METRIC LEARNING BASED DATA AUGMENTATION FOR ENVIRONMENTAL SOUND CLASSIFICATION

Rui Lu,[1] *Zhiyao Duan,*[2*] *Changshui Zhang,*[1 †]

[1] Department of Automation, Tsinghua University
State Key Lab of Intelligent Technologies and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, P.R.China
[2] Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY USA

ABSTRACT

Deep neural networks have been widely applied in the field of environmental sound classification. However, due to the scarcity of carefully labeled data, their training process suffers from overfitting. Data augmentation is a technique that alleviates this issue. It augments the training set with synthetic data that are created by modifying some parameters of the real data. However, not all kinds of augmentations are helpful, and some are in fact harmful for the recognition of certain sound concepts. Figuring out the appropriate augmentations for the appropriate training data is thus an interesting question. In this paper, we propose a framework for data augmentation through metric learning. The idea is to first learn a metric from the original training data, and then use it to filter out augmented data samples that are far from original ones in the same class. Experiments on a widely used dataset show that our framework achieves the same performance compared to other augmentation strategies while reducing the amount of training data by a large margin.

Index Terms— Data augmentation, deep neural networks, metric learning, environmental sound classification

1. INTRODUCTION

Great progresses have been made in automatic speech recognition (ASR) and music information retrieval (MIR) for better use of audio modality in intelligent systems. Environmental sound is another important medium in our everyday life [1]. Therefore, environmental sound classification tasks have drawn more and more attention in recent years due to their wide applications in auditory scene understanding [2], machine hearing [3] and surveillance [4, 5]. Moreover, the most advanced machine learning techniques such as deep learning methods have been introduced to this field [6, 7, 8, 9].

Deep learning models have strong capacity to capture complex patterns in data due to their ability of feature learning and highly-nonlinear transformations. To achieve better generalization ability, deep models are particularly dependent on the availability of large quantities of training data. Thus, ever since deep learning becomes popular in computer vision, large carefully-labeled image datasets have been introduced [10, 11]. It is also the same for computer audio literature: except for those relatively small environmental sound datasets [12, 13, 14], google has recently introduced a huge dataset of generic audio events collected from thousands of hours of Youtube videos [15]. However, the amount and quality of annotation are still limited for many audio understanding tasks. Therefore, in recent years, data augmentation has been proposed to synthesize annotated training data from real training data seeds. However, a main problem of existing methods is that the augmentation treats all training data seeds equally, without considering their positive or negative effects on the final task. The proposed framework in our paper is intended to deal with this problem by dynamically selecting those useful augmented samples; we will detail our model in Sect.2.

Similar to the approaches for computer vision, data augmentation methods are also widely used for acoustic models. For ASR, existing augmentation methods are mainly parametric ones such as varying intensity and speed [16], adding background noise at various signal-to-noise ratios (SNRs) [17], etc. And for MIR tasks, musically-inspired deformations such as pitch shifting and time stretching are adopted to enhance the models' robustness [18]. Environmental sound classification tasks also employ similar transformations for additional training data [8, 19]. Despite their effectiveness, the above mentioned augmentation methods for acoustic models have mainly been handled by trial and error, which are time consuming and experience-dependent. An importance weighting method [20] has recently been proposed for ASR which is designed to automatically weight the augmented data. However, this method still makes full use of the augmented data and does not explicitly reduce the amount of training data.

In this paper, we propose a two-stage data augmentation framework for environmental sound classification. Based on observations [8] that classification accuracies for different sound classes are influenced differently by various augmentation techniques, we carry out experiments to show that the class-conditional augmentation strategy is effective to maintain the same performance while reducing the amount of training data required. Furthermore, by employing the proposed deep metric learning method, we dynamically filter out those augmented training samples that may impair our model. We conduct experiments on the UrbanSound8K dataset [12] and compare performances of the trained models with different augmentation strategies. Results show that while preserving the comparable performance, models trained with the proposed data augmentation framework require much less training data and time. Further analyses show that, after the proposed data augmentation procedure, augmented data of different sound classes have different acceptance rates due to their specific augmentation methods.

The remainder of this paper is structured as follows: Sect. 2 introduces the models for classification and the framework for data selection. Sect. 3 are experimental results to illustrate the effectiveness of the proposed framework. Sect. 4 concludes this paper.

[*] ZD acknowledges the National Science Foundation grant No. 1617107.

[†] CZ acknowledges the funding by NSFC (Grant No. 61473167, No. 91420203 and No. 61621136008) and the German Research Foundation (DFG) in Project Crossmodal Learning, DFC TRR-169.

978-1-5386-1631-4/17/$31.00 ©2017 IEEE

Network Structure				
layer	**out-size**	**filters**	**non-linear**	**regular**
conv1	124×124	5×5, 24, (1, 1)	ReLU	BN
pool1	31×62	(4 2), (4, 2)	-	-
conv2	27×58	5×5, 48, (1, 1)	ReLU	BN
pool2	6×29	(4 2), (4, 2)	-	-
conv3	2×25	5×5, 48, (1, 1)	ReLU	BN
full4	64	-	ReLU	DO
full5	10	-	Softmax	DO

Table 1: Input size with 128×128 (#frequency bands × #time frames); filters are denoted as "(#frequency bands × #time frames), #filters, (frequency stride, time stride)"; Column "regular" stands for regularization, "BN" stands for batch normalization and "DO" for dropout with probability 0.5.

2. METHOD

2.1. Data and network

For audio data, we extract log-mel spectrograms with 128 bands covering 0 Hz to 22050 Hz, with a window size of 1024 without overlap at sampling rate 44.1 kHz. We aggregate 128 adjacent frames (2.97 seconds) without overlap to form the time-frequency patches (TF-patches) as inputs of our CNN. Since our main purpose is to evaluate the effectiveness of the proposed data selection method, we adopt the same CNN structure as that used in previous work [8] for ease of comparison. Table 1 details the settings.

For data augmentation, we make use of the MUDA library [18] and JAMS files containing deformation annotations[1] to get the augmented sets as [8]. We apply the following deformations: time stretch (4 factors: 0.81, 0.93, 1.07, 1.23), pitch shift 1 (4 conservative values: -2, -1, 1, 2), pitch shift 2 (4 less conservative values: -3.5, -2.5, 2.5, 3.5), dynamic range compression (4 parameterizations: music and film standard, speech and radio), background noise (4 different background sound recordings). Thus for each sample x, it has 20 possible augmented samples, let's denote them as $\mathcal{A}$. We use three kinds of augmentation schemes in the experiments:

- Brute-force augmentation: for each sample in the training set, we apply all the 20 possible deformations for augmentation.
- Class-conditional augmentation: we apply all the deformations separately as in [8]. By comparing the class-wise accuracy before and after the application of a certain deformation, we can determine the class-conditional beneficial deformations.
- Metric-based augmentation: we augment the training set according to the metric learning based scheme in Algorithm 1.

For all the augmentation experiments, we train on the augmented datasets, carry out validation and testing on the original datasets.

2.2. Metric learning based data augmentation

We show through experiments in Sect. 3 that the class-conditional augmentation approach can achieve comparable performance to the brute-force one. Besides this naive approach, we propose a metric learning based strategy that selects the augmented data on a finer grained level which further reduces the computation consumption.

Algorithm 1: Metric learning based data augmentation.

```
Input  : Data set $\mathcal{S}$
Output: Augmented data set $\mathcal{S}_{aug}$
$\mathcal{S}_{aug} \leftarrow \mathcal{S}$
Partition $\mathcal{S}$ into subsets $\{\mathcal{S}^{(i)}\}_{i=1}^{N}$
for $i = 1, \cdots, N$ do
    $\mathcal{D}_{train}^{(i)} \leftarrow \mathcal{S} \backslash \mathcal{S}^{(i)}$
    $\mathcal{D}_{valid}^{(i)} \leftarrow \mathcal{S}^{(i)}$
    Stage 1: Learn metric
    $f^{(i)} \leftarrow \arg\min_f \mathcal{L}(\mathcal{D}_{train}^{(i)}; f)$
    Stage 2: Select augmented data
    for $\langle x, y \rangle \in \mathcal{D}_{valid}^{(i)}$ do
        Generate its augmented set $\mathcal{A}$ without labels as
          described in Sect. 2.1
        for $a \in \mathcal{A}$ do
            if $y == kNN(a, \mathcal{D}_{train}^{(i)}; f^{(i)})$ then
                $\mathcal{S}_{aug} \leftarrow \mathcal{S}_{aug} \bigcup \{\langle a, y \rangle\}$
            end
        end
    end
end
```

The whole procedure is shown in Algorithm 1. To augment the training set $\mathcal{S}$, we first randomly partition it into subsets $\{\mathcal{S}^{(i)}\}_{i=1}^{N}$ and iteratively augment each of the subsets. For a particular subset $\mathcal{S}^{(i)}$, we regard it as the validation set for metric learning, and $\mathcal{S} \backslash \mathcal{S}^{(i)}$ the training set, let's denote them as $\mathcal{D}_{valid}^{(i)}$ and $\mathcal{D}_{train}^{(i)}$.

Let $x \in \mathcal{X}$ be the input data and $y \in \mathcal{Y} = \{1, \cdots, |\mathcal{Y}|\}$ be its label. Metric learning aims to learn an embedding of the data so that similar data points are closer and dissimilar ones are far from each other, let's denote the learned embedding function $f(x) : \mathcal{X} \rightarrow \mathbb{R}^d$. Standard deep metric learning methods use neural networks to learn f by employing contrastive loss [21] or triplet loss [22, 23]. Since we focus on the classification problem, we integrate the recently proposed multi-class N-pair (N-pair-mc) loss [24] into our framework. Let $\{(x_1, x_1'), (x_2, x_2'), ..., (x_C, x_C')\}$ be C pair of examples from C different classes ($y_i \neq y_j, \forall i \neq j$). We can then build C tuplets denoted as $\{T_i\}_{i=1}^{C}$ from the C pair of data, where $T_i = \{x_i, x_1', x_2', ..., x_C'\}$. Here, x_i is the query data for T_i, x_i' is the positive example and $x_j'(j \neq i)$ are all the negative examples. Then the multi-class N-pair loss can be formulated as:

$$L(\{(x_i, x_i')\}_{i=1}^{C}; f) = \frac{1}{C} \sum_{i=1}^{C} \log(1 + \sum_{j \neq i} \exp(f_i^T f_j' - f_i^T f_i')) \tag{1}$$

Where $f_i = f(x_i), f_i' = f(x_i')$. By the above equation, we apply the N-pair-mc loss for each training batch, it shares the same spirits with existing losses for metric learning. However, this loss function can incorporate data samples across all the classes at once if we set $C = |\mathcal{Y}|$. By this means, we not only get rid of the needs of the time-consuming hard negative mining [25] which aims to reduce false positive rate, but also require an input example to be distinguishable from all the negative examples in current batch at the same time. In our case, we train the metric on $\mathcal{D}_{train}^{(i)}$, and validate on $\mathcal{D}_{valid}^{(i)}$. We formulate the metric learning problem with

[1] https://github.com/justinsalamon/UrbanSound8K-JAMS

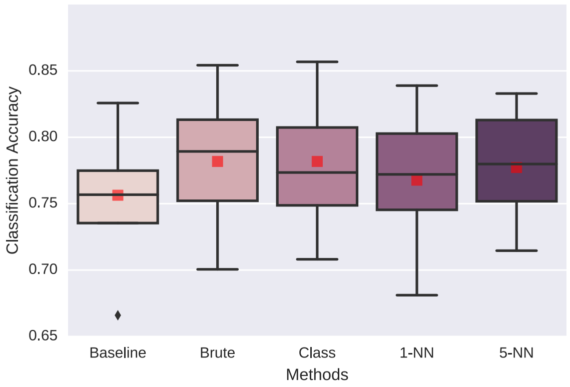

Figure 1: Classification accuracies of different augmentation schemes. Where baseline means no augmentations are implemented. ”Brute” stands for ”Brute-force” augmentation, ”Class” stands for ”Class-conditional” augmentation. ”1-NN” means the proposed ”Metric-based” augmentation with 1-nearest neighbor in Line 12 of Algorithm 1 while ”5-NN” indicates 5-nearest neighbor.

the loss defined in Eqn. 1 as following:

$$f^{(i)}(x) = \arg\min_{f} \mathcal{L}(\mathcal{D}^{(i)}_{train}; f) \tag{2}$$

Given $\langle x, y\rangle \in \mathcal{D}^{(i)}_{valid}$, we first get the corresponding augmented set $\mathcal{A}$. Then $\forall a \in \mathcal{A}$, we calculate its similarity to all the samples in $\mathcal{D}^{(i)}_{train}$ with the equation below:

$$S(x, x') = \frac{f(x)^T f(x')}{||f(x)|| \cdot ||f(x')||} \qquad \forall x, x' \in \mathcal{X} \tag{3}$$

Finally, the predicted label of a is determined through kNN of all the training data, we accept a if y_a agrees with y, or we discard it:

$$y_a = kNN(a, \mathcal{D}^{(i)}_{train}; f^{(i)}) \tag{4}$$

3. EXPERIMENTS

3.1. Setup

We experiment on the UrbanSound8K [12] dataset which contains 10 classes of environmental sound clips with various durations up to 4 seconds. During training, patches are extracted randomly from the full log-mel spectrogram if the clip's length is longer than 2.97 seconds while during testing, prediction is performed by averaging across all the TF-patches since both our experiments and previous work [8] show that averaging performs better than majority voting.

To train our CNN for classification, we set initial learning rate to 0.01 and optimize cross-entropy loss with Adagrad [26]. Each minibatch consists of 256 patches randomly drawn from different sound clips and one epoch goes through all the sound clips. We train for 300 epochs and follow previous work [8] to report 10-fold cross-validation results since the dataset is officially split into 10 folds: when we take a certain fold for testing, another of the rest nine is used for validation. However, we find that data in the 10 folds are not uniformly distributed so that the choice of the validation fold greatly influences the performance of our model. Thus, we modify this setting to make the results more stable: to collect the accuracy on a certain fold, we use the other nine iteratively for validation to train nine models, and finally ensemble these models for prediction.

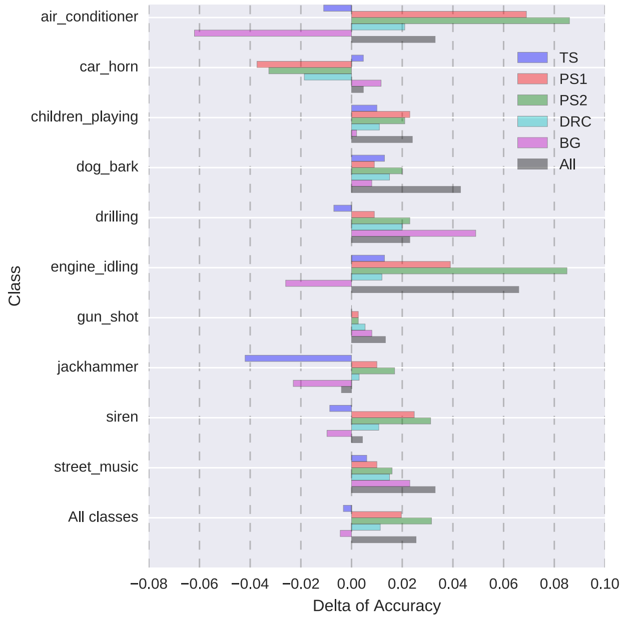

Figure 2: Class-wise difference in accuracy as function of the deformation methods when they are separately applied. Time Stretch (TS), Pitch Shift (PS1 and PS2), Dynamic Range Compression (DRC), Background Noise (BG) and all combined (All).

For metric learning, our partitions $\{S^{(i)}\}_{i=1}^{N}$ just follow the official splitting which means that our training set is partitioned into $N = 8$ subsets. We take the same CNN structure except that we remove the final full-connect layer and optimize N-pair-mc loss by Adam [27] with a learning rate of 0.0001. Each minibatch consists of 20 samples with 2 samples from each of the 10 classes, and one epoch goes through all the sound clips. We train for 300 epochs and select the model with the minimum rank loss on validation set:

$$loss_{rank} = \frac{\sum\limits_{x\in\mathcal{D}}}{|\mathcal{D}|} \frac{\sum\limits_{x^+\in\mathcal{D}^+} \sum\limits_{x^-\in\mathcal{D}^-} \mathbb{I}[S(x, x^+) < S(x, x^-)]}{|\mathcal{D}^+||\mathcal{D}^-|}, \tag{5}$$

where $|\mathcal{D}|$ is the validation set, $\mathcal{D}^+$ is the data with the same label of x and $\mathcal{D}^- = \mathcal{D}\backslash\mathcal{D}^+$. Since we compute the similarity of two data points with Eqn. 3, f_i in Eqn. 1 should have unit norm. However, normalizing f_i in Eqn. 1 directly makes the optimization difficult [24], we thus regularize L^2 norm of the embedding vectors instead.

3.2. Brute-force and class-conditional augmentation

Fig. 1 shows 10-fold cross validation results. Brute-force method gets a mean accuracy of 78.2%, comparable to 79% in [8]. The confusion matrix is shown in Fig. 3(a), and Fig. 3(b) shows difference in confusion matrices with and without augmentation. Though the overall performance is increased, confusions between certain classes also increase. We evaluate the class-wise difference in accu-

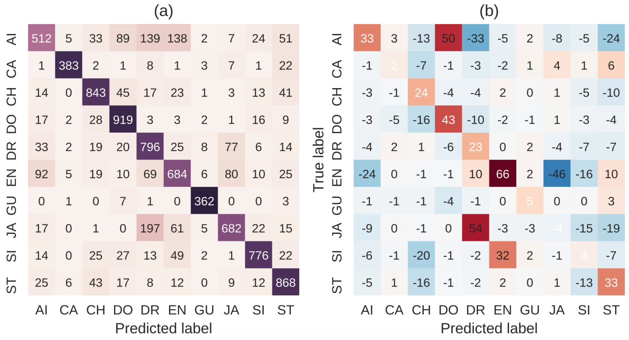

Figure 3: (a) Confusion matrix of the brute-force augmentation method. (b) Differences between the confusion matrices with and without brute-force augmentation. Positive values (red) along the diagonal indicate improvements of the classification results and negative ones (blue) indicate decrease in performance. Negative values (blue) off the diagonal indicate decrease in confusion while those positive ones (red) mean that the confusion is increased.

racy as a function of deformation in Fig. 2 by applying one deformation at a time to inspect its influence on different sound classes.

Once we know the effects of different deformations on different sound classes, the most straight forward idea is to employ class-conditional augmentation method . For example in Fig. 2, for "air conditioner", "PS1", "PS2" and "DRC" are beneficial while "TS" and "BG" are harmful. Thus, for class-conditional augmentation, we simply conduct beneficial augmentations given sound classes. An average accuracy of 77.9% is reported in Fig. 1. It is worth to mention that this experiment distills information of test set since we have known the beneficial augmentations beforehand. We regard this result as the upper bound of class-conditional scheme.

3.3. Metric-based augmentation

For our proposed augmentation method, when applying 5-nearest neighbor in Algorithm 1, we get an average accuracy of 77.7% as in Fig. 1 (5-NN). Though the result is comparable to those obtained by the brute-force and the class-conditional augmentation methods, we have reduced the amount of training data by a large margin, i.e., we only employ 68.75% of the data for training compared to the brute-force method while the class-conditional needs 79.05% of the training data. As a comparison, we also experiment with 1-nearest neighbor and get an average accuracy of 76.8% as in Fig. 1 (1-NN). The boxplot of 1-NN results shows a larger performance variance, which means that the 5-NN approach not only achieves a higher accuracy, but also has a more stable performance.

Since our method makes use of a learned metric accompanied by kNN for data selection in Algorithm 1, we can delve into the details of data augmentation by inspecting the acceptance ratio during kNN (Line 11 to Line 15 of Algorithm 1) of all the sound classes. We detail the class-wise acceptance ratios given different kinds of deformations in Fig. 4 and the average acceptance ratio of each class in Table 2. During this experiment, we have explored 5-NN, 7-NN,$\cdots$, 23-NN and find no obvious difference in acceptance ratios, thus we finally adopt 5-nearest neighbor for data selection.

We have three observations for our metric-based strategy: First, in Fig. 4, different sound classes have different acceptance ratios to all of the five deformations; this indicates that a fine-grained data selection strategy is reasonable. Second, when comparing the acceptance ratios with the confusion matrix of the brute-force scheme in Fig. 3, we see that the more a sound class confuses with other classes, the less our metric-based method accepts its augmented data. For example, class "AI", "EN" and "JA" confuse with other classes the most and they have the lowest acceptance ratios. Intuitively, when we apply deformations to these classes, the augmented data also have higher probabilities of lying in spaces of other classes. Third, we find that most of those rejected samples are surrounded by data points from other classes; this indicates that with infeasible deformations, semantics of the data can be changed: they may sound like other classes or are completely crashed to noise.

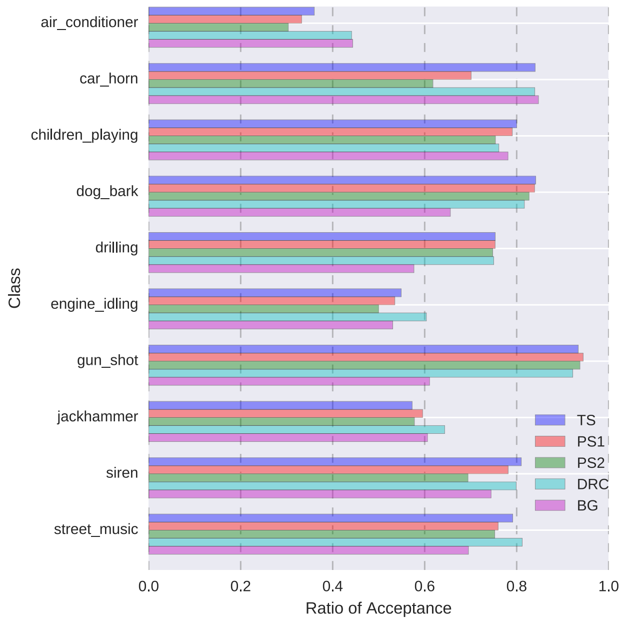

Figure 4: Class-wise acceptance ratios with 5-nearest neighbor by Algorithm 1. For each sound class, we record the acceptance ratios of different deformation methods.

Class	Accept Ratio	Class	Accept Ratio
AI	37.62%	EN	54.37%
CA	76.92%	GU	86.97%
CH	77.74%	JA	59.92%
DO	79.60%	SI	76.58%
DR	71.63%	ST	76.22%

Table 2: Acceptance ratio for each sound class with 5-nearest neighbor by Algorithm 1. The overall acceptance ratio is: 68.75%

4. CONCLUSIONS

We proposed a metric-based framework of data augmentation for environmental sound classification. With this finer-grained strategy, we achieved comparable performance to other strategies while greatly reduced the amount of data required for training. We may also use the same CNN for data selection as is used in classification, which can be explored in future work. Moreover, we plan to make use of generative models for better augmentation schemes.

5. REFERENCES

[1] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[2] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[3] R. Lyon, "Machine hearing: An emerging field [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 5, no. 27, pp. 131–139, 2010.

[4] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on.* IEEE, 2005, pp. 158–161.

[5] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: a systematic review," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, p. 52, 2016.

[6] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, 2015.

[7] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on.* IEEE, 2015, pp. 1–6.

[8] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *arXiv preprint arXiv:1608.04363*, 2016.

[9] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *42th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2017)*, 2017.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[11] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[12] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia.* ACM, 2014, pp. 1041–1044.

[13] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia.* ACM, 2015, pp. 1015–1018.

[14] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1128–1132.

[15] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dartaset for audio events," in *42th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2017)*, 2017.

[16] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 173–182.

[17] T. Schrank, L. Pfeifenberger, M. Zöhrer, J. Stahl, P. Mowlaee, and F. Pernkopf, "Deep beamforming and data augmentation for robust speech recognition: Results of the 4th chime challenge," *Proc. CHiME*, pp. 18–20, 2016.

[18] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation." in *ISMIR*, 2015, pp. 248–254.

[19] J. Salamon, J. P. Bello, A. Farnsworth, and S. Kelling, "Fusing shallow and deep learning for bioacoustic bird species classification," *PLOS ONE*, vol. 11, no. 11, p. e0166866, 2016.

[20] S. Sivasankaran, E. Vincent, and I. Illina, "Discriminative importance weighting of augmented training data for acoustic model training," in *42th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2017)*, 2017.

[21] R. Hadsell, S. Chopra, and Y. Lecun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[23] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, "Embedding label structures for fine-grained feature representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1114–1123.

[24] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, 2016, pp. 1849–1857.

[25] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3908–3916.

[26] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.