PROTOTYPICAL NETWORKS FOR DOMAIN ADAPTATION IN ACOUSTIC SCENE CLASSIFICATION

Shubhr Singh[1], *Helen L. Bear*[1], *Emmanouil Benetos*[1,2]

[1]Centre for Digital Music, Queen Mary University of London, UK [2]The Alan Turing Institute, UK

ABSTRACT

Acoustic Scene Classification (ASC) refers to the task of assigning a semantic label to an audio stream that characterizes the environment in which it was recorded. In recent times, Deep Neural Networks (DNNs) have emerged as the model of choice for ASC. However, in real world scenarios, domain adaptation remains a persistent problem for ASC models. In the search for an optimal solution to the said problem, we explore a metric learning approach called prototypical networks using the TUT Urban Acoustic Scenes dataset, which consists of 10 different acoustic scenes recorded across 10 cities. In order to replicate the domain adaptation scenario, we divide the dataset into source domain data consisting of data samples from eight randomly selected cities and target domain data consisting of data from the remaining two cities. We evaluate the performance of the network against a selected baseline network under various experimental scenarios and based on the results we conclude that metric learning is a promising approach towards addressing the domain adaptation problem in ASC.

Index Terms— Metric learning, domain adaptation, acoustic scene classification, episodic training.

1. INTRODUCTION

The task of Acoustic scene classification (ASC) [1] is to accurately identify and categorize an audio stream to an acoustic scene class. “Scene” here refers to an environment which consists of an ensemble of sound events and background noises that humans are used to associate with a semantic label such as “airport”, “home”, “office”, “park” etc. Recently proposed approaches to ASC are based on deep neural networks (DNNs) where the audio stream is transformed to a suitable time-frequency representation, usually a mel-spectrogram, and fed into a convolutional neural network (CNN) as input [2, 3]. The CNN learns discriminative features from the input spectrogram representation, which in turn are used to predict the acoustic scene class of the audio segment.

In real-world scenarios, a performance degradation of ASC models has been observed when evaluated on test datasets having a different distribution than the training data. The datasets can differ from each other in various ways such as acoustic environment, recording conditions, and recording device amongst other factors, leading to a domain shift [4, 5] and subsequently a degradation in the performance of the model. The domain from which the training dataset comes is known as the *source domain* (SD) and the domain from which the test dataset comes is known as the *target domain* (TD).

A typical approach to counter the domain shift phenomenon is to first train the base network on the SD dataset; fix the first n layers of the network; and use labelled TD data to *fine tune* the last few layers of the network [6]. While this strategy is effective, the fine-tuned model often suffers from *catastrophic forgetting* [7, 8] and acts as a barrier to continual learning of new tasks by the model.

Domain adaptation can be supervised [9, 10], unsupervised [11, 12] or semi-supervised [13], depending on how much labelled data is available from both the SD and TD data. Unsupervised domain adaptation requires a large number of unlabelled samples from TD data which might not be always available. Supervised domain adaptation requires a large amount of labelled TD data which also poses a challenge since data annotation is an expensive and resource intensive process and not always feasible. It has been observed that for the same quantity of data, supervised approaches outperform unsupervised approaches [14], hence a supervised approach which can adapt to the TD distribution with only a small number of samples is an attractive proposition because in the majority of scenarios it is not difficult to obtain a small number of labelled examples.

In this work, we use the TUT Urban Acoustic Scenes 2019 development dataset [15] for our experiments. The dataset consists of audio files from 10 different acoustic scenes recorded across 10 different European cities. For the entire scope of this work, we consider cities as different domains and group 8 cities as a part of the source domain data and the remaining two cities as part of the target domain data. The model used in this work is known as *Prototypical Networks* and has been adopted from [16]. In [16], the network was originally used for addressing the problem of few-shot classification in a computer vision application, where the classifier must generalize to new classes that are not seen in the training set, given only a few examples from each new class. Prototypical networks have also been adopted for audio related tasks such as sound event detection [17, 18] and audio tagging [19]. The above works evaluated the effectiveness of prototypical networks in the context of transfer learning, whereas we employ prototypical networks for domain adaptation. Based on the experiments discussed later in the paper, we observe that using prototypical networks in the domain adaptation scenario leads to an increase in classification accuracy over both a CNN baseline network trained on SD data and a CNN network trained on both SD and TD data.

The rest of the paper is organized as follows. In Section 2 we discuss related work, in Section 3 the proposed methodology is described, Section 4 introduces the dataset used and Section 5 discusses the experiment settings, model architecture, and the training & testing procedure. In Section 6, we discuss the results of the experiments and finally, in Section 7 we summarize the paper and briefly discuss future work.

S. Singh is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by UK Research and Innovation [grant number EP/S022694/1] and Queen Mary University of London. E. Benetos is supported by a Turing Fellowship under the EPSRC grant EP/N510129/1.

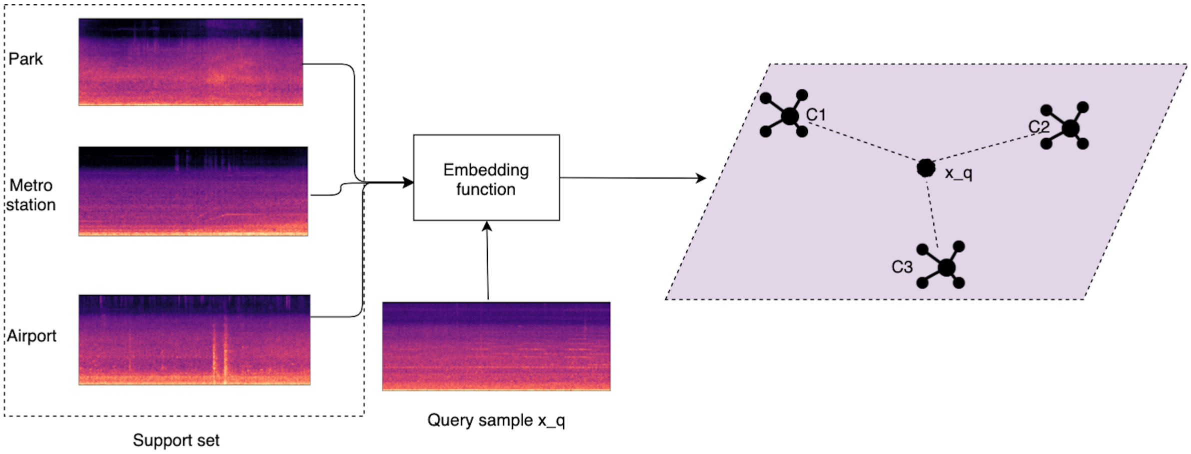

Fig. 1. High level depiction of prototypical networks. The support set samples are projected to the embedding space using the embedding function and prototypes are calculated by taking the mean of the class samples. The query sample is classified based on the distance from the class prototypes.

2. RELATED WORK

A significant body of research in metric based domain adaptation methods have explored different metrics to measure the variational distance between the SD and TD data and then used a DNN to minimize this distance. For example, maximum mean discrepancy (MMD) [20] is based on the idea that the distance between the distribution of SD & TD data in the original space is equivalent to the distance between the means of the projected SD & TD data in the embedded space. MMD has been used in supervised adaptation scenarios where a small amount of target labels is available or unsupervised adaptation scenarios where no target labels are available [10, 21]. Another popular methodology for unsupervised domain adaptation involves adversarial training where the aim is to minimize an approximate domain density distance through an adversarial objective with respect to a domain discriminator [22]. This line of approach has been used for domain adaptation in acoustic scene classification [23]. In [24], first unlabeled TD data is matched to the SD prototype and assigned a “pseudo” label, following which prototypes are calculated on source only, target only and source-target data. A general purpose adaptation is then performed to bring the three different kinds of prototypes close to each other. Our work differs from the previous domain adaptation work in ASC because we intend to learn a domain agnostic embedding space by providing a source domain data comprising multiple sub-domains (different cities).

The core idea of the prototypical networks methodology is that there exists an embedding space in which points cluster around their respective class prototype. The class prototypes are calculated by taking the average of the learned representation of randomly selected few examples from each class and then classify unlabeled input data based on its distance from each class prototype. The training procedure followed in the original work is known as *episodic training*, a training approach adopted from [25]. The principle behind episodic training is that training and test conditions must match, hence each episode mimics a few-shot learning task [26]. So for *K-way-N-shot classification*, each episode includes K classes with N examples from each class. These $K \times N$ samples form the *support* set, which is used to learn the embedding function required to solve the task. In addition, there are further samples from the same classes which are used to evaluate the performance of the model. These are collectively known as the *query* set. In the case of prototypical networks, the support set is used for calculating the embedded prototypes, following which the prediction is made on each embedded query set point based on their squared Euclidean distance from the prototypes. Assuming that the distribution of the feature vectors of acoustic classes would be different across cities, we intend to evaluate if a domain agnostic embedding space can be learnt without attempting to match the SD and TD domain embeddings separately.

3. DOMAIN ADAPTATION MODEL

The goal of prototypical networks and the episodic training methodology is to learn a classifier which can adapt quickly to the target domain with only a few examples. Training is conducted on SD and testing is conducted on TD data.

In each episode of the training, a mini batch is sampled from the training set and is split into a support set consisting of N labelled examples $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ where $x_i \in \mathbb{R}^D$ and $y_i \in \{1, 2, \ldots, K\}$ is the corresponding label, where $K \leq 10$ in this work. The remaining samples of the mini batch are collectively known as the query set Q. S_k denotes the set of samples from class $k \in \{1, 2, \ldots, K\}$.

Prototypical networks compute an M-dimensional prototype $c_k \in \mathbb{R}^M$ through an embedding function $f_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$ with learnable parameters ϕ. For this work, D is 128 and M is 64. Each class prototype is the mean of the embedded support points belonging to its class:

$$c_k = \frac{1}{|S_k|} \sum_{(x_i) \in S_k} f_\phi(x_i) \quad (1)$$

Once the prototypes are calculated, for each sample x_q from the query set, the Euclidean distance $d : \mathbb{R}^M \times \mathbb{R}^M \rightarrow [0, \infty)$ is calculated from each prototype, following which a softmax function over the distances produces a distribution over the classes:

$$p_\phi(y = k|x_q) = \frac{\exp(-d(f_\phi(x_q), c_k))}{\sum_{k'} \exp(-d(f_\phi(x_q, c_{k'}))}. \quad (2)$$

Learning proceeds by minimizing the negative log probability

$J(\phi) = -\log p_\phi(y = k|x_q)$ over the true class k via stochastic gradient descent.

The embedding function is learnt using a simple CNN network described in Section 5. An important point to mention here is that each class in the support set has an equal number of samples, hence a 10-way-5-shot classification would imply that there are 10 classes with 5 samples from each class in the support set. There is no such restriction on the query set, however for the scope of this work, we select equal number of samples per class for both support and query set. The support set here mimics a training set and the query set mimics a test set. The Euclidean distance has been selected based on the reasoning presented in [16] that Euclidean distances belong to the family of Bregman divergences, where it has been shown in [27] that the cluster representative with the minimum distance to its assigned points is the cluster mean. An illustration of the training procedure adopted by prototypical networks is shown in Fig. 1

In this work the classes in the SD dataset and TD dataset are the same, although in some experiments we use only a subset of the total classes in each episode to ascertain if the methodology leads to learning of a more robust embedding function.

4. DATASET & METRICS

We use the TUT Urban Acoustic Scenes 2019 development dataset [28], which has previously been used in Task 1 of the *Detection and Classification of Acoustic Scenes and Events* (DCASE) 2019 [29] challenge. The dataset consists of 14400 audio files distributed equally across 10 acoustic scene classes: “airport”, “shopping mall”, “metro station”, “pedestrian street”, “public square”, “street traffic”, “tram”, “bus”, “metro”, “park”. The acoustic scenes have been recorded across 10 European cities: London, Paris, Barcelona, Lisbon, Lyon, Milan, Prague, Stockholm, Vienna, and Helsinki. The dataset is balanced in terms of numbers of samples per class (1440) and also in terms of duration as each audio file is 10 seconds long. The recordings have been made using 4 devices that captured the audio simultaneously. For this work, both the SD & TD datasets were recorded from the same device - Soundman OKM II Klassik/Studio A3, electret binaural microphone and a Zoom F8 audio recorder [29] challenge. The audio files were recorded at a sampling rate of 48 kHZ and were stored in WAV file format.

5. EXPERIMENTS

5.1. Feature Extraction

Each audio file is transformed to a 128 band log-mel spectrogram with a window size of 2048 samples and hop length of 512 samples. All data samples are z-score normalized . After normalization, the SD data is divided into training and validation sets with a split ratio of 90-10.

5.2. Baseline & Prototypical Model Architecture

The baseline model used in this work is the 4-layered CNN model from [30]. The motivation for using the model was its simple yet effective architecture, which is evident from its performance in the DCASE 2018 ASC task [31].

The baseline model consists of 4 convolutional layers with filter size 5×5. Batch normalization (BN) is applied after each convolutional layer to stabilize the training of the networks, followed by RELU nonlinearity. Global max pooling is applied to the feature maps of the last convolutional layer to summarize the feature maps to a vector. Finally, the vector is fed into a fully connected layer with softmax non-linearity to output the class probabilities for the given input. A detailed overview of the network and its performance on different DCASE tasks can be found in [30].

Table 1. Dataset division by city for each experiment. SD denotes source domain and TD denotes target domain.

Experiment #	SD	TD
Exp1	Rest	Stockholm, Vienna
Exp2	Rest	Paris, London
Exp3	Rest	Milan, Barcelona
Exp4	Rest	Lisbon, Prague
Exp5	Rest	Stockholm, London

The prototypical networks model consists of 4 convolutional blocks with each block comprising 64 filters of shape 3×3, a batch normalization layer, a RELU non linearity, followed by a 2×2 max pooling layer. This encoder architecture acts as the embedding function for each incoming input. The architecture of the model has been directly adapted from [16] without any change since the intention was to evaluate the performance of the model for a particular task.

5.3. Procedure

There are 5 main experiments that cover the scope of this work. The primary difference between each experiment is the configuration of the SD & TD data. In each experiment, we randomly selected 8 cities to form the training set and the data from the remaining 2 cities comprise the test set as shown in Table 1. For each experiment, the baseline & prototypical models are trained and evaluated in the following manner:

- The **Baseline** network is trained on the SD dataset for 150 epochs without early stopping and we evaluate the accuracy on the test/TD dataset.
- We fine-tune the weights of the baseline network by retraining it for 50 epochs with early stopping under 3 different scenarios, i.e. with 50, 150 & 250 samples from the TD dataset and name the models as **Baseline50**, **Baseline150** & **Baseline250**, respectively.
- **Prototypical networks (5-way-10-shot)**: In this particular case, for each episode during training, 5 classes are randomly selected with 10 samples from each class to form the support set. Validation & test episodes follow a 5-way-10-shot procedure. The idea is to evaluate if such an experiment would lead to learning a more robust embedding space.
- **Prototypical networks (10-way-5-shot)**: For each episode, a minibatch of 100 samples is selected from the SD dataset and divided into support & query sets. The support set is formed by selecting 5 samples from each of the 10 classes, hence it is called 10-way-5-shot and the remaining samples go into the query set. The prototypes are calculated from the support set. The same configuration is used for the validation set. Once the embedding space is learnt, we test the accuracy of the prototypical networks model on the TD dataset. During testing, a support set and query set are selected from the TD dataset in each episode with the same configuration as the one used during training. The support set is used to calculate the prototypes and the classification accuracy is calculated over the query set. The final classification accuracy is calculated by

Table 2. Classification accuracy of the Baseline and Prototypical Network models across all the 5 experiments.

Model	Exp1	Exp2	Exp3	Exp4	Exp5	Mean accuracy	Avg. number of TD data samples used for evaluation
Baseline	12 %	17 %	9%	10.2%	11%	11.84 %	2877
Baseline50	24%	33%	29%	26%	29%	28.2%	2827
Baseline150	34%	45%	40.1 %	37%	39%	39.02%	2727
Baseline250	42%	53%	48%	46%	47%	47.2%	2627
Prototypical network (5-way-10-shot)	61%	57%	58.3%	56.4%	54%	57.34%	1440
Prototypical network (10-way-5-shot)	64%	60.3%	63%	61.3%	60%	61.74%	1440
Prototypical network (1-shot)	42%	44%	41%	43.7%	40.0%	42.14%	2577
Prototypical network (zero-shot)	50%	49.2%	52.0%	50.3%	47%	49.7%	2877

averaging over the accuracy across 100 episodes. In order to maintain consistency, we calculate the final accuracy score for all the prototypical networks configurations (5-way-10 shot, 10-way-1 shot and zero-shot) in the same manner.

- **Prototypical networks (10-way-1-shot)**: Same procedure as 10-way-5-shot, except that only 1 sample per class is used for calculating the prototypes.

- **Prototypical networks (zero shot)**: In this scenario, training remains the same as with 10-way-5-shot, however during testing, the samples for the support set are selected from the SD dataset and the query set is sampled from the test/TD dataset.

6. RESULTS & DISCUSSION

All results are shown in Table 2 and are recorded on the TD dataset used for evaluation. We use classification accuracy metric to evaluate the performance of the ASC models as in [32]. The mean accuracy column in the table depicts the average accuracy per model across the five experiments. The last column in the table depicts the average number of samples used from the TD dataset for evaluation across all 8 models (we can only directly compare the test accuracy of models that have the same number of TD samples in the evaluation set).

As we can see from Table 2, the baseline model performs quite poorly on the test/TD dataset. During training, the baseline model accuracy was around 65% on average across all the 5 experiments on the SD dataset, however there is a noticeable drop in accuracy when the baseline model is evaluated on the TD dataset. The evaluation of the baseline model on a vanilla ASC task where the test and training sets come from the same distribution can be referred to from [30] where the model accuracy on the test dataset was 67% on Task 1-A and 59% on Task 1-B of the DCASE 2018 challenge [31]. On adding samples from the TD dataset in the SD dataset and retraining the baseline from scratch (Baseline50, Baseline150 and Baseline 250), the performance of the model improves as the number of TD dataset samples included in the training set is increased.

The prototypical networks on the other hand do not use any TD dataset samples for training. Only once the embedding space is learnt, the test dataset samples are used to form the prototypes and record the accuracy score per episode. It can be observed from the table that prototypical networks show an improvement over the best baseline result. Although the best results across all the experiments are obtained using 10-way-5-shot, these cannot be directly compared with any of the baseline results since the size of the TD dataset used for evaluation is almost half of that used for baseline model and its variants. Only zero-shot prototypical networks can be compared directly with the baseline model and its variants. It can be seen that both zero-shot & 1-shot prototypical networks perform better than the Baseline, Baseline50, and Baseline150. Although the zero-shot prototypical network performs better than Baseline250, the accuracy achieved by the 1-shot network is lower than Baseline250.

Another interesting point to note is that as the number of samples per class increases in the support set, the performance of the prototypical network improves. This can be seen in the case of 1-shot and 10-way-5-shot prototypical networks. The jump in mean accuracy is close to 20%, however since the size of the TD dataset used for evaluation in both the cases is different, we cannot confirm with absolute certainty that the increase in support set samples leads to better performance, although in the case of zero-shot prototypical networks, where the SD data samples are used for calculating the prototypes, the performance improvement over 1-shot prototypical networks is noticeable. It would also be interesting to see if increasing the number of samples per class in the support set leads to diminishing improvement in accuracy. The overall results from the prototypical networks are encouraging and motivate us to explore this line of approach further with different domain adaptation scenarios, as well as with different metrics such as cosine similarity [33].

7. CONCLUSIONS

In this paper, we propose to use a transfer learning methodology referred to as prototypical networks for an acoustic scene classification domain adaptation problem. We conduct 5 different experiments using 8 models within each experiment to evaluate the proposed methodology with respect to an established baseline model. From the obtained results we show that metric learning and in particular prototypical networks are a promising step towards domain adaptation as a problem in the acoustic domain even in the presence of limited data, although it remains to be seen how the network would perform in more complex cases of domain adaptation where the recording devices, acoustic environment, and weather conditions are different.

As future work, we intend to experiment with different domain adaptation scenarios across ASC, not only urban datasets but also across domestic scene datasets to further ascertain the effectiveness of the proposed method. We do acknowledge that key domain shift aspects such as the different label space across the SD & TD datasets have not been explored in this work and we intend to address this also.

8. REFERENCES

[1] Tuomas Virtanen, Mark Plumbley, and Dan Ellis, *Computational Analysis of Sound Scenes and Events*, 09 2017.

[2] Tao Zhang, Jinhua Liang, and Biyun Ding, "Acoustic scene classification using deep CNN with fine-resolution feature," *Expert Systems with Applications*, vol. 143, 2020.

[3] M. Valenti, S. Squartini, A. Diment, G. Parascandolo, and T. Virtanen, "A convolutional neural network approach for acoustic scene classification," in *IJCNN*, 2017, pp. 1547–1554.

[4] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation.," in *CVPR*, 2019, pp. 2507–2516.

[5] Wang Mei and Weihong Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, 02 2018.

[6] Brian Chu, Vashisht Madhavan, Oscar Beijbom, Judy Hoffman, and Trevor Darrell, "Best practices for fine-tuning visual classifiers to new domains," in *ECCV Workshops*, 2016.

[7] Robert M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128 – 135, 1999.

[8] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," in *ICLR*, 2019.

[9] Masayuki Suzuki, Ryuki Tachibana, Samuel Thomas, Bhuvana Ramabhadran, and George Saon, "Domain adaptation of cnn based acoustic models under limited resource settings," in *Interspeech*, 2016, pp. 1588–1592.

[10] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko, "Simultaneous deep transfer across domains and tasks.," in *ICCV*, 2015, pp. 4068–4076.

[11] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," *2019 IEEE/CVF CVPR*, pp. 4888–4897, 2019.

[12] Timothy Miller, "Simplified neural unsupervised domain adaptation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, June 2019, pp. 414–419.

[13] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko, "Semi-supervised domain adaptation via minimax entropy," in *2019 IEEE/CVF ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. 2019, pp. 8049–8057, IEEE.

[14] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto, "Unified deep supervised domain adaptation and generalization," in *ICCV*, 2017.

[15] "TUT Urban Acoustic Scenes 2019," https://zenodo.org/record/2589280.XqdV4C-ZObc.

[16] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., pp. 4077–4087. 2017.

[17] Yu Wang, Justin Salamon, Nicholas J. Bryan, and Juan Pablo Bello, "Few-shot sound event detection," in *2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020 - Proceedings*, 2020.

[18] B. Shi, M. Sun, K. C. Puvvada, C. Kao, S. Matsoukas, and C. Wang, "Few-shot acoustic event detection via meta learning," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[19] Jordi Pons, Joan Serra, and Xavier Serra, "Training neural audio classifiers with few data," 05 2019, pp. 16–20.

[20] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. null, pp. 723–773, Mar. 2012.

[21] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, 2015, p. 97–105.

[22] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," *CVPR*, pp. 2962–2971, 2017.

[23] Konstantinos Drossos, Paul Magron, and Tuomas Virtanen, "Unsupervised Adversarial Domain Adaptation Based On The Wasserstein Distance For Acoustic Scene Classification," in *WASPAA*, 2019.

[24] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *CVPR*, 2019.

[25] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, pp. 3630–3638. 2016.

[26] Fei-Fei Li, Robert Fergus, and Pietro Perona, "One-shot learning of object categories.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, 2006.

[27] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh, "Clustering with bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, Dec. 2005.

[28] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "A multi-device dataset for urban acoustic scene classification," in *DCASE 2018 Workshop*, 2018.

[29] "IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events 2019 (DCASE 2019 Challenge)," http://dcase.community/challenge2019/index.

[30] Qiuqiang Kong, Turab Iqbal, Yong Xu, Wenwu Wang, and Mark D Plumbley, "DCASE 2018 Challenge Surrey cross-task convolutional neural network baseline," in *DCASE2018 Workshop*, 2018.

[31] "DCASE Acoustic Scene Classification Challenge," http://dcase.community/challenge2018/task-acoustic-scene-classification.

[32] Daniele Barchiesi, Dimitrios Giannoulis, Dan Stowell, and Mark D. Plumbley, "Acoustic scene classification," *CoRR*, vol. abs/1411.3715, 2014.

[33] Jinlu Liu, Liang Song, and Yongqiang Qin, "Prototype rectification for few-shot learning," *ArXiv*, vol. abs/1911.10713, 2019.