Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification

Justin Salamon and Juan Pablo Bello

***Abstract*—The ability of deep convolutional neural networks (CNNs) to learn discriminative spectro-temporal patterns makes them well suited to environmental sound classification. However, the relative scarcity of labeled data has impeded the exploitation of this family of high-capacity models. This study has two primary contributions: first, we propose a deep CNN architecture for environmental sound classification. Second, we propose the use of audio data augmentation for overcoming the problem of data scarcity and explore the influence of different augmentations on the performance of the proposed CNN architecture. Combined with data augmentation, the proposed model produces state-of-the-art results for environmental sound classification. We show that the improved performance stems from the combination of a deep, high-capacity model and an augmented training set: this combination outperforms both the proposed CNN without augmentation and a "shallow" dictionary learning model with augmentation. Finally, we examine the influence of each augmentation on the model's classification accuracy for each class, and observe that the accuracy for each class is influenced differently by each augmentation, suggesting that the performance of the model could be improved further by applying class-conditional data augmentation.**

***Index Terms*—Deep convolutional neural networks (CNNs), deep learning, environmental sound classification, urban sound dataset.**

I. Introduction

The problem of automatic environmental sound classification has received increasing attention from the research community in recent years. Its applications range from context aware computing [1] and surveillance [2] to noise mitigation enabled by smart acoustic sensor networks [3].

To date, a variety of signal processing and machine learning techniques have been applied to the problem, including matrix factorization [4]–[6], dictionary learning [7], [8], wavelet filterbanks [8], [9] and most recently deep neural networks [10], [11]. See [12]–[14] for further reviews of existing approaches. In particular, deep convolutional neural networks (CNNs) [15] are, in principle, very well suited to the problem of environmental sound classification: first, they are capable of capturing energy modulation patterns across time and frequency when applied to spectrogram-like inputs, which has been shown to be an important trait for distinguishing between different, often noise-like, sounds such as engines and jackhammers [8]. Second, by using convolutional kernels (filters) with a small receptive field, the network should, in principle, be able to successfully learn and later identify spectro-temporal patterns that are representative of different sound classes even if part of the sound is masked (in time/frequency) by other sources (noise), which is where traditional audio features such as Mel-Frequency Cepstral Coefficients fail [16]. Yet the application of CNNs to environmental sound classification has been limited to date. For instance, the CNN proposed in [11] obtained comparable results to those yielded by a dictionary learning approach [7] (which can be considered an instance of "shallow" feature learning), but did not improve upon it.

Deep neural networks, which have a high model capacity, are particularly dependent on the availability of large quantities of training data in order to learn a nonlinear function from input to output that generalizes well and yields high classification accuracy on unseen data. A possible explanation for the limited exploration of CNNs and the difficulty to improve on simpler models is the relative scarcity of labeled data for environmental sound classification. While several new datasets have been released in recent years (e.g., [17]–[19]), they are still considerably smaller than the datasets available for research on, for example, image classification [20].

An elegant solution to this problem is *data augmentation*, that is, the application of one or more deformations to a collection of annotated training samples which result in new, additional training data [20]–[22]. A key concept of data augmentation is that the deformations applied to the labeled data do not change the semantic meaning of the labels. Taking an example from computer vision, a rotated, translated, mirrored or scaled image of a car would still be a coherent image of a car, and thus it is possible to apply these deformations to produce additional training data while maintaining the semantic validity of the label. By training the network on the additional deformed data, the hope is that the network becomes invariant to these deformations and generalizes better to unseen data. Semantics-preserving deformations have also been proposed for the audio domain, and have been shown to increase model accuracy for music classification tasks [22]. However, in the case of environmental sound

Manuscript received August 15, 2016; revised November 21, 2016; accepted January 17, 2017. Date of publication January 23, 2017; date of current version February 8, 2017. This work was supported in part by NSF under Award 1544753. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sascha Spors.

J. Salamon is with the Music and Audio Research Laboratory and the Center for Urban Science and Progress, New York University, New York, NY 10003 USA (e-mail: justin.salamon@nyu.edu).

J. P. Bello is with the Music and Audio Research Laboratory at New York University, New York, NY 10003 USA (e-mail: jpbello@nyu.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2017.2657381

classification the application of data augmentation has been relatively limited (e.g., [11], [23]), with Piczak [11] [which used random combinations oftime shifting, pitch shifting and time stretching for data augmentation] reporting that "simple augmentation techniques proved to be unsatisfactory for the UrbanSound8K dataset given the considerable increase in training time they generated and negligible impact on model accuracy."

In this letter, we present a deep CNN architecture with localized (small) kernels for environmental sound classification. Furthermore, we propose the use of data augmentation to overcome the problem of data scarcity and explore different types of audio deformations and their influence on the model's performance. We show that the proposed CNN architecture, in combination with audio data augmentation, yields state-of-the-art performance for environmental sound classification.

II. Method

A. Deep CNN

The deep CNN architecture proposed in this study is comprised of three convolutional layers interleaved with two pooling operations, followed by two fully connected (dense) layers. Similar to previously proposed feature learning approaches applied to environmental sound classification (e.g., [7]), the input to the network consists of time–frequency patches (TF-patches) taken from the log-scaled mel-spectrogram representation of the audio signal. Specifically, we use Essentia [24] to extract log-scaled mel-spectrograms with 128 components (bands) covering the audible frequency range (0–22 050 Hz), using a window size of 23 ms (1024 samples at 44.1 kHz) and a hop size of the same duration. Since the excerpts in our evaluation dataset (described below) are of varying duration (up to 4 s), we fix the size of the input TF-patch X to 3 s (128 frames), i.e., $X \in \mathbb{R}^{128\times128}$. TF-patches are extracted randomly (in time) from the full log-mel-spectrogram of each audio excerpt during training as described below.

Given our input X, the network is trained to learn the parameters Θ of a composite nonlinear function $\mathcal{F}(\cdot|\Theta)$, which maps X to the output (prediction) Z

$$Z = \mathcal{F}(X|\Theta) = f_L(\cdots f_2(f_1(X|\theta_1)|\theta_2)|\theta_L) \qquad (1)$$

where each operation $f_\ell(\cdot|\theta_\ell)$ is referred to as a *layer* of the network, with $L = 5$ layers in our proposed architecture. The first three layers, $\ell \in \{1, 2, 3\}$, are convolutional, and expressed as

$$Z_\ell = f_\ell(X_\ell|\theta_\ell) = h(W * X_\ell + b), \quad \theta_l = [W, b] \qquad (2)$$

where X_ℓ is a three-dimensional (3-D) input tensor consisting of N *feature maps*, W is a collection of M 3-D kernels (also referred to as filters), $*$ represents a valid convolution, b is a vector bias term, and $h(\cdot)$ is a point-wise activation function. Thus, the shapes of X_ℓ, W, and Z_ℓ are (N, d_0, d_1), (M, N, m_0, m_1) and $(M, d_0 - m_0 + 1, d_1 - m_1 + 1)$, respectively. Note that for the first layer of our network $d_0 = d_1 = 128$, i.e., the dimensions of the input TF-patch. We apply strided max-pooling after the first two convolutional layers $\ell \in \{1, 2\}$ using a stride size equal to the pooling dimensions (provided below), which reduces the dimensions of the output feature maps and consequently speeds up training and builds some scale invariance into the network. The final two layers, $\ell \in \{4, 5\}$, are fully connected (dense) and consist of a matrix product rather than a convolution:

$$Z_\ell = f_\ell(X_\ell|\theta_\ell) = h(WX_\ell + b), \quad \theta_\ell = [W, b] \qquad (3)$$

where X_ℓ is flattened to a column vector of length N, W has shape (M, N), b is a vector of length M, and $h(\cdot)$ is a point-wise activation function.

The proposed CNN architecture is parameterized as follows:

1) ℓ_1: 24 filters with a receptive field of (5,5), i.e., W has the shape (24,1,5,5). This is followed by (4,2) strided max-pooling over the last two dimensions (time and frequency, respectively) and a rectified linear unit (ReLU) activation function $h(x) = \max(x, 0)$.
2) ℓ_2: 48 filters with a receptive field of (5,5), i.e., W has the shape (48, 24, 5, 5). Like ℓ_1, this is followed by (4,2) strided max-pooling and an ReLU activation function.
3) ℓ_3: 48 filters with a receptive field of (5,5), i.e., W has the shape (48, 48, 5, 5). This is followed by an ReLU activation function (no pooling).
4) ℓ_4: 64 hidden units, i.e., W has the shape (2400, 64), followed by an ReLU activation function.
5) ℓ_5: 10 output units, i.e., W has the shape (64,10), followed by a softmax activation function.

Note that our use of a small receptive field $(5, 5)$ in ℓ_1 compared with the input dimensions $(128, 128)$ is designed to allow the network to learn small, localized patterns that can be fused at subsequent layers to gather evidence in support of larger "time–frequency signatures" that are indicative of the presence/absence of different sound classes, even when there is spectro-temporal masking by interfering sources.

For training, the model optimizes cross-entropy loss via minibatch stochastic gradient descent [25]. Each batch consists of 100 TF-patches randomly selected from the training data (without repetition). Each 3 s TF-patch is taken from a random position in time from the full log-mel-spectrogram representation of each training sample. We use a constant learning rate of 0.01. Dropout [26] is applied to the input of the last two layers, $\ell \in \{4, 5\}$, with probability 0.5. L2-regularization is applied to the weights of the last two layers with a penalty factor of 0.001. The model is trained for 50 epochs and is checkpointed after each epoch, during which it is trained on random minibatches until one-eighth of all training data is exhausted (where by training data we mean all the TF-patches extracted from every training sample starting at all possible frame indices). A validation set is used to identify the parameter setting (epoch) achieving the highest classification accuracy, where prediction is performed by slicing the test sample into overlapping TF-patches (1-frame hop), making a prediction for each TF-patch and finally choosing the sample-level prediction as the class with the highest mean output activation over all frames. The CNN is implemented in Python with Lasagne [27], and we used Pescador [28] to manage and multiplex data streams during training.

B. Data Augmentation

We experiment with four different audio data augmentations (deformations), resulting in five augmentation sets, as detailed below. Each deformation is applied directly to the audio signal prior to converting it into the input representation used to train the network (log-mel-spectrogram). Note that for each augmentation it is important that we choose the deformation parameters such that the semantic validity of the label is maintained. The deformations and resulting augmentation sets are described below.

1) *Time stretching (TS):* slow down or speed up the audio sample (while keeping the pitch unchanged). Each sample was time stretched by four factors: $\{0.81, 0.93, 1.07, 1.23\}$.
2) *Pitch shifting (PS1):* raise or lower the pitch of the audio sample (while keeping the duration unchanged). Each sample was pitch shifted by four values (in semitones): $\{-2, -1, 1, 2\}$.
3) *Pitch shifting (PS2):* since our initial experiments indicated that pitch shifting was a particularly beneficial augmentation, we decided to create a second augmentation set. This time each sample was pitch shifted by four larger values (in semitones): $\{-3.5, -2.5, 2.5, 3.5\}$.
4) *Dynamic range compression (DRC):* compress the dynamic range of the sample using four parameterizations, three taken from the Dolby E standard [29] and one (radio) from the icecast online radio streaming server [30]: {music standard, film standard, speech, radio}.
5) *Background noise (BG):* mix the sample with another recording containing background sounds from different types of acoustic scenes. Each sample was mixed with four acoustic scenes: {street-workers, street-traffic, street-people, park}[1]. Each mix z was generated using $z = (1 - w) \cdot x + w \cdot y$, where x is the audio signal of the original sample, y is the signal of the background scene, and w is a weighting parameter that was chosen randomly for each mix from a uniform distribution in the range $[0.1, 0.5]$.

The augmentations were applied using the MUDA library [22], to which the reader is referred for further details about the implementation of each deformation. MUDA takes an audio file and corresponding annotation file in JAMS format [31], [32], and outputs the deformed audio together with an enhanced JAMS file containing all the parameters used for the deformation. We have ported the original annotations provided with the dataset used for evaluation in this study (see below) into JAMS files and made them available online along with the post-deformation JAMS files.[2]

C. Evaluation

To evaluate the proposed CNN architecture and the influence of the different augmentation sets we use the UrbanSound8K dataset [17]. The dataset is comprised of 8732 sound clips of up to 4 s in duration taken from field recordings. The clips span ten environmental sound classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. By using this dataset, we can compare the results of this study to previously published approaches that were evaluated on the same data, including the dictionary learning approach proposed in [7] (spherical k-means, henceforth SKM) and the CNN proposed in [11] (PiczakCNN) which has a different architecture to ours and did not employ augmentation during training. PiczakCNN has two convolutional layers followed by three dense layers, the filters of the first layer are “tall” and span almost the entire frequency dimension of the input, and the network operates on two input channels: log mel-spectra and their deltas.

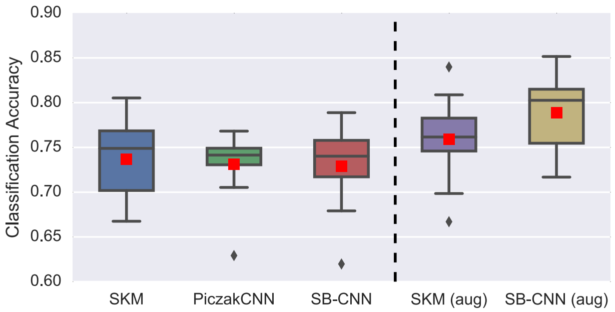


Fig. 1. Left of the dashed line: classification accuracy without augmentation—dictionary learning (SKM [7]), Piczak’s CNN (PiczakCNN [11]), and the proposed model (SB-CNN). Right of the dashed line: classification accuracy for SKM and SB-CNN with augmentation.

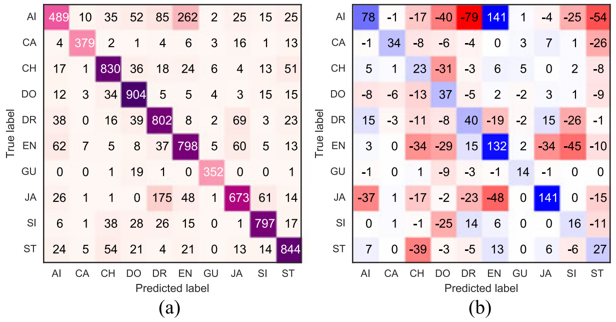
The proposed approach and those used for comparison in this study are evaluated in terms of classification accuracy. The dataset comes sorted into ten stratified folds, and all models were evaluated using 10-fold cross validation, where we report the results as a box plot generated from the accuracy scores of the ten folds. For training the proposed CNN architecture we use one of the nine training folds in each split as a validation set for identifying the training epoch that yields the best model parameters when training with the remaining eight folds.

III. Results

The classification accuracy of the proposed CNN model (SB-CNN) is presented in Fig. 1. To the left of the dashed line we present the performance of the proposed model on the original dataset without augmentation. For comparison, we also provide the accuracy obtained on the same dataset by the dictionary learning approach proposed in [7] (SKM, using the best parameterization identified by the authors in that study) and the CNN proposed by Piczak [11] (PiczakCNN, using the best performing model variant (LP) proposed by the author). To the right of the dashed line we provide the performance of the SKM model and the proposed SB-CNN once again, this time when using the augmented dataset (all augmentations described in Section II-B combined) for training.

We see that the proposed SB-CNN performs comparably to SKM and PiczakCNN when training on the original dataset without augmentation (mean accuracy of 0.74, 0.73, and 0.73 for SKM, PiczakCNN and SB-CNN, respectively). The original dataset is not large/varied enough for the convolutional model to outperform the “shallow” SKM approach. However, once we increase the size/variance in the dataset by means of the proposed augmentations, the performance of the proposed model increases significantly, yielding a mean accuracy of 0.79. The corresponding per-class accuracies (with respect to the list of classes provided in Section II-C) are 0.49, 0.90, 0.83, 0.90, 0.80, 0.80, 0.94, 0.68, 0.85, 0.84. Importantly, we note that while the proposed approach performs comparably to the “shallow” SKM

[1] We ensured these scenes did not contain any of the target sound classes.
[2] https://github.com/justinsalamon/UrbanSound8K-JAMS.



(a) (b)

Fig. 2. (a) Confusion matrix for the proposed SB-CNN model with augmentation. (b) Difference between the confusion matrices yielded by SB-CNN with and without augmentation: negative values (red) off the diagonal mean the confusion is reduced with augmentation, positive values (blue) off the diagonal mean the confusion is increased with augmentation. The positive values (blue) along the diagonal indicate that overall the classification accuracy is improved for all classes with augmentation.

learning approach on the original dataset, it significantly outperforms it ($p = 0.0003$ according to a paired two-sided t-test) using the augmented training set. Furthermore, increasing the capacity of the SKM model (by increasing the dictionary size from $k = 2000$ to $k = 4000$) did not yield any further improvement in classification accuracy. This indicates that the superior performance of the proposed SB-CNN is not only due to the augmented training set, but rather thanks to the combination of an augmented training set with the increased capacity and representational power of the deep learning model.

In Fig. 2(a) we provide the confusion matrix yielded by the proposed SB-CNN model using the augmented training set, and in Fig. 2(b) we provide the difference between the confusion matrices yielded by the proposed model with and without augmentation. From the latter we see that overall the classification accuracy is improved for all classes with augmentation. However, we observe that augmentation can also have a detrimental effect on the confusion between specific pairs of classes. For instance, we note that while the confusion between the air conditioner and drilling classes is reduced with augmentation, the confusion between the air conditioner and the engine idling classes is increased.

To gain further insight into the influence of each augmentation set on the performance of the proposed model for each sound class, in Fig. 3 we present the difference in classification accuracy (the delta) when adding each augmentation set compared to using only the original training set, broken down by sound class. At the bottom of the plot we provide the delta scores for all classes combined. We see that most classes are affected positively by most augmentation types, but there are some clear exceptions. In particular, the air conditioner class is negatively affected by the DRC and BG augmentations. Given that this sound class is characterized by a continuous "hum" sound, often in the background, it makes sense that the addition of background noise that can mask the presence of this class will deteriorate the performance of the model. In general, the pitch augmentations have the greatest positive impact on performance, and are the only augmentation sets that do not have a negative impact on any of the classes. Only half of the classes benefit from applying all augmentations combined more than they would from the application of a subset of augmentations. This suggests that the performance of the model could be improved further by the application of class-conditional augmentation during training—one could use the validation set to identify which augmentations improve the model's classification accuracy for each class, and then selectively augment the training data accordingly. We intend to explore this idea further in future work.

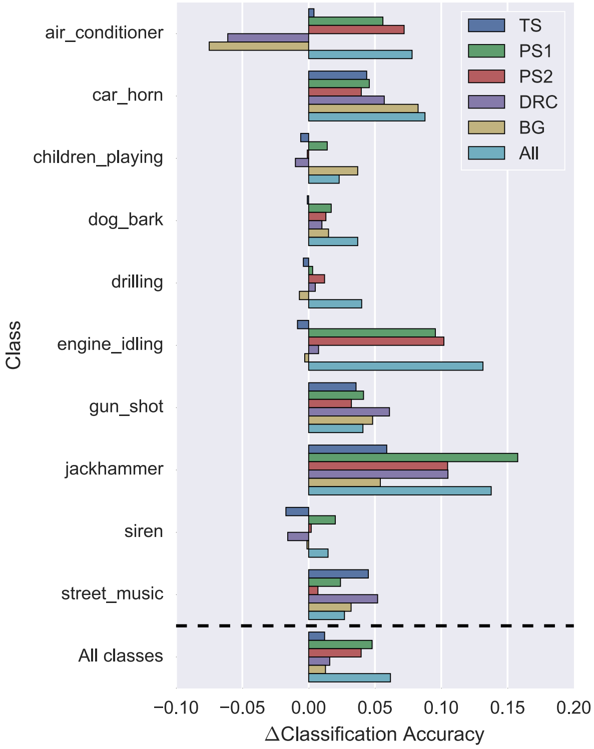


Fig. 3. Difference in classification accuracy for each class as a function of the augmentation applied: time shift (TS), pitch shift (PS1 and PS2), dynamic range compression (DRC), background noise (BG), and all combined (All).

IV. Conclusion

In this letter we proposed a deep CNN architecture which, in combination with a set of audio data augmentations, produces state-of-the-art results for environmental sound classification. We showed that the improved performance stems from the combination of a deep, high-capacity model and an augmented training set: this combination outperformed both the proposed CNN without augmentation and a "shallow" dictionary learning model with augmentation. Finally, we examined the influence of each augmentation on the model's classification accuracy. We observed that the performance of the model for each sound class is influenced differently by each augmentation set, suggesting that the performance of the model could be improved further by applying class-conditional data augmentation.

Acknowledgment

The authors would like to thank Brian McFee and Eric Humphrey for their valuable feedback, and Karol Piczak for providing details on the results reported in [11].

REFERENCES

[1] S. Chu, S. Narayanan, and C.-C. Kuo, “Environmental sound recognition with time-frequency audio features,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 6, pp. 1142–1158, Aug. 2009.

[2] R. Radhakrishnan, A. Divakaran, and P. Smaragdis, “Audio analysis for surveillance applications,” in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, New Paltz, NY, USA, Oct. 2005, pp. 158–161.

[3] C. Mydlarz, J. Salamon, and J. P. Bello, “The implementation of low-cost urban acoustic monitoring devices,” *Appl. Acoust.*, vol. 117, pp. 207–218, 2016.

[4] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, “Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, Apr. 2015, pp. 151–155.

[5] E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, “Detection of overlapping acoustic events using a temporally-constrained probabilistic model,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 6450–6454.

[6] V. Bisot, R. Serizel, S. Essid, and G. Richard, “Acoustic scene classification with matrix factorization for unsupervised feature learning,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 6445–6449.

[7] J. Salamon and J. P. Bello, “Unsupervised feature learning for urban sound classification,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, Apr. 2015, pp. 171–175.

[8] J. Salamon and J. P. Bello, “Feature learning with deep scattering for urban sound analysis,” in *Proc. 2015 23rd Eur. Signal Process. Conf.*, Nice, France, Aug. 2015, pp. 724–728.

[9] J. T. Geiger and K. Helwani, “Improving event detection for audio surveillance using gabor filterbank features,” in *Proc. 23rd Eur. Signal Process. Conf.*, Nice, France, Aug. 2015, pp. 714–718.

[10] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, “Polyphonic sound event detection using multi label deep neural networks,” in *Proc. 2015 Int. Joint Conf. Neural Netw.*, Jul. 2015, pp. 1–7.

[11] K. J. Piczak, “Environmental sound classification with convolutional neural networks,” in *Proc. 25th Int. Workshop Mach. Learning Signal Process.*, Boston, MA, USA, Sep. 2015, pp. 1–6.

[12] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, “Detection and classification of acoustic scenes and events: An IEEE AASP challenge,” in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, New Paltz, NY, USA, Oct. 2013, pp. 1–4.

[13] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, “Detection and classification of acoustic scenes and events,” *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.

[14] S. Sigtia, A. Stark, S. Krstulovic, and M. Plumbley, “Automatic environmental sound recognition: Performance versus computational cost,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 11, pp. 2096–2107, Nov. 2016.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[16] C. V. Cotton and D. P. W. Ellis, “Spectral vs. spectro-temporal features for acoustic event detection,” in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, New Paltz, NY, USA, Oct. 2011, pp. 69–72.

[17] J. Salamon, C. Jacoby, and J. P. Bello, “A dataset and taxonomy for urban sound research,” in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[18] K. J. Piczak, “ESC: Dataset for environmental sound classification,” in *Proc. 23rd ACM Int. Conf. Multimedia*, Brisbane, Australia, Oct. 2015, pp. 1015–1018.

[19] A. Mesaros, E. Fagerlund, A. Hiltunen, T. Heittola, and T. Virtanen, “TUT sound events 2016, development dataset,” 2016. [Online]. Available: http://dx.doi.org/10.5281/zenodo.45759. Accessed on: Aug. 10, 2016.

[20] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[21] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in *Proc. Int. Conf. Document Anal. Recognit.*, Edinburgh, U.K., Aug. 2003, vol. 3, pp. 958–962.

[22] B. McFee, E. Humphrey, and J. Bello, “A software framework for musical data augmentation,” in *Proc. 16th Int. Soc. Music Inf. Retrieval Conf.*, Malaga, Spain, Oct. 2015, pp. 248–254.

[23] G. Parascandolo, H. Huttunen, and T. Virtanen, “Recurrent neural networks for polyphonic sound event detection in real life recordings,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 6440–6444.

[24] D. Bogdanov *et al.* “ESSENTIA: An audio analysis library for music information retrieval,” in *Proc. 14th Int. Soc. Music Inf. Retrieval Conf.*, Curitiba, Brazil, Nov. 2013, pp. 493–498.

[25] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proc. 19th Int. Conf. Comput. Statist.*, Paris, France, Aug. 2010, pp. 177–186. [Online]. Available: http://dx.doi.org/10.1007/978-3-7908-2604-3_16

[26] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learning Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] S. Dieleman *et al.* “Lasagne: First release,” 2015. [Online]. Available: https://github.com/Lasagne/Lasagne

[28] B. McFee and E. J. Humphrey, “Pescador: 0.1.0,” 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.32468

[29] Dolby Labortories, Inc., “Standards and practices for authoring Dolby Digital and Dolby E bitstreams,” 2002.

[30] “Icecast streaming media server forum,” [Online]. Available: http://icecast.imux.net/viewtopic.php?t=3462. Accessed on: Aug. 12, 2016.

[31] E. J. Humphrey, J. Salamon, O. Nieto, J. Forsyth, R. Bittner, and J. P. Bello, “JAMS: A JSON annotated music specification for reproducible MIR research,” in *Proc. 15th Int. Soc. Music Inf. Retrieval Conf.*, Taipei, Taiwan, Oct. 2014, pp. 591–596.

[32] B. McFee *et al.*, “Pump up the JAMS: V0.2 and beyond,” Music and Audio Research Laboratory, New York University, New York, NY, USA, Oct. 2015, unpublished.