Classification of UrbanSound8k: A study using Convolutional Neural Network and Multiple Data Augmentation Techniques

Aamer Abdul Rahman[1][0000-0003-1189-9472] and J. Angel Arul Jothi[2][0000-0002-1773-8779]

[1] Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science, Pilani, Dubai Campus, Dubai, UAE
[2] Department of Computer Science, Birla Institute of Technology and Science, Pilani, Dubai Campus, Dubai, UAE
ar.aamer@gmail.com, angeljothi@dubai.bits-pilani.ac.in

Abstract. Audio data augmentation methods such as pitch shifting, time stretching and background noise insertion have been proven to improve the performance of deep learning models when it comes to learning discriminative spectro-temporal patterns such as environmental sounds. This work aims to classify the audio samples from the UrbanSound8k dataset using a 5-layer deep convolutional neural network (CNN). The audio samples from the dataset are transformed into their respective mel-spectrogram representation using Short-Time Fourier Transform (STFT). Different audio data augmentation techniques such as pitch shifting, time stretching, background noise and SpecAugment are applied individually and as combinations on the UrbanSound8k dataset and their influence on the performance of the CNN model is studied and compared. 10-fold cross validation is used to ensure the robustness of the results achieved. Results show that the highest accuracy and F1-score of 79.2% and 80.3% respectively is achieved when the CNN model used SpecAugment and time stretching augmented audio data.

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

1 Introduction

As various regions become urbanized and densely populated, the permeation of noises such as cars honking, busses whirring, siren wailing and the mechanical clattering from construction becomes inescapable. It has been shown in various studies that exposure to noise pollution can have a direct impact on the health and wellbeing of the population [1]. As the population continues to grow, noise pollution will continue to expand as a health risk. Understanding these problems better will lead to the creation of new potential solutions. Research in the areas of audio datasets with respect to music and speech is prevalent and continues to grow. Much progress has been made in the last few years in the area of classification of environmental sounds. Classification of urban sounds has a plethora of applications as well, such as, systems that assist those with impaired hearing, predictive maintenance in industrial environments, security and safety in smart homes and multimedia indexing and retrieval [2]. The recognition of these sounds on

mobile devices can also lead to new and exciting applications. This paper aims to study the influence of various augmentation methods like the SpecAugment, pitch shifting, time stretching and background noise insertion on the performance of a 5-layer convolutional neural network (CNN) for the classification of environmental sounds from the UrbanSound8k dataset.

The reminder of this paper is organized as follows: Section 2 covers the previous research that has been done with respect to the UrbanSound8k dataset and environmental sound classification. Section 3 provides a brief description on the UrbanSound8k dataset. Section 4 details the various augmentation methods used in this work. Section 5 explains the CNN model used. Section 6 elucidates the evaluation metrics and data preparation. The results obtained are given in Section 7. The conclusion is given in Section 8.

2 Related Work

Salamon et al. presented a taxonomy for urban sounds to enable a common framework for research and introduced a new dataset, UrbanSound8k, consisting of 10 classes of audio that spans 27 hours with 18.5 hours of annotated event occurrences [2]. The ten classes of sound include the following: “air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music”. Piczak implemented an environmental sound classification model with a convolutional neural network (CNN) architecture [3]. The datasets were preprocessed into segmented spectrograms. The CNN architecture consisted of 2 fully connected layers and 2 convolutional layers with max pooling. The proposed model achieved an accuracy of 73%.

The performance of the CNN’s on the UrbanSound8k dataset was improved by Salamon et al. using data augmentation [4]. Data augmentation was carried out on raw audio waveform before being transformed into their log-scaled Mel-spectrogram representation. The authors used audio augmentation methods such as time stretching, pitch shifting, dynamic range compression and background noise insertion. Their model achieved 73% accuracy without data augmentation and 79% accuracy with data augmentation. Sang et al. proposed a convolutional recurrent neural network model trained on raw audio waveforms for environmental sound classification [5]. The convolutional layers extracted high level features and the recurrent layers made temporal aggregations of the extracted features. Long short-term networks were used as the recurrent layers. The proposed architecture achieved an accuracy of 79.06%.

Dai et al. proposed a deep CNN architecture for environmental sound classification with 34 weight layers with no addition of fully connected layers [6]. Audio samples were used in their raw waveform as opposed to various feature extraction transformations. The model achieved decent performance and matched the accuracy of some models trained on log-scaled Mel-features. This model scored an accuracy of 71.8%. Zhang et al. used dilated convolutional neural networks and tested the effectiveness of different activation functions such as rectified linear unit (ReLU), LeakyReLU, exponential linear unit (ELU), parametric ReLU (PReLU) and Softplus [7]. The models

were tested on the UrbanSound8k, ESC-10 and CICESE datasets. 1D data augmentations were used. The results showed that the LeakyReLU activation function worked better on the UrbanSound8k and ESC-10 datasets whereas the PReLU activation function worked better on the CICESE dataset. The reasoning behind the superior performance of the LeakyReLU activation function being that it trades of network sparsity for more network information leading to better performance by the classifier. The proposed model with the LeakyReLU activation function achieved an accuracy of 81.9% on the urban sound dataset.

Though there are work in the literature that have aimed at classifying the UrbanSound8k dataset, this work is different from the previous research in the following ways: 1) This work uses a 5-layer deep CNN for learning the features and classifying the audio samples from the UrbanSound8k dataset. 2) In this work an in-depth study is conducted to analyze the effect of the SpecAugment augmentation method combined with other augmentation methods such as pitch shifting, time stretching and background noise insertion on the classification of the UrbanSound8k dataset. To the best of our knowledge, there is no previous work in the literature investigating the influence of the SpecAugment augmentation method on the UrbanSound8k dataset.

3 Dataset Description

This work uses the UrbanSound8k dataset introduced by Salamon et al. [2]. The UrbanSound8k consists of 8732 audio tracks belonging to 10 different classes like the air conditioner, car horn, children playing, dog barking, drilling, engine idling, gun shot, jackhammer, siren, and street music. These 10 classes were chosen due to the high frequency of occurrence in the city environments apart from the gun shot class which was added for variety. Each audio track spanned a duration of 4 seconds. This time length was chosen after testing showed that 4 seconds are enough for models to classify the dataset with decent accuracy. The sequences which lasted more than 4 seconds were segmented into slices of 4 seconds using a sliding window with a hop size of 2s. The total duration of the audio recording for the 8732 audio tracks was 8.7 hours. Fig. 1 shows the amplitude vs time plot of sample audio tracks from each of the 10 classes of the UrbanSound8k dataset. The matplotlib and librosa python library are used to obtain the plots.

4 Data Preprocessing

This section details the various augmentations and transformations used in this study.

1D Data Augmentation. In order to overcome the deficit of samples in the dataset, various data augmentation methods are being used that produce new samples from the existing samples by applying transformations to the samples in the dataset [8]. To produce augmentations to the raw audio dataset, methods such as pitch shifting, time

stretching and background noise insertion are used in the literature [4]. The values of pitch shifting and time stretching were chosen based on the results obtain in [4].

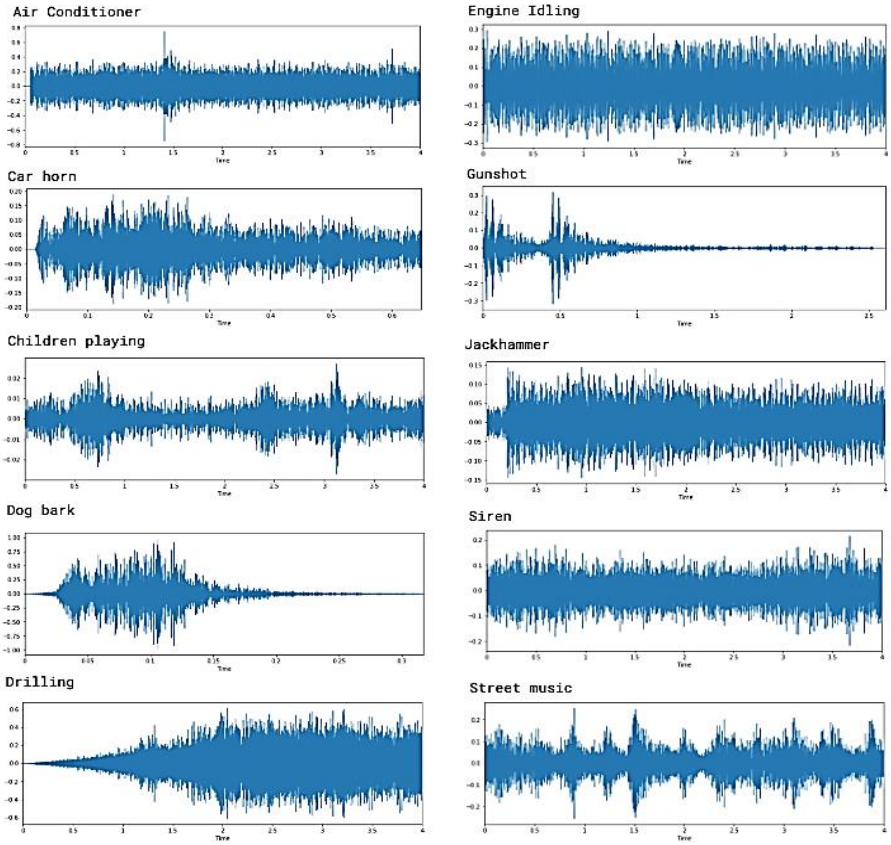


Fig. 1. Amplitude vs time plot of sample audio tracks from each of the 10 class of the UrbanSound8k dataset.

Time Stretching. The audio samples are sped up or slowed down by desired factors. The rates chosen for this study were 0.81 and 1.07.

Pitch Shifting. The pitch of the samples is increased or decreased according to specified values. Previous studies have shown that this augmentation method can improve results obtained by a classifier. The values chosen for this study were -1, -2, 1 and 2.

Background Noise. Noise was generated using the NumPy library and inserted into the audio samples in the dataset.

Audio Transformations. Classification algorithms, especially CNNs, have been shown to work significantly better on audio signals after the signals have been transformed into other representations such as Mel-spectrograms, MFCCs, chroma, tonnetz

and spectral contrast. Mel-spectrograms and MFCCs have been found to be the most effective and popular transformations used for audio classification [9]. In this work, the audio signals present in the dataset are converted to Mel-spectrogram representation.

Mel Spectrogram. Spectrogram is a visual representation of audio signal. The x-axis and the y-axis of a spectrogram represent time and the frequency respectively and the colour gradient represents the amplitude of the audio signal [10]. It can be obtained using any one of the following methods namely: The Short-Time Fourier Transforms (STFT), the Discrete Wavelet Transform (DWT), and the Cross Recurrence Plots (CRP). In this study, the librosa library is used to extract STFT Mel-spectrograms with 40 bands covering a range of up to the audible frequency of 22050 Hz using a window size of 93 ms. Fig. 2 shows the Mel-spectrogram representation of an audio sample from the class 'dog barking' from the UrbanSound8k dataset. Zero padding is added to samples whose frame count does not reach the maximum value which was calculated to be 174.

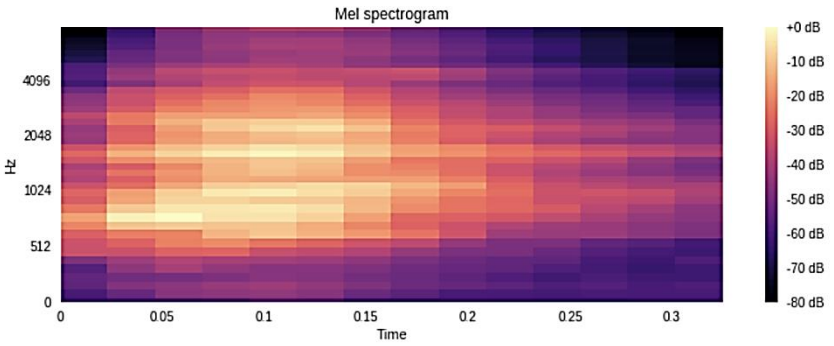


Fig. 2. Mel-Spectrogram of a dog barking.

SpecAugment (SA). Traditionally, data augmentation has been applied to audio signals before they were transformed into various visual representations such as spectrograms. Park et al. developed an augmentation method called SpecAugment [11] that can be applied directly to spectrograms by warping it in the time axis and applying blocks of consecutive horizontal (Mel-frequency channels) and vertical masks (time steps). It has been shown that SpecAugment audio signals make the classifier more robust against deformations in the time domain for speech recognition tasks.

5 Methodology

Deep Learning. Deep learning models are able to learn representations of raw data through multiple levels of abstractions due to these models being composed of multiple processing layers. Deep learning models have been able to achieve remarkable levels of performance and accuracy over the years [14][15]. Recurrent neural networks and convolutional neural networks are the most popular deep learning models prevalent in

the literature. Recurrent neural networks work well with sequential data, i.e. text and speech and convolutional neural networks have had tremendous success in processing video, images, speech and audio [12].

Convolutional Neural Network. Convolutional Neural Network is composed of multiple layers that include convolutional, pooling and usually one or more fully connected layers. The convolutional layers consist of kernels or filters that are trained to optimally extract features from an image. The pooling layer reduces the number of parameters by subsampling the feature map output by the convolutional layers. Activation functions are used in order to add non-linearity to the model so that it can perform complex tasks. The ReLU activation function has particularly gained popularity. This is because the ReLU type activation functions overcome the vanishing gradient problem faced by other activation functions such as the sigmoid and tanh activation function. In order to compute and minimize the error of the model, loss functions are used. Cross-entropy and mean squared error are two commonly used loss functions.

Proposed Model. The CNN model used in this study consists of four 2D convolutional layers with batch normalization. Batch normalization stabilizes the learning process and reduces training time by standardizing the inputs of each minibatch to a layer. The audio samples from the UrbanSound8k dataset will first go through various augmentations before being fed into the CNN model for training. The input to the CNN is the Mel-spectrogram representation of an audio signal.

The first convolutional layer (Conv2D) has 32 filters with a filter size of 3x3. It receives inputs of dimensions (128,40,174,1), where 128 is the batch size, 40 is the number of Mel-bands and 174 is the frame counts. Its outputs are of dimensions (128,38,172,32) where 32 represents the feature maps created by the filters in the first Conv2D layer. Feature maps are the outputs produced by the filters. The input batch size to the CNN model is set to 128. There is no zero padding and the stride is kept as one.

2D spatial dropout regularization with rates of 0.07 and 0.14 are added to prevent overfitting by regularizing the model by setting random units to zero. Studies by Zhang et al. showed that the leakyReLU activation function worked relatively better on the UrbanSound8k dataset, hence we decided to use the same in this study [7]. 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 second Conv2D layer has 32 filters with a filter size of 3x3 and the output with the dimensions of (128,36,170,32). A max pooling layer of size 2x2 was added after the batch normalization of the second Conv2D layer. Max pooling layers down samples the feature maps along the height and the width by choosing the highest values from each patch the layer passes over. The feature map is down sampled by the max pooling layer to the dimensions of (128,18,85,32).

The third Conv2D layer has 64 filters with a filter size of 3x3 and the output having dimensions of (128,16,83,64). The fourth Conv2D layer has 64 filters with a filter size of 3x3 with the output having dimensions of (128,14,81,64).

A global average pooling layer is added after the batch normalization layer of the fourth 2D convolutional layer without spatial dropout. The global average pooling layer down samples the collection of feature maps into a single value of dimensions (128,64) by taking the average of the values in the feature map.

A fully connected output layer with 10 nodes is added for classification of the data. The final dense 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 the CNN architecture used in this work. The CNN model is implemented using Keras.

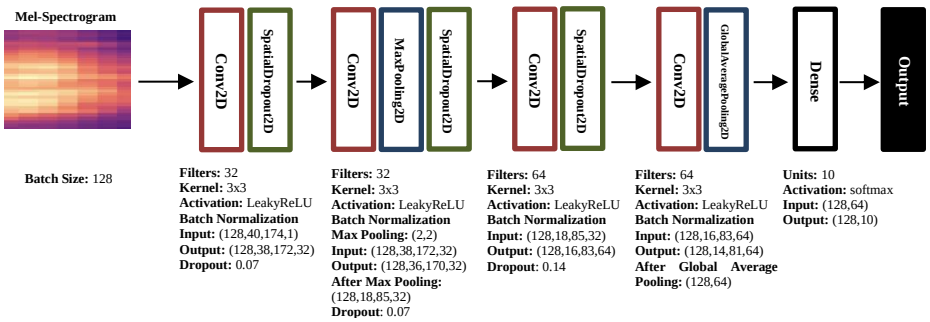


Fig. 3. The proposed CNN Architecture.

6 Evaluation metrics and Data preparation

Evaluation metrics. Let TP represent true positive, TN represent true negative, FP represent false positive and FN represent false negative. Accuracy and F1-score are used for comparing and evaluating the results of the CNN model.

To be comparable to previous work on the UrbanSound8k dataset and to ensure robust results, 10-fold cross validation is conducted. For 10-fold cross validation, the dataset is divided into 10-folds. The CNN model is trained on 9 folds and evaluated on the remaining evaluation fold. This process is repeated 10 times. Every time an evaluation fold is chosen such that it had not been chosen previously.

Data Preparation. As mentioned earlier the aim of this work is to explore various audio data augmentation techniques and their combinations in order to identify the individual or combined augmentation technique that yields the best classification accuracy value for the proposed CNN model to classify the given UrbanSound8k dataset. For this purpose, 10 different sets of data are generated using the various augmentation methods mentioned earlier to train and test the CNN model. The 10 different sets of data are denoted as NA, TS, PS, BN, SAxTS, SAxPS, SAxBN, PS+TS+BN, SA and ALL. The way in which these 10 different datasets are obtained is detailed in Table 1.

Table 1. A description of datasets with different combination of augmentations 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. 4 shows the process pipeline by which the different data are formed. The number of samples and their distribution in the 10 data have been presented in the bar graph in Fig. 5. Pitch shifting, time stretching and background noise insertion augmentations are applied to the audio samples before transforming into the Mel-spectrograms whereas SpecAugment augmentation is applied after the samples are transformed into their corresponding Mel-spectrogram representation. The CNN models trained on the different datasets are then evaluated on the evaluation fold on which no augmentation has been applied.

7 Results and Discussion

The CNN model is trained with 10 different data explained in the previous section. The average accuracy and F1-scores of the different augmentation methods are tabulated in Table 2. It can be observed from Table 2 that applying augmentations to the dataset did indeed improve the performance of the CNN model. The CNN model when trained and tested on the NA data achieved an average accuracy and F1-score of 76.9% and 78.3% respectively while the highest average accuracy and F1-score of 79.2% and 80.3% is obtained when the CNN model is trained using SAxTS data.

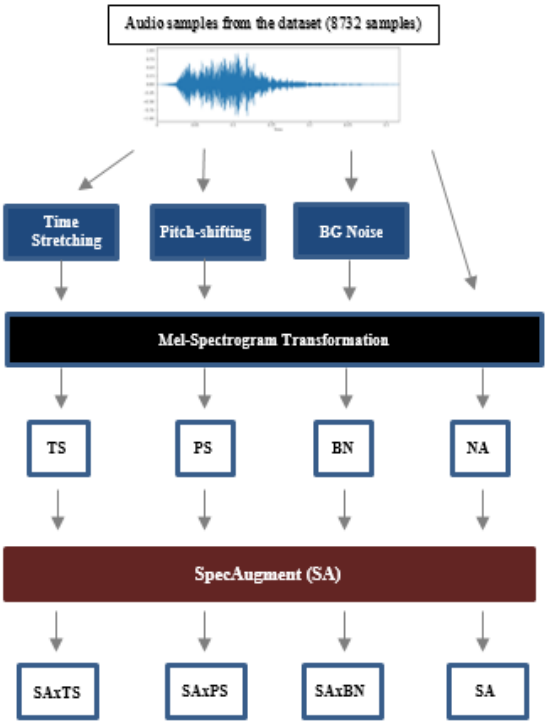


Fig. 4. Process pipeline depicting the formation of the various input data from the UrbanSound8k dataset using various audio data augmentation techniques.

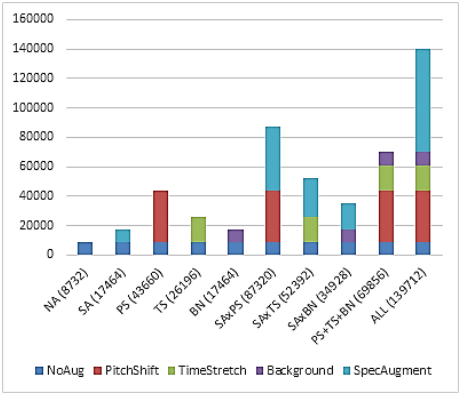


Fig. 5. The number of samples and their distribution in each of the 10 data.

Fig.6 shows the boxplot representation of the accuracy values obtained by 10-fold cross validation of the CNN model using NA and the SAxTS data. It can be observed from the figure that the proposed CNN model using SAxTS data is more robust as the corresponding boxplot covers less area. Also, it can be noted that the CNN model exhibits better accuracy values when it uses SAxTS data. It is worth noting that the performance of the CNN model did not improve as the number of samples in each dataset increased as the model trained on the ALL dataset (139,712 samples) performed worse than the model trained on the SAxTS dataset (52392 samples).

Fig. 7 (a) shows the confusion matrix obtained when the CNN model is trained and tested with SAxTS data. Fig. 7 (b) shows the difference matrix obtained by computing the difference between the confusion matrices obtained when the CNN model is trained using SAxTS and NA data. Positive values along the diagonal of the difference matrix indicate that the CNN model performs better when trained using SAxTS data. In other words, it shows that the CNN model has better performance when trained using SAxTS data when compared with the NA.

The per-class classification accuracy values of the CNN model when trained and tested with the SAxTS data are 0.98, 0.89, 0.88, 0.87, 0.85, 0.83, 0.78, 0.75, 0.69 and 0.54 for the Gun Shot, Car Horn, Dog bark, Siren, Street Music, Children Playing, Drilling, Engine Idling, Jackhammer, and Air Conditioner classes respectively. Fig. 8 shows the per-class accuracy difference values obtained when the CNN model is trained and tested on different data. For computing the accuracy difference, the accuracy value obtained while training the CNN model with the NA data is taken as the base value. It can be seen that for certain classes data augmentation greatly improved the performance of the CNN model. However, it can also be noted that in spite of data augmentation, the CNN model did not achieve good results for certain classes, such as ‘air conditioner’, ‘drilling’ and ‘jackhammer’.

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

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 3. Performance Comparison of Different Classifiers.

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 (Proposed model)	79.2

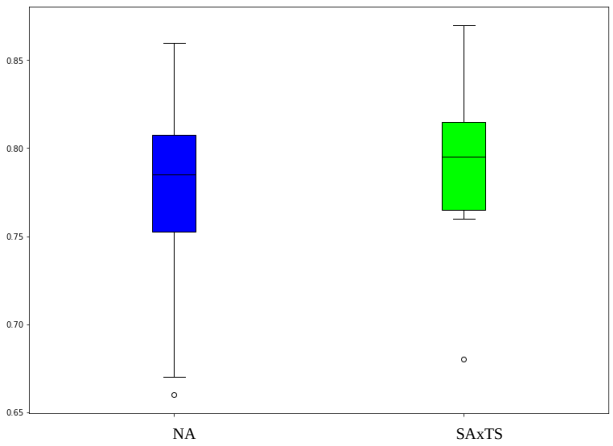
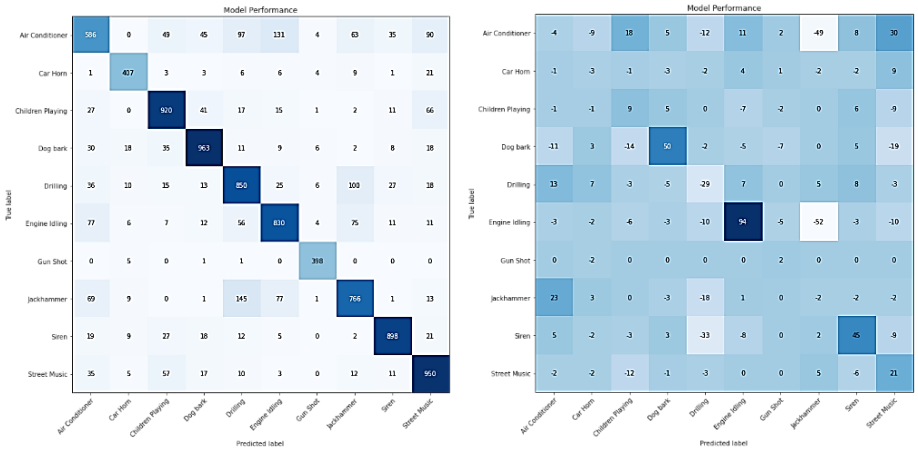


Fig. 6. Boxplot showing the accuracy values obtained by 10-fold cross validation of the CNN model using NA and SAxTS data augmentation methods.



(a) (b)

Fig. 7. (a) Confusion matrix of the CNN model when trained and tested with SAxTS data (b) Difference between confusion matrix values of the CNN when trained with SAxTS and NA.

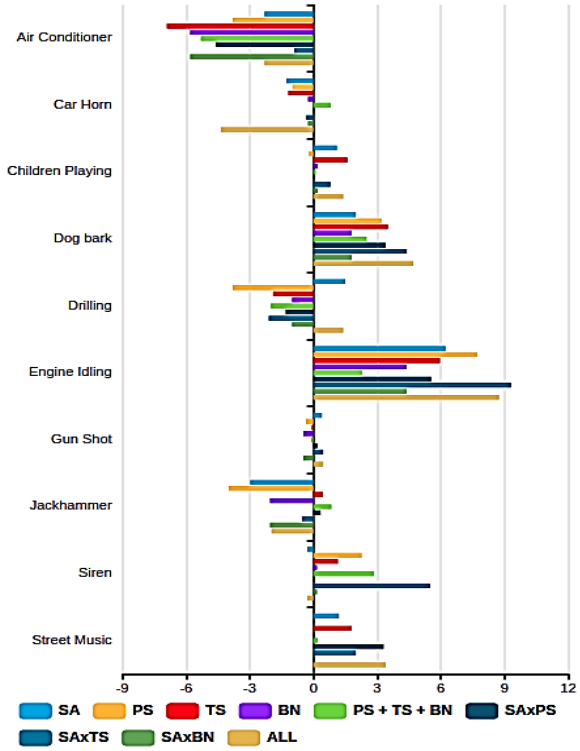


Fig. 8. Per-class accuracy difference values when the CNN model is trained and tested on different data. Note the vertical line corresponding to the value 0 represents the accuracy value of the CNN when trained and tested using the NA data.

Finally, we compared the performance of our proposed CNN model with other state-of-the-art deep learning models like PiczacCNN [3] and SB-CNN [4] with data augmentation for classifying the audio samples from the UrbanSound8k dataset. The SB-CNN used time stretching, pitch shifting, background noise insertion and dynamic range compression as the data augmentation techniques. It was observed that Piczac-CNN without data augmentation produced an average accuracy of 73% and SB-CNN with data augmentation achieved an accuracy of 79%. Further comparisons with other models, as depicted in Table 3, shows that the proposed CNN model along with SpecAugment and time stretching augmentations is able to achieve performance on par with the state-of-the-art deep models.

8 Conclusion

This paper investigated the effectiveness of various audio data augmentation techniques on the proposed CNN model for classifying the environmental sounds belonging to 10

classes from the Urbansound8k dataset. Several augmentation methods and their combinations were explored in this study. The experiments showed that the proposed CNN model achieved better results when the data was augmented using SpecAugment and time stretching data augmentation techniques. Additionally, the results also support the fact that applying augmentation to the datasets improves the performance of the CNN models.

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. McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., Nieto, O.: librosa: Audio and music signal analysis in python. In: Proceedings of the 14th python in science conference, pp. 18-25 (2015).
14. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet Classification with Deep Convolutional Neural Networks. In: Neural Information Processing Systems. 25. 10.1145/3065386 (2012).
15. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556 (2014).