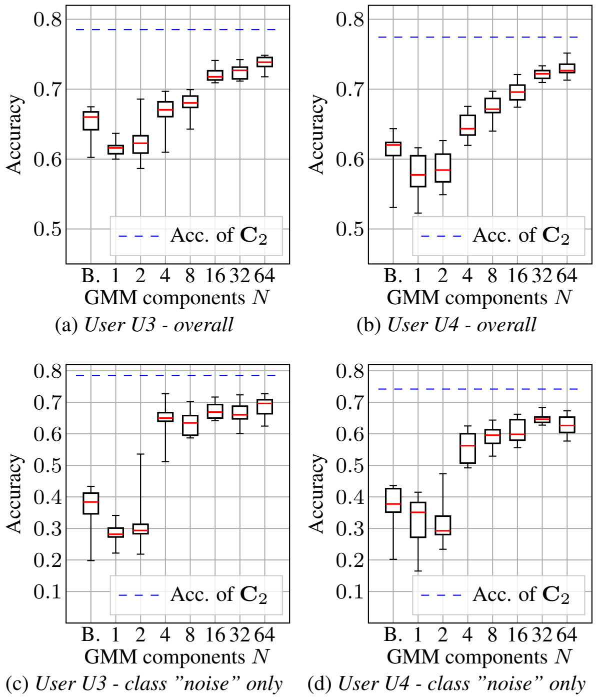
In Fig. 3 we display the classification accuracy of the embedded classifier $\mathbf{C}_1$ evaluated on data of two typical users, U1 and U2. We compare a system trained on the comprehensive data set (cf. Sec. 4.2) with the previously defined baseline system which has been trained with a consolidated training set. Overall, we find a slight advantage for the baseline system which is more significant in the case of U1, as the underlying acoustic conditions in the training data sets are very similar. For U2, we observe marginal improvements when using the baseline system. These results indicate that the baseline classifier should not necessarily be trained on highly diverse data sets (e.g., including all possible musical genres and styles) but should rather focus on the most common acoustic situations. Since U1 represents a typical ED user in this work, we consider the above baseline system as the starting point for the adaptation of the embedded classifier $\mathbf{C}_1$ in the following experiments.

5.2. User-specific personalization of classifier $\mathbf{C}_1$

The two user profiles U3 and U4 are designed to represent acoustic conditions clearly different from the baseline training data (see Sec. 4.2). In the upper part of Fig. 4 we display the overall classification accuracy for the two user sets given the baseline system and several personalized versions of $\mathbf{C}_1$. As expected, we find a decrease of the baseline performance for both users compared to the previously presented results of U1 and U2. However, after applying the proposed (privacy-preserving) personalization method, we can observe a significant performance improvement. When using only very few GMM components for feature aggregation ($N \leq 2$), we observe a slight decrease of performance. A single Gaussian (with diagonal covariance) is clearly not sufficient to model the feature space. For more complex models with $N \geq 16$ we find improvements of more than 10% compared to the baseline. Note that we achieve similar results with diagonal covariance matrices and $N = 32$ and a GMM of order $N = 4$ using full covariance matrices. However, as the latter model requires three times more parameters, we show results only for diagonal covariance matrices.

In the lower part of Fig. 4 we display the results specifically for the class "noise" which we consider to be most representative for a user's acoustic environment. As data sets of U3 and U4 contain entirely different noise types as compared to the baseline set, we consequently observe initial accuracies lower than 45% for both users. After the application of the GMM-based personalization, however, the performance is significantly improved even when only four components are used ($N = 4$). Increasing N beyond four does not seem to provide a significant advantage here.

Overall, results in Fig. 4 indicate that a sufficiently accurate description of the underlying distribution of the user-related audio feature space is required. It is also apparent that increasing the model complexity leads to a better performance of the proposed method with only minor gains beyond $N = 32$. Hereby, the performance of classifier $\mathbf{C}_2$ seems to set an upper bound for the performance of the embedded classifier.



(a) *User U3 - overall* (b) *User U4 - overall*

(c) *User U3 - class "noise" only* (d) *User U4 - class "noise" only*

Figure 4: *ASC performance of the baseline system ("B.") compared to personalized classifiers with increasing GMM complexity, evaluated for users U3 and U4.*

6. Conclusions and outlook

In this work, we have presented a privacy-preserving method for improving ASC performance in ultra-low power EDs such as assistive hearing devices. The method capitalizes on user-specific data through feature aggregation in the ED, data augmentation via resampling, and by incorporating a powerful cloud-based classifier.

At the outset, we have shown that for the small foot-print embedded classifier, it can be beneficial to use a consolidated (baseline) training set which covers the most relevant user scenes but not the full range of all available scenes and signals.

Then, the proposed adaptation method may strongly improve the baseline system for users with deviating specific data sets. Since we use diagonal covariance matrices, the required statistical model for achieving significant improvements is of moderate complexity.

In fact, the success of our GMM-based approach suggests the use of alternative, low-complexity methods. Instead of computing a full-fledged GMM on the ED, the aggregation process could be controlled, for instance, via additional information on the user's daily activities such that the statistical model on the ED can be further simplified. Thus, in future works, we strive to eliminate the necessity for computing a GMM through aggregation methods which rely on auxiliary information gathered in the ED, and will investigate strategies for the back-end supported adaptation of the personalized classifier over longer periods of time.

7. References

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*. Springer, 2018.

[2] J. Abeßer, "A review of deep learning based methods for acoustic scene classification," *Applied Sciences*, vol. 10, no. 6, 2020.

[3] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Scenes and Events 2018 Workshop (DCASE2018)*, 2018, p. 9.

[4] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information Systems*, vol. 42, pp. 245 – 284, 2013.

[5] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 902–909.

[6] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan, "Billion-scale semi-supervised learning for image classification," *arXiv preprint arXiv:1905.00546*, 2019.

[7] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[8] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *International Conference on Learning Representations*, 2017.

[9] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," in *Scenes and Events 2018 Workshop (DCASE2018)*, 2018, p. 19.

[10] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Conformer-based sound event detection with semi-supervised learning and data augmentation," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 100–104.

[11] E. van Engelen Jesper and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[12] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *33rd Annual Meeting of the Association for Computational Linguistics*, 1995, pp. 189–196.

[13] M. Li and Z.-H. Zhou, "Setred: Self-training with editing," in *Advances in Knowledge Discovery and Data Mining*, vol. 3518, 05 2005, pp. 611–621.

[14] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.

[15] S. Sun and F. Jin, "Robust co-training," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 07, pp. 1113–1126, 2011.

[16] A. Goldberg, X. Zhu, A. Singh, Z. Xu, and R. Nowak, "Multi-manifold semi-supervised learning," in *Artificial intelligence and statistics*. PMLR, 2009, pp. 169–176.

[17] A. Demiriz, K. P. Bennett, and M. J. Embrechts, "Semi-supervised clustering using genetic algorithms," *Artificial Neural Networks in Engineering (ANNIE-99)*, pp. 809–814, 1999.

[18] J. Yu, R. Shi, T. He, and K. Guo, "Acoustic scene classification based on feature fusion and dilated-convolution," DCASE2022 Challenge, Tech. Rep., June 2022.

[19] A. Singh, J. A. King, X. Liu, W. Wang, and M. D. Plumbley, "Low-complexity CNNs for acoustic scene classification," DCASE2022 Challenge, Tech. Rep., June 2022.

[20] V. Hamacher, U. Kornagel, T. Lotter, and H. Puder, "Binaural signal processing in hearing aids: Technologies and algorithms," in *Advances in Digital Speech Transmission*, R. Martin, U. Heute, and C. Antweiler, Eds. Hoboken, NJ: John Wiley & Sons, 2008.

[21] S. Ağcaer and R. Martin, "Model-based optimization of a low-dimensional modulation filter bank for DRR and T60 estimation," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[22] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[23] GitHub, "audiosetdl," 2016. [Online]. Available: https://github.com/speedyseal/audiosetdl

[24] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2018, cite arxiv:1801.04381.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[27] PyTorch, "Mobilenet v2." [Online]. Available: https://pytorch.org/hub/pytorch_vision_mobilenet_v2/

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