A Comparison of Deep Learning Techniques for the Classification of Urban Sounds with Multiple Data Augmentation

Aamer Abdul Rahman[1][0000-0003-1189-9472] and Shazia Hasan[2]

[1] Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science, Pilani, Dubai Campus, Dubai, UAE
[2] Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science, Pilani, Dubai Campus, Dubai, UAE
ar.aamer@gmail.com

Abstract. Growing levels of urbanization brings higher levels of noise pollution. Automatic detection and classification of these sounds can help city planners keep track of the sources of noise pollution. Deep learning methods have proven to outperform traditional classification methods in the presence of large datasets. This paper aims to study the and compare the performance of different deep learning models, which includes deep neural networks (DNN), recurrent neural networks (RNN), convolutional neural networks (CNN) and Convolutional Recurrent Neural Networks (CRNN) on the classification of the UrbanSound8k dataset. Previous works have shown that audio data augmentation methods such as pitch shifting, time stretching and background noise insertion have led to an improvement in the performance of deep learning models when it comes to learning discriminative spectro-temporal patterns such as environmental sounds. The influence of data augmentation methods such as SpecAugment, pitch shifting, time stretching and background noise on the performance of the deep learning algorithms are observed. The audio samples from the dataset are transformed into their respective 2D spectrogram representation using short time Fourier transform (STFT). Two of the best performing models are ensembled using the model averaging method to achieve an accuracy and F1-score of 79.5% and 80.6% respectively.

Keywords: Environmental Sound Classification, Deep Learning, Data Augmentation, Mel-Spectrogram, Deep Neural Network, Convolutional Neural Network, Recurrent Neural Network.

1 Introduction

The permeation of noises such as cars honking, busses whirring, sirens wailing and the mechanical clattering from construction becomes inescapable with the increasing levels of urbanization and population growth. Various studies have indicated that prolonged exposure to noise pollution not only affects the quality of life, but it can have a direct impact on the health and wellbeing of the population [1]. It can lead to disturbances in hearing, concentration and sleep patterns which consequently leads to stress, fatigue and other health issues. Hearing loss can result upon continuous exposure to sounds

above 85 db. While many laws and restrictions limiting noise have been placed in many cities and urban areas, these restrictions have been inconsistently enforced. Therefore, there is an imperative need for smart city solutions that can detect and classify sources of noise pollution in urban areas so that this. Other potential application for classification of urban sounds include predictive maintenance systems in industrial environments, assist systems for those who face issues with hearing, multimedia indexing and retrieval and security systems in smart homes [2]. The aim of this paper is to study the performance of deep learning models such as deep neural network (DNN), recurrent neural network (RNN), convolutional neural network (CNN) and CRNN on the classification of the UrbanSound8k dataset as well as evaluate influence of various audio data augmentation methods such as SpecAugment, pitch shifting, time stretching and background noise insertion on the performance of these different deep learning models.

2 Related Work

In [2], Salamon et al. introduced the UrbanSound8k dataset and a taxonomy for these sounds. The dataset included 8732 samples from 10 classes: “air conditioner, car horn, children playing, dog barking, drilling, engine idling, gun shot, jackhammer, siren and street music.” A convolutional neural network was used in [3] to classify the UrbanSound8k dataset once it had been preprocessed into segmented spectrograms. This architecture achieved an accuracy of 73%.

Salamon et al. improved the performance of the convolutional neural network by applying data augmentation [4]. Pitch shifting, time stretching and background noise insertion was carried out on the raw audio samples before being transformed into log-scaled Mel-spectrograms. The CNN model with data augmentation achieved an accuracy of 79%. In [5], Sang et al. proposed a convolutional recurrent neural network that learned spectro-temporal representations from the raw audio waveforms as opposed to transforming the audio samples into their spectrogram representation. The model achieved a competitive accuracy of 79.06%.

Li et al. compared the effectiveness of 5 deep learning models: Gaussian Mixture Model, Deep Neural Network, Recurrent Neural Network, Convolutional Deep Neural Network and i-vector on the DCASE 2016 challenge dataset [13]. The experiments were carried out on the following features: MFCC, Binaural MFCC, log Mel-spectrogram and temporal features extracted using OpenSMILE. 4-fold cross validation was used for evaluating the model.

A convolutional neural network architecture with no fully connected layers at the end of the network was proposed by Dai et al. in [6]. The model was trained on the raw audio samples and achieved an accuracy of 71.8% which is competitive with other models trained on Mel-spectrogram representations. A dilated convolutional neural network with 1D data augmentation was proposed in [7] by Zhang et al. Furthermore, this paper evaluated the effectiveness of different activation functions such as Softplus, rectified linear unit (ReLU), LeakyReLU, parametric ReLU (PReLU), and exponential linear unit (ELU). The models were evaluated on the UrbanSound8k, CICESE and ESC-10 datasets. It was reported that the LeakyReLU activation function produced the

best performance on the UrbanSound8k ESC-10 datasets while the PReLU activation produced better results on the CICESE dataset. The dilated convolutional neural network with the leaky ReLU activation function and data augmentation achieved an accuracy of 81.9% on the UrbanSound8k dataset.

This paper differs from previous work in the following ways: 1) This work compares the performance of different deep learning models such as DNN, CNN, RNN and CRNN for the classification of the UrbanSound8k dataset. 2) There is an in-depth comparison of the influence of data augmentation techniques such as SpecAugment, pitch shifting, time stretching, and background noise insertion and their combinations on different deep learning models. To the best of our knowledge, there is not any prior literature that investigates the influence of the different augmentation methods with these different deep learning methods on the UrbanSound8k dataset.

3 Dataset Description

The dataset used in this paper, UrbanSound8k, was introduced in [2]. The dataset comprises of 10 classes of samples with 8732 audio data samples. The 10 classes are as follows: “street music, drilling, engine idling, dog barking, children playing, car horn, air conditioner, gun shot, siren and jackhammer.” Each of the audio tracks spanned 4 seconds and those samples that exceeded 4 seconds were sliced with a sliding window with a hope size of 2 seconds.

4 Data Preprocessing

1D Data Augmentation. In order to overcome the sparsity of data issue, data augmentation is applied on the UrbanSound8k dataset. The data augmentation methods used on the raw audio dataset include time stretching, pitch shifting and background noise insertion. The values for pitch shifting and time stretching are based on the work done in [4].

Time Stretching. The audio datasets were slowed down and sped up by factors of 0.81 and 1.07 respectively.

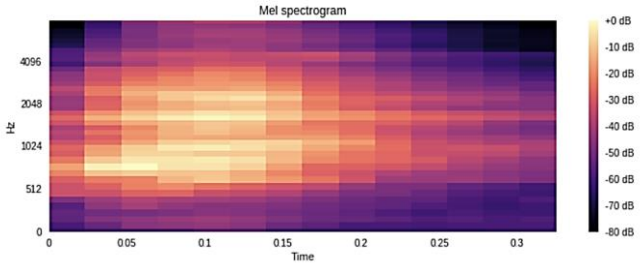
Pitch Shifting. The pitch of audio tracks was decreased and increased by -1, -2, 1 and 2.

Background Noise. Background noise, created with Numpy, was inserted into each audio sample.

Audio Transformations. Previous work has shown transforming audio samples into Mel scaled spectrograms has improved the performance of classifiers on audio datasets as opposed using their raw amplitude vs time form. The audio samples are transformed into their log scaled Mel-spectrogram forms in this study.

Mel Spectrogram. Spectrograms are a 3-dimensional representation of the audio sample. The three dimensions being time, frequency and amplitude represented by the x-axis, y-axis and colour gradient respectively. The Mel-spectrograms are obtained using the Short-Time Fourier Transform (STFT). The librosa python library is used to extract the Mel-spectrograms with a window size of 93 ms, covering a frequency range of 22050 Hz with of 40 bands. The max frame count of the audio samples was calculated to be 174. Zero padding was used on the samples that did not reach the maximum frame count. Fig. 1 depicts the Mel-spectrogram of an audio sample pertaining to the "dog barking" class from UrbanSound8k.

Fig. 1. Mel-Spectrogram.



SpecAugment (SA). Unlike the previous augmentation methods, SpecAugment is applied to the dataset after the audio samples have been transformed into their Mel-spectrogram representation [11]. SpecAugment has proven to improve robustness of audio classifiers trained on spectrograms. SpecAugment warps the spectrogram in the time scale and applies successive mask blocks on the time steps and Mel-frequency channels.

5 Methodology

Deep Learning. Deep learning models have been able to achieve extraordinary levels of performance over the years. The multiple processing layers in deep learning models enabled them learn patterns and representation from raw data. Deep learning has proven to be highly effective in processing video, images, speech and audio [12].

Deep Neural Network. Artificial neural networks (ANN), inspired by the working of neurons in the human brain, consist of layers of interconnected nodes. The connections between these nodes represent weights that are updated by methods such as backpropagation as the neural network learns from training data. As the neural network learns from data where the desired outputs are already known, it is a form of supervised learning. There are generally three types of layers present in ANN: the input layer, hidden layer and output layer. When there are many hidden layers, it is termed as a deep neural network (DNN).

The DNN model used in this study consists of 4 dense layers. The first dense layer consists of 256 units and receives inputs of dimensions (128,6960), where 128 is the

batch size and 6960 is the product of the number of Mel-bands (40) and the frame counts (174). The ReLU activation function returns the same value if the input value is positive and returns zero if the input value is negative. The leakyReLU activation function returns small negative values when the input value is less than zero. The work by Zhang et al. indicated that the leakyReLU activation function worked relatively better on the UrbanSound8k dataset, hence the same is used in this study [7]. Dropout regularization, set to 0.4, is added to prevent overfitting of the model. The output of the first dense layer is of dimensions (128,256). Similar to the first layer, the second and third hidden layers consisted of 256 units with the leakyReLU activation function and dropout regularization set to 0.4. The final dense layer consists of 10 units, as there are 10 classes of datasets in UrbanSound8k. This layer is followed by the softmax activation function to represent probability distribution over the 10 different classes.

The network is trained using the Adam optimizer to minimize the categorical cross-entropy loss function with early stopping. Fig. 3 shows a representation of the DNN architecture which was used. The learning rate is set to 0.0001.

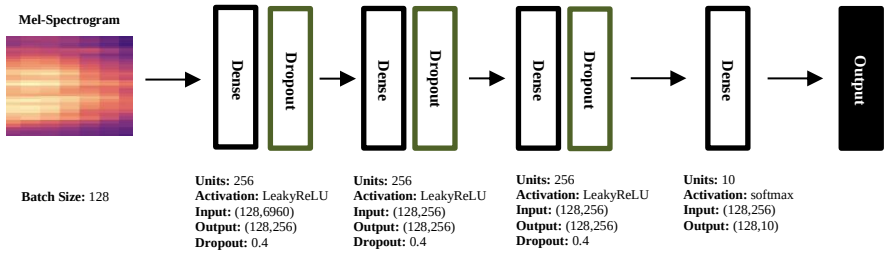


Fig. 3. DNN Architecture.

Recurrent Neural Network. Recurrent neural networks are a type of neural network where the output is fed back into the neural network as inputs for the next state. This enables the neural network to make use of historical data for computation making RNN's highly effective for sequenced data such as speech or handwriting datasets where the current data state is highly dependent on the previous data state. Long short term memory (LSTM) networks are a variant of RNNs. RNNs tend to discard long term memory, making them unable to make computations based on data further in the past. LSTMs overcomes this by having a mechanism to retain maintain relevant long term data at each step and do not suffer from the vanishing gradient problem.

The LSTM architecture used in this study, as seen in Fig. 4, consists of 2 LSTM layers and 1 dense layer. The two LSTM layers have 128 and 64 units respectively. The dense layer has 10 units. Adam optimizer was used to minimize the categorical cross-entropy loss function. The learning rate is set to 0.0001.

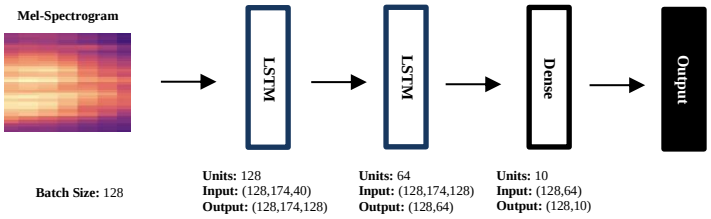


Fig. 4. RNN Architecture.

Convolutional Neural Network. Conventionally, CNNs consist of convolutional layers, pooling layers and one or more dense layers. The convolutional layers comprise of kernels that learn to extract relevant representations from images while training. In order to reduce the number of parameters in the neural network, pooling layers are used. The CNN model used in this study, as represented in Fig. 5, consists of 4 convolutional layers with batch normalization and 1 fully connected layer. Batch normalization is added stabilize and speed up the learning process. Zero padding is not used with the convolutional layers here. The first convolutional layer consists of 32 filters with 3x3 kernels. Spatial dropout layers are added to prevent the model from overfitting. The dropout rates are set to 0.07 and 0.14. The leakyReLU activation function is used. The subsequent convolutional layers consist of 32, 64 and 64 filters. A spatial dropout layer is not added after the fourth convolutional layer. Finally, a global average pooling layer is added followed by a fully connected layer with 10 nodes that represent the probability distribution over the 10 classes of the dataset. Adam optimizer is used to minimize the categorical cross-entropy loss function. The learning rate is set to 0.0001.

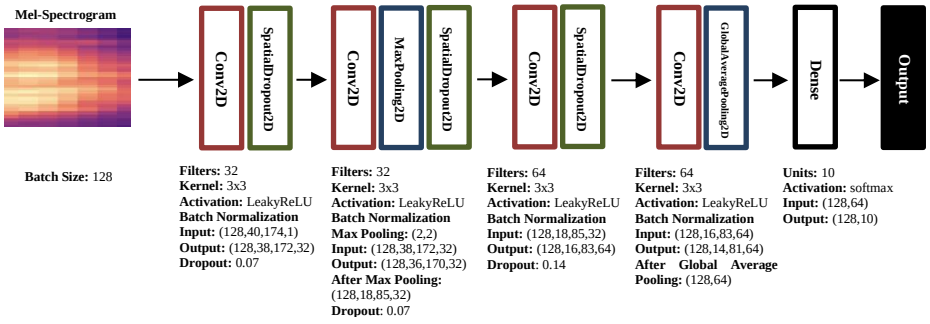


Fig. 5. CNN Architecture.

Convolutional Recurrent Neural Network. CRNNs consist of both CNN and RNN layers. The model consists of 4 convolutional layers and 2 LSTM layers followed by a fully connected layer. The CNN components of the model are identical to the model used above except that the global average pooling layer is replaced by a 2D max pooling layer. The data is reshaped as it passes from the convolutional layer to the LSTM layer. The LSTM layers consist of 32 units each followed by dropout of 0.3. The final dense

layer consists of 10 nodes. Adam optimizer was used to minimize the cross categorical cross-entropy loss function. The learning rate was set to 0.0001. The CRNN architecture is depicted in Fig. 6.

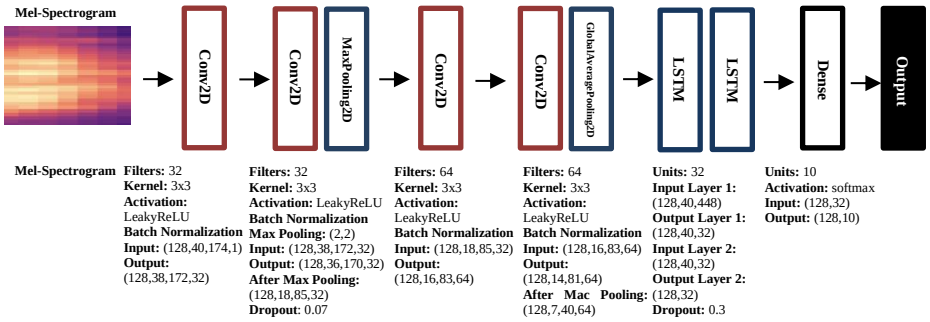


Fig. 6. CRNN Architecture.

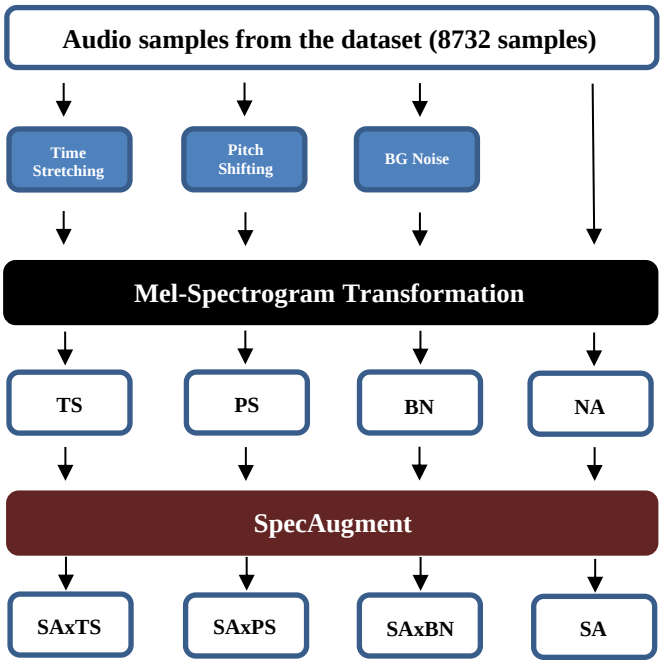


Fig. 7. Data augmentation pipeline.

6 Data Preparation

The models are evaluated and compared using the accuracy and F1-score evaluation metrics. The official 10-fold cross evaluation was used to train and evaluate the models. Accordingly, the data is divided into 10 folds, 9 folds used for training and 1 fold is used as the test set. Upon applying different combinations of the data augmentation techniques, ten datasets are obtained and are described in Table 1. The pipeline by which these datasets are formed is shown in Figure 7.

Table 1. Ten datasets with varying combination of data augmentation methods applied.

Dataset	Description
NoAug (NA)	NA is the data containing the original 8732 audio samples from the dataset that is not subjected to any augmentation method
TS	TS data is obtained by applying time stretching data augmentation to the 8732 audio samples to produce 8732 × 2 samples. Consequently, the 8732 NA audio samples are also combined to produce a total of 26,196 audio samples.
PS	PS data contains the audio data samples from the dataset that are subjected to pitch shifting data augmentation to produce 8732 × 4 samples. In addition to this the 8732 NA audio samples are also combined to produce a total of 43,660 audio samples.
BN	BN data contains the 8732 audio data samples from the dataset that are subjected to background noise insertion data augmentation. In addition to this the 8732 NA audio samples are also combined to produce a total of 17,464 audio samples.
SA	SA data contains the 8732 audio data samples from the dataset that are subjected to SpecAugment data augmentation. In addition to this the 8732 NA audio samples are also combined to produce a total of 17,464 audio samples.
SAxTS	SAxTS data is obtained by subjecting the 26,196 audio data samples from TS data to SpecAugment data augmentation to produce a total of 26,196 × 2 samples.
SAxPS	SAxPS data is obtained by subjecting the 43,660 audio data samples from PS data to SpecAugment data augmentation to produce a total of 43,660 × 2 samples. SAxBN: SAxBN data is obtained by subjecting the 17,464 audio data samples from BN data to SpecAugment data augmentation to produce a total of 17,464 × 2 samples.
PS+TS+BN	Contains 69,856 data samples from PS, TS, BN and NA data.
ALL	ALL data is obtained by subjecting the 69,856 data samples from PS+TS+BN data to SpecAugment to contain 69856 × 2 samples.

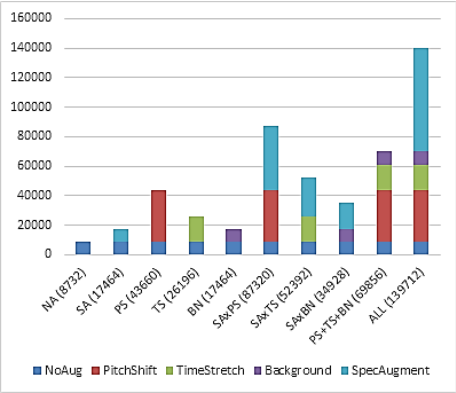


Fig. 8. The number of samples and their distribution in each of the 10 data

7 Results and Discussion

The deep learning models are trained on the 10 different data described in the previous section. The performance of the different models and augmentation methods are evaluated using their respective accuracy and F1-scores. The results of all the models are displayed in tables 2-5.

Table 2. Average accuracy and F1-score of the different augmentation methods for DNN.

Data	**NA**	**SA**	**PS**	**TS**	**BN**
Accuracy	59	58.3	**59.4**	59.3	**59.4**
F1-score	60.9	60.7	**61.4**	61.3	**61.4**
Data	**PS + TS + BN**	**SAxPS**	**SAxTS**	**SAxBN**	**ALL**
Accuracy	59.3	57.8	58.1	58.1	58.9
F1-score	61.1	60.4	60.3	60.3	60.6

Table 3. Average accuracy and F1-score of the different augmentation methods for RNN.

Data	**NA**	**SA**	**PS**	**TS**	**BN**
Accuracy	56.3	54.9	55.7	56.2	53.8
F1-score	56.3	56	56.8	57.5	53.7
Data	**PS + TS + BN**	**SAxPS**	**SAxTS**	**SAxBN**	**ALL**
Accuracy	55.9	56.9	**57.9**	56.7	57.1
F1-score	57.4	57.8	**59.6**	58.1	58.6

Table 4. Average accuracy and F1-score of the different augmentation methods for CNN.

Data	**NA**	**SA**	**PS**	**TS**	**BN**
Accuracy	76.9	78.8	77	77.4	76.8
F1-score	78.3	78.9	78.3	78.8	77.7
Data	**PS + TS + BN**	**SAxPS**	**SAxTS**	**SAxBN**	**ALL**
Accuracy	77.1	77.6	**79.2**	78.5	78.6
F1-score	78.2	78.9	**80.3**	79.4	79.6

Table 5. Average accuracy and F1-score of the different augmentation methods for CRNN.

Data	**NA**	**SA**	**PS**	**TS**	**BN**
Accuracy	69	70.6	68.9	69.1	69.3
F1-score	70.3	72	70.3	70.5	70.8
Data	**PS + TS + BN**	**SAxPS**	**SAxTS**	**SAxBN**	**ALL**
Accuracy	69.1	69.8	**72.7**	71.4	71.6
F1-score	70	72.3	**74.2**	72.9	72.7

It can be observed that the CNN architecture achieved the best results from the deep learning models tested with an average accuracy and F1-score of 79.2% and 80.3% respectively. The CRNN model achieved decent performance as well while the RNN and DNN models performed subpar.

Among the different data augmentation methods, the SAxTS, i.e. SpecAugment with time stretching, proved to be most effective in improving the performance of the CNN, RNN and CRNN models. However, the augmentation methods did not make much of a difference when subject to data to be trained on the DNN architecture. The two best performing CNN models, SAxTS-CNN and ALL-CNN, were ensembled using the model averaging technique which leading to the best performance results obtained in this study scoring an accuracy and F1-score of 79.5% and 80.6% respectively. Fig. 9 shows a boxplot representation of the accuracy values obtained by the 10-fold cross validation of the best performing data augmented deep learning models.

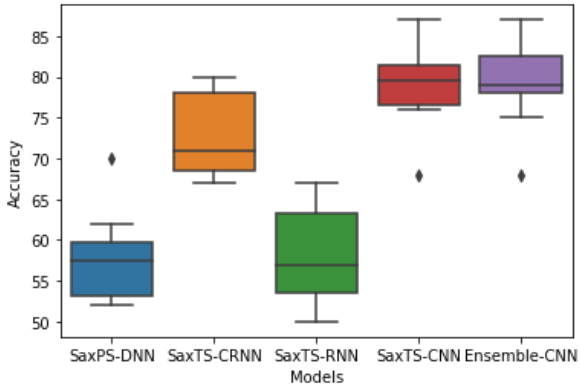
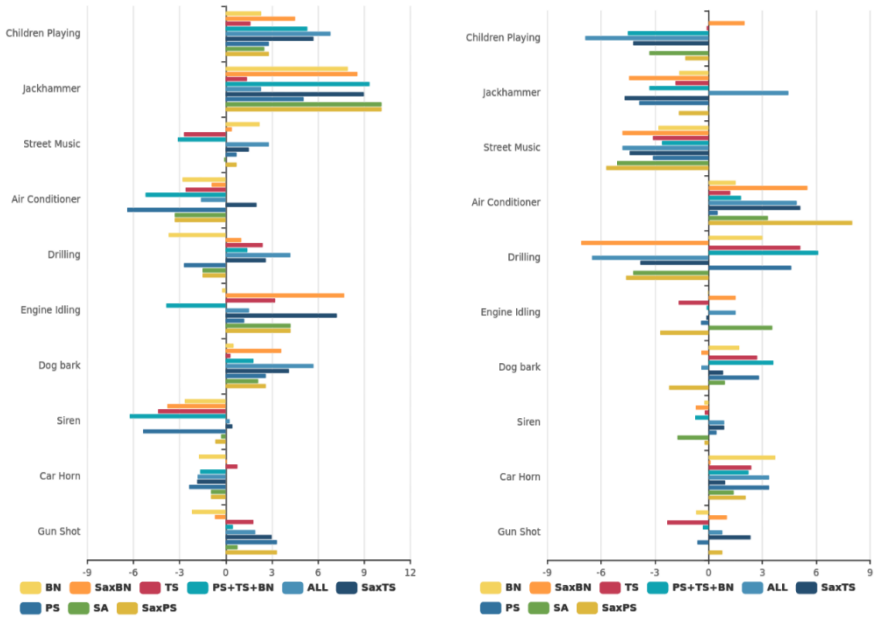


Fig. 9. Boxplot showing the accuracy values obtained by 10-fold cross validation of the best performing deep learning models and data augmentation methods.

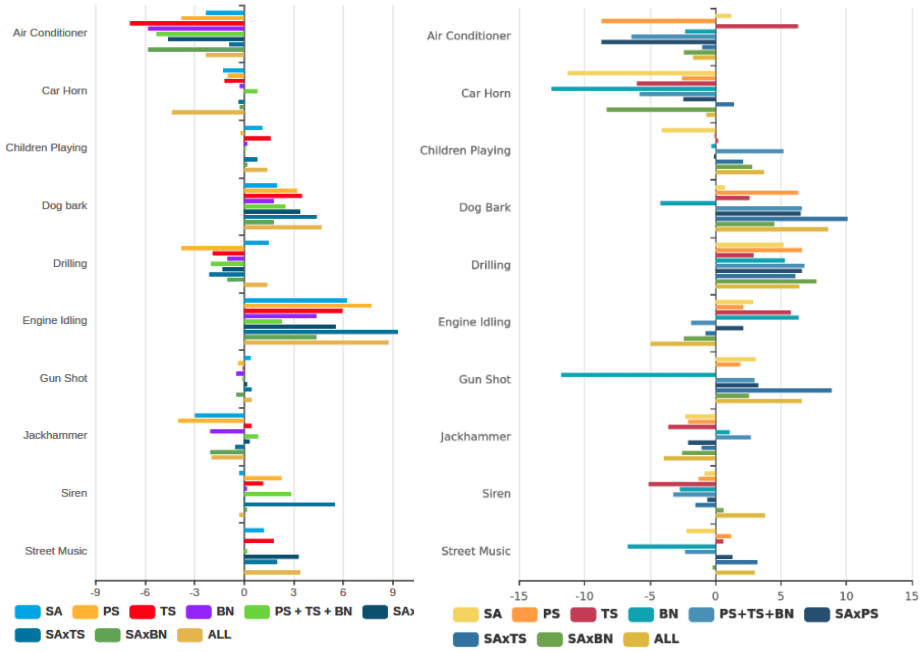
Table 4. Average accuracy and F1-score of the different augmentation methods for CNN.

Model	Accuracy
PiczacCNN [3]	73
SB-CNN (without data augmentation) [4]	73
SB-CNN (with data augmentation) [4]	79
Dai et al. [6]	71.8
Zhang et al. [7]	81.9
Sang et al. [5]	79.06
SAxTS-CNN (This paper)	79.2
SAxTS-CRNN (This paper)	72.7
SAxTS-RNN (This paper)	57.9
PS-DNN (This paper)	59.4

Fig. 10 shows the per-class accuracy difference values obtained when the deep learning models are trained and tested on different data. For computing the accuracy difference, the accuracy value obtained while training the deep learning model with the NA data is taken as the base value. It can be observed that the augmentation methods did not have a uniform effect on all the class across the different deep learning models. For instance, data augmentation did not improve performance of the CRNN model on the 'air conditioner', 'siren' and 'car horn' classes. On the CNN model, the performance on the classes 'air conditioner', 'drilling' and 'jackhammer' did not improve. While on the RNN model, the performance of the classes 'air conditioner' and 'car horn' did not improve. The models tested in this paper are compared with models from state-of-the-art models in table 5. It can be observed that the best performing CNN and ensemble model achieves competitive results.



(a) (b)



(c) (d)

Fig. 10. △ % per class accuracy for (a) CRNN (b) DNN (c) CNN (d) RNN.

8 Conclusion

This paper investigated the performance of the DNN, RNN, CNN and CRNN deep learning models on the classification of 10 classes of environmental sounds belonging to the UrbanSound8k dataset. The influence of data augmentation on the performance of these deep learning models were also studied. It can be inferred from the experiments conducted that the CNN architecture with time stretching and SpecAugment produced the best results. Moreover, upon ensembling the two best performing CNN models with the model averaging method, there was a further improvement in performance.

References

1. Stansfeld, S., Matheson, M.: Noise pollution: non-auditory effects on health. In: British Medical Bulletin 68:243-257. doi: 10.1093/bmb/ldg033 (2003).
2. Salamon, J., Jacoby, C., Bello, J. P.: A Dataset and Taxonomy for Urban Sound Research. In: Proceedings of the ACM International Conference on Multimedia - MM 14. doi: 10.1145/2647868.2655045 (2014).
3. Piczak, K. J.: Environmental sound classification with convolutional neural networks. In: IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP). https://doi.org/10.1109/mlsp.2015.7324337 (2015).
4. Salamon, J., Bello, J. P.: Deep Convolutional neural networks and data augmentation for environmental sound classification. In: IEEE Signal Processing Letters, 24(3), 279-283. https://doi.org/10.1109/lsp.2017.2657381 (2017).
5. Sang, J., Park, S., Lee, J.: Convolutional Recurrent Neural Networks for Urban Sound Classification Using Raw Waveforms. In: 2018 26th European Signal Processing Conference (EUSIPCO), doi:10.23919/eusipco.2018.8553247 (2018).
6. Dai, W., Dai, C., Qu, S., Li, J., Das, S.: Very deep convolutional neural networks for raw waveforms. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/icassp.2017.7952190 (2017).
7. Zhang, X., Zou, Y., Shi, W.: Dilated convolution neural network with LeakyReLU for environmental sound classification. In: 2017 22nd International Conference on Digital Signal Processing (DSP). https://doi.org/10.1109/icdsp.2017.8096153 (2017).
8. Shorten, C., Khoshgoftaar, T. M.: A survey on image data augmentation for deep learning. In: Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0197-0 (2019).
9. Md Shahrin, M. H.: Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks (2017).
10. Rabiner, L. R., Schafer, R.W..: Theory and Applications of Digital Speech Processing. In: NJ: Pearson, (2010).
11. Park, D. S., Chan, W., Zhang, Y., Chiu, C., Zoph, B., Cubuk, E. D., Le, Q. V.: SpecAugment: A simple data augmentation method for automatic speech recognition. In: Interspeech 2019. https://doi.org/10.21437/interspeech.2019-2680 (2019).
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. In: MIT Press (2016).
13. Li, Juncheng, et al. “A Comparison of Deep Learning Methods for Environmental Sound Detection.” 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, doi:10.1109/icassp.2017.7952131.