Data Augmentation Using Generative Adversarial Network for Environmental Sound Classification

Aswathy Madhu
Department of ECE
College of Engineering
Thiruvananthapuram, India
aswathymadhu@cet.ac.in

Suresh Kumaraswamy
Department of ECE
Government Engineering College, Barton Hill
Thiruvananthapuram, India
sureshk@cet.ac.in

***Abstract*—Various types of deep learning architecture have been steadily gaining impetus for automatic environmental sound classification. However, the relative paucity of publicly accessible dataset hinders any further improvement in this direction. This work has two principal contributions. First, we put forward a deep learning framework employing convolutional neural network for automatic environmental sound classification. Second, we investigate the possibility of generating synthetic data using data augmentation. We suggest a novel technique for audio data augmentation using a generative adversarial network (GAN). The proposed model along with data augmentation is assessed on the UrbanSound8K dataset. The results authenticate that the suggested method surpasses state-of-the-art methods for data augmentation.**

***Index Terms*—data augmentation, generative adversarial network, deep learning, environmental sound classification**

I. Introduction

A substantial amount of research has been conducted in the realm of automatic environmental sound classification over the past few years. Environmental sounds can be defined as the sounds encountered in day-to-day life excluding speech and music. Automatic environmental sound classification plays a decisive role in a wide range of applications such as content based audio search [1], automatic tagging of audio [2], surveillance [3], etc. A variety of signal processing algorithms have been proposed for environmental sound classification. In particular, deep learning approaches like convolutional neural networks (CNN) have been steadily gaining impetus in this context. However, the relative scarcity of publicly available datasets cripple any further improvement in this direction.

A classic solution to this dilemma is data augmentation. In data augmentation, the network is trained with additional synthetic data.The advantages of data augmentation is four-fold. (1) It increases the training data size (2) It eliminates the overfitting problem (3) It makes the network more robust to the variations in data that may be present in any real world application (4) It makes the network learn the most relevant features in the training data. The basic idea behind data augmentation is that the transformations are applied such that the semantic meaning of the labels associated with the data do not change. For example, a pitch-shifted or time-stretched audio of environmental sound would still be an audio of environmental sound. By training the network with this additional data, its performance towards unseen data is enhanced.

Data augmentation was introduced in object recognition in 1998 [4]. Motivated by the promising results of data augmentation in object recognition, Jaitley and Hinton [5] introduced audio data augmentation in speech processing. They transformed the spectrogram of TIMIT speech dataset by using a linear warping technique in the frequency domain called Vocal Tract Length Perturbation (VTLP). Other successful implementations of data augmentation in speech processing include [6], [7] and [8]. Similar attempts can be found in music signal classification also ([9], [10] and [11]). Despite the positive results of data augmentation with speech recognition and music information retrieval, its application is limited in environmental sound classification (eg. [12], [13]).

A major shortcoming of standard augmentation techniques is that they are task specific.They increase the number of hyper parameters in a deep learning structure. This is the premise of using Generative Adversarial Networks (GAN) for augmenting data. Bousmalis et al. [14] suggested a GAN-based model for pixel level domain adaptation. They employed GAN conditioned on source data and noise vectors. They defined two types of losses-a task specific loss and a content similarity loss to stabilize the proposed method. Other successful implementations of GAN based augmentation are [15], [16] and [17]. Despite its utility in image processing applications, the use of GAN for audio data augmentation is in its infancy yet. The first attempt of generating audio using GAN was rendered by Donahue et al. [18]. They explored both time domain and frequency domain strategies for generating audio with GANs. Lee et al. [19] investigated the possibility of using GAN conditioned on class labels for generating audio. They explored concatenation based conditioning and conditional scaling along with several methods for tuning hyper-parameters.

In this paper, we investigate the influence of data augmentation in the context of environmental sound classification using a deep convolutional neural network. Furthermore, a novel augmentation technique based on generative adversarial network is proposed. The performance of the recom-

978-9-0827-9703-9/19/$31.00

mended method is assessed on a publicly accessible dataset-UrbanSound8K. We demonstrate that the proposed method outperforms the state-of-the-art methods for augmentation. The rest of the paper is organized as follows. Section II describes the dataset used and the method adopted in this work. The results obtained from experimental observations and subsequent discussions are presented in section III. Section IV concludes the paper and provides some directions for future research.

II. Method

A. Dataset

The proposed method in this work is evaluated on UrbanSound8K dataset [20]. This is the biggest publicly accessible research oriented dataset of urban sound events characterized by labels. This dataset of 8.75 hours of field recordings contains 8732 sound extracts ($\leq 4s$) of urban sounds belonging to 10 classes: *air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gunshot, jackhammer, siren, and street_music*. The classes are derived from urban sound taxonomy. Based on literature survey, it was found that 4 seconds of audio were sufficient for subjects to identify environmental sounds with 82% accuracy [25]. Hence in UrbanSound8K, the occurrences are limited to a duration of 4 seconds. Longer occurrences are segmented to 4 second slices using a sliding window with a hop size of 2 seconds. To avoid wide variability of class distributions, the authors limit the number of slices per class to 1000, resulting in 8732 labeled slices. The slices are randomly allocated into folds while ensuring that all slices originating from the same Freesound recording fall in the same fold. It is also ensured that the number of audio slices per class in each fold is balanced. The 8732 audio slices are prearranged into 10 folds generated using this method. The audio slices are available in .wav format and the corresponding metadata file is available in .csv format.

B. Method

1) Deep CNN for classification: The deep convolutional neural network in this work is adopted from [21]. The raw audio files are read and processed using Librosa-a python package for music and audio analysis. The input audio is chosen such that it has a length of at least 3 seconds. The valid data which satisfy the above criterion are converted to the feature space by log scaled mel spectrogram computed with *librosa.feature.melspectrogram* using a 2048 point fft window and a hop length of 512. In order to deal with the variable length of samples in the dataset, the length of the training audio is fixed at 2.97 seconds yielding 65,489 samples at a sampling rate of 22050 Hz. This results in a 128×128 log mel spectrogram. Fig. 1 shows exemplary audio signals from the dataset and their corresponding mel spectrograms. Given the input, the network is trained to learn the parameters of the function which maps the training audio to a particular label. The details of the CNN architecture employed is as follows:

- Layer 1: Convolutional layer containing 24 5×5 filters with stride (1,1). This is followed by a max pooling layer with stride (4,2). The activation function is ReLU (Rectified Linear Unit)
- Layer 2: Convolutional layer containing 48 5×5 filters with no padding followed by max pooling layer with stride (4,2) and activation function ReLU.
- Layer 3: Convolutional layer containing 48 5×5 filters with no padding followed by max pooling layer with stride (4,2) and activation function ReLU.
- Layer 4: Flattening layer which converts the output of layer 3 to a numpy array. Dropout is introduced in this layer with rate 0.5 to introduce more randomness.
- Layer 5: Fully connected layer of 64 hidden units with a dropout rate of 0.5 and activation function of ReLU.
- Layer 6: Fully connected layer of 10 output units with a softmax activation function.

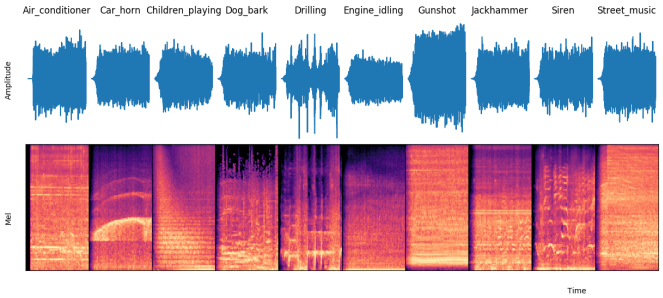


Fig. 1: Exemplary audio signals from the dataset and mel spectrograms

The output shape and the number of parameters in each layer of the employed CNN is shown in Fig. 2. For training, the model optimizes categorical cross entropy using Adam. A constant learning rate of 0.001 was used. The training is stopped after 25 epochs. A validation set is used for hyper parameter tuning. The CNN was implemented in Python with Keras.

2) Data Augmentation: Four simple augmentation techniques resulting in five different augmentation sets as suggested in [21] are implemented. The implementation details of the augmentation techniques are given below.

1) Time stretching (TS): The audio sample is speeded up or slowed down without changing pitch. The time stretching is implemented using *librosa.effects* with a factor of 1.07.
2) Pitch shifting (PS1, PS2): The audio sample is shifted in pitch while there is no change in duration. Pitch shifting is implemented using *librosa.effects* with two factors 2 and 2.5.
3) Additive background noise (BG): A street scene is added to each audio sample using the equation $z = x + w.y$ where x is the original audio sample, y is the background scene sample and w is a weighting parameter whose value is fixed at 0.9.
4) Dynamic range compression (DRC): The dynamic range of each audio sample is compressed either by limiting loud sounds or by amplifying quiet sounds. In this work,

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 124, 124, 24)	624
max_pooling2d_1 (MaxPooling2	(None, 31, 62, 24)	0
activation_1 (Activation)	(None, 31, 62, 24)	0
conv2d_2 (Conv2D)	(None, 27, 58, 48)	28848
max_pooling2d_2 (MaxPooling2	(None, 6, 29, 48)	0
activation_2 (Activation)	(None, 6, 29, 48)	0
conv2d_3 (Conv2D)	(None, 2, 25, 48)	57648
activation_3 (Activation)	(None, 2, 25, 48)	0
flatten_1 (Flatten)	(None, 2400)	0
dropout_1 (Dropout)	(None, 2400)	0
dense_1 (Dense)	(None, 64)	153664
activation_4 (Activation)	(None, 64)	0
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 10)	650
activation_5 (Activation)	(None, 10)	0

Total params: 241,434
Trainable params: 241,434
Non-trainable params: 0

Fig. 2: Model summary of CNN

dynamic range compression was applied using a cross platform command line utility, SoX (Sound eXchange) [22].

By each augmentation 7479 audio samples were generated and prearranged into 10 folds similar to the UrbanSound8K dataset.

3) Data augmentation using GAN: The Generative Adversarial Network for data augmentation is based on WaveGAN defined in [18]. The output dimensionality of WaveGAN is 16384 samples (corresponding to slightly more than 1s of audio at 22.05 kHz). Also length-25 1D convolutions are used with stride 4. The details of the GAN architecture employed is as follows:

For generator, the input is a random noise uniformly distributed between -1 and 1.

- Layer 1: Fully connected and Reshape layer with batch normalization and ReLU activation function which converts the input into 16×1024
- Layer 2: Transposed convolution layer with stride 4, batch normalization and ReLU activation function which converts the 16×1024 input to 64×512
- Layer 3: Transposed convolution layer with stride 4, batch normalization and ReLU activation function which converts the 64×512 input to 256×256
- Layer 4: Transposed convolution layer with stride 4, batch normalization and ReLU activation function which converts the 256×256 input to 1024×128
- Layer 5: Transposed convolution layer with stride 4, batch normalization and ReLU activation function which converts the 1024×128 input to 4096×64
- Layer 6: Transposed convolution layer with stride 4, and tanh activation function which converts the 4096×64 input to 16384×1

The discriminator does the opposite of the generator. Its input is 16384×1 audio sample.

- Layer 1: Convolution layer with stride 4, zero padding, leaky ReLU activation function ($\alpha = 0.2$) and phase shuffle which converts 16384×1 input to 4096×64
- Layer 2: Convolution layer with stride 4, zero padding, batch normalization, phase shuffle and leaky ReLU activation function ($\alpha = 0.2$) which converts the 4096×64 input to 1024×128
- Layer 3: Convolution layer with stride 4, zero padding, batch normalization, phase shuffle and leaky ReLU activation function ($\alpha = 0.2$) which converts the 1024×128 input to 256×256
- Layer 4: Convolution layer with stride 4, zero padding, batch normalization, phase shuffle and leaky ReLU activation function ($\alpha = 0.2$) which converts the 256×256 input to 64×512
- Layer 5: Convolution layer with stride 4, batch normalization and leaky ReLU activation function ($\alpha = 0.2$) which converts the 64×512 input to 16×1024
- Layer 6: Flattening layer which reshapes the input and output layer which connects the input to a single logit.

In this work, the dataset is separated classwise and the wave-GAN is trained on each class for 1000 epochs.The generated sound files are prearranged to 10 folds similar to the original dataset. Inorder to ensure that WaveGAN was properly trained, the generated sound files were evaluated using a similarity score with the original dataset. For two 1D N-length audio signals $A = \{a[n]\}$ and $B = \{b[n]\}$ where $n = 1, 2, ..., N$, the similarity metric is defined as

$$S = \sum_{n=1}^{N} \frac{(a[n] - b[n])^2}{a[n]^2}$$

The lesser the value of S, the more similar the audio files are. The generated audio files with $S > 0.1$ are considered valid. In this work, 7479 valid audio files are generated in 1.53 hours (using Acer Veriton P330 F3 workstation with NVIDIA Quadro GPU). Since WaveGAN generates only 16384 samples (around 1 second audio) the generated samples are periodically extended to 65489 samples to match the audio samples from the UrbanSound8K dataset. The periodic extension has an added advantage that it enhances spectral resolution.

III. Results and Discussions

For each experiment, the performance of the model is estimated with 10-fold cross validation scheme. A single training fold is used as a validation set for hyper parameter tuning. First the CNN is evaluated on the original dataset without augmentation. This serves as the baseline model for comparison. For evaluation we use mean per fold classification accuracy:the most widely used evaluation metric. A mean

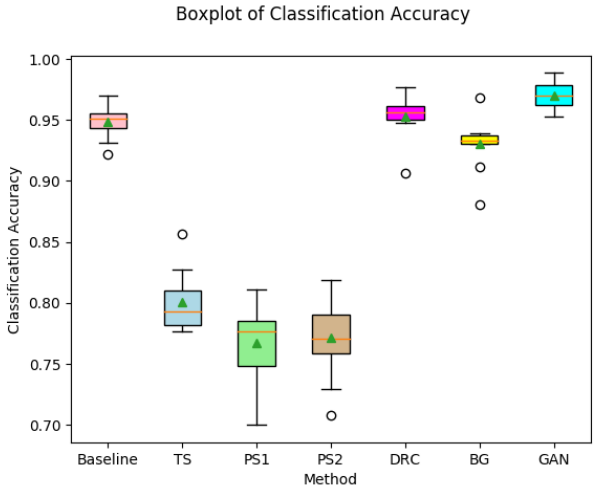


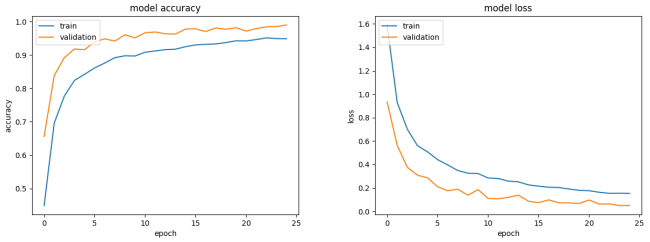
Fig. 3: Distribution of Classification accuracies obtained from 10 fold cross validation

accuracy of 94.84% is obtained for the baseline model. The obtained result is compared with the accuracies obtained by various other approaches such as SKM [23], SB_CNN [21], PiczakCNN [12] and the image recognition networks used by Boddapati et al. [24]. The results are summarized in Table I. The results indicate that the baseline model outshines the state-of-the-art approaches.

TABLE I

Model	Mean per fold Accuracy
SKM [23]	75%
SB_CNN [21]	73%
PiczakCNN [12]	74%
AlexNet [24]	90%
GoogLeNet [24]	93%
Proposed Baseline Model	94.84%

To investigate the effect of augmentation on the dataset, two basic deformations (time stretching and pitch shifting) and two progressive deformations (dynamic range compression and background noise) are applied to the dataset individually. The results are summarized as a box plot shown in Fig. 3. In the plot, the symbol Baseline denotes the model without augmentation, TS denotes Time Stretching by a factor of 1.07, PS1 denotes Pitch Shifting by a factor of 2, PS2 denotes Pitch Shifting by a factor of 2.5, DRC denotes Dynamic Range Compression, BG denotes Additive Backgorund Noise and GAN denotes proposed augmentation technique using GAN. The line inside each box denotes the median of the data. The mean is identified with a marker (green triangle). The circles indicate the outliers. From the box plot, the following observations can be made: (1) The box length indicates that the classification accuracy has a large variability when basic deformations implemented with *librosa.effects* are applied. (2) For baseline method and the proposed approach (GAN), the box lengths are almost the same suggesting that



(a) Model Accuracy (b) Model Loss

Fig. 4: Training history of model with proposed augmentation

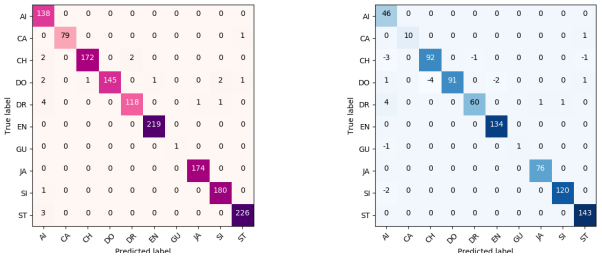
the standard deviations are almost same. (3) For GAN, the whiskers extending from both sides of the box have the same length indicating that the classification accuracies are distributed symmetrically around the mean value. For all other methods, the whiskers are of unequal length indicating that the distribution of classification accuracies is skewed. The mean classification accuracies and standard deviations obtained from this plot along with percentage improvement in accuracy are presented in Table II. From the table, it is clear that basic deformations (time stretching (TS) and pitch shifting (PS1 and PS2)) degrade the performance of the baseline model on UrbanSound8K dataset as indicated by the negative values in the last column. The progressive deformations (dynamic range compression (DRC) and additive background noise (BG)) give comparable performance to the baseline model. A significant improvement in accuracy is obtained only in the proposed method (GAN) as indicated by the last row of Table II.

TABLE II

Method	Mean per fold Classification Accuracy(%)	Standard Deviation	Percentage Improvement in Accuracy
Baseline	94.84	0.0131	
TS	80.08	0.0241	-15.56
PS1	76.69	0.0301	-19.14
PS2	77.16	0.0332	-18.64
DRC	95.31	0.0173	+0.5
BG	93	0.021	-1.94
GAN	**97.03**	**0.0112**	**+2.31**

Fig. 4 and Fig. 5 summarize the results obtained by applying the proposed augmentation technique using GAN on UrbanSound8K dataset. The training history of the model on training and validation datasets can be visualized in terms of two plots (a) A plot of model accuracy over training epochs and (b) A plot of model loss over training epochs as shown in Fig. 4. The plot of accuracy indicates that the performance of the model is comparable on both training and validation datasets. We can see that the training for the model is adequate since the trend for accuracy is stagnant on both datasets. The plot of loss indicates that the model is neither underfit nor overfit since validation loss is comparable to training loss. The parallel plots suggest that early stopping is not needed.

The method yielded a mean per fold classification accuracy



(a) Confusion matrix (b) Difference matrix

Fig. 5: (a) Confusion matrix of augmentation using GAN (b) Difference between the confusion matrices with and without augmentation

of 97.03% with a standard deviation of 0.01. Fig. 5 shows the performance of the proposed augmentation method by means of confusion matrix and the difference between the confusion matrices with and without augmentation. In these matrices, the symbols AI, CA, CH, DO, DR, EN, GU, JA, SI and ST stand for the classes *air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gunshot, jackhammer, siren, and street_music* respectively. The positive main diagonal entries indicate that the classification accuracy is enhanced for all classes with augmentation.The negative off diagonal entries suggest that the confusion between the concerned classes is decreased while the positive off diagonal entries suggest that the confusion between the concerned classes is increased. For example, the confusion between *air_conditioner* and *children_playing* classes is reduced while that between *drilling* and *air_conditioner* is enhanced.

IV. Conclusion

In this work, we investigated the possibility of generating synthetic audio data using a novel technique based on Generative Adversarial Networks. A baseline model employing a deep convolutional neural network was developed for environmental sound classification. The baseline model outperformed the state-of-the-art approaches for environmental sound classification. The performance of the baseline model can further be improved by using the proposed augmentation method. We compared the performance of the proposed augmentation method with four different basic deformations. We observed that the basic deformations degrade the performance of the baseline model. The adopted GAN architecture outputs 16384 samples (slightly more than 1s of audio at 22.05 kHz). This output length is not sufficient for automatic environmental sound classification. We circumvent this problem by periodic extension of the GAN output. The generalization to longer output by modifying the GAN architecture is an interesting direction for future research. We wish to explore this idea further.

References

[1] T. Virtanen and M. Helén, "Probabilistic model based similarity measures for audio query-by-example," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2007, pp. 82-85.

[2] S. Duan, J. Zhang, P. Roe, and M. Towsey, "A survey of tagging techniques for music, speech and environmental sound," *Artificial Intelligence Review*, vol. 42, pp. 637-661, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10462-012-9362-y.

[3] M. Cristani, M. Bicego, and V. Murino, "Audio-visual event recognition in surveillance video sequences," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 257-267, 2007.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, " Gradient- based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2323, 1998.

[5] N. Jaitly and G. E. Hinton, "Vocal Tract Length Perturbation (VTLP) improves speech recognition" in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, 2013.

[6] X. Cui, V. Goel and B. Kingsbury, " Data Augmentation for Deep Neural Network Acoustic Modeling". *ICASSP*, pp. 5582-5586, 2014.

[7] T. Ko, V. Peddinti, D. Povey and S. Khudanpur, "Audio augmentation for speech recognition", *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, Jan 2015, pp. 3586-3589.

[8] T. DeVries and G. W. Taylor, "Dataset Augmentation in Feature Space", 2017, pp. 1-12. [Online]. Available: https://arxiv.org/abs/1702.05538v1

[9] T. L. H. Li and A. B. Chan, " Genre classification and the invariance of MFCC features to key and tempo"in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6523 LNCS, pp. 317-327.

[10] E. J. Humphrey, J. P. Bello and Y. LeCun, "Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics" in *International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 403-408.

[11] Y. Han and K. Lee, "Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation", vol. 14, no. 8, pp. 1-11, 2016.

[12] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *25th International Workshop on Machine Learning for Signal Processing (MLSP)*, Boston, MA, USA, Sep. 2015, pp. 1-6.

[13] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 6440-6444.

[14] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] A. Radford, L. Metz, and S. Chintala, " Unsupervised representation learning with deep convolutional generative adversarial networks" in *ICLR*, 2016.

[16] D. Berthelot, T. Schumm and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks", ArXiv:1703.10717, 2017.

[17] T. Karras, T. Aila, S. Laine and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation". in *ICLR*, 2018.

[18] C. Donahue, J. McAuley, M. Puckette, "Synthesizing Audio with Generative Adversarial Networks", [Online]. Available: https://arxiv.org/abs/1802.04208.

[19] C. Y. Lee, A. Toffy, G. J. Jung and W. Han "Conditional WaveGAN", *IEEE Signal Processing Letters*, Nov. 2016.

[20] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM14)*, Orlando, FL, USA, Nov. 2014.

[21] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification", *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279-283, Mar. 2017.

[22] [Online]. Available: http://sox.sourceforge.net/sox.html.

[23] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 171-175.

[24] V. Boddapati, A. Petef, J. Rasmusson and L. Lundberg, "Classifying environmental sounds using image recognition networks", in *International Conference on Knowledge Based and Intelligent Information and Engineering Systems, (KES2017)*, 6-8 September, Marseille, France, 2017.

[25] S. Chu, S. Narayanan and C. C. Kuo, "Environmental Sound Recognition with time-frequency audio features", *IEEE TASLP*, vol. 17, no. 6, pp. 1142-1158, 2009.