Improved Deep CNN with Reduced Parameters for Automatic Identification of Environmental Sounds

Aswathy Madhu
(is with the Department of Electronics,
College of Engineering,
Trivandrum)

Suresh K
Member, IEEE
(is with the Department of Electronics,
Govt. Engineering College,
Barton Hill, Trivandrum)

***Abstract*— Deep learning techniques like Convolutional Neural Network (CNN) are steadily gaining impetus in the context of environmental sound classification. Despite their excellent performance CNN poses a challenge in terms of hardware and memory requirements due to its computationally intensive nature. Recent trends in deep learning research focus on reducing the number of parameters in the deep learning framework without performance degradation. In this paper, we put forward a novel CNN architecture with reduced parameters for automatic environmental sound classification. The proposed architecture offered a parameter reduction of 24.16% and reduced the MAC operations by 20.17%. This indicates that the proposed architecture results in reduced computational complexity during hardware deployment. The impact of parameter reduction on model accuracy is analyzed by evaluating the proposed model on a publicly available database. The results indicate that the proposed architecture outshines the state of the art approaches for automatic identification of environmental sounds.**

Keywords*— Deep Learning, Convolutional Neural Network, Environmental Sound Classification.

I. INTRODUCTION

The recent developments in Brain Computer Interface (BCI) technology has pushed its boundaries beyond mere transfer of information between human and machine. Current efforts are in the direction of improving machine perception (vision and audition). This has given rise to two related fields of research - computer vision and computer audition. Motivated by the promising results of deep learning in computer vision, researchers have introduced deep learning in computer audition. The demand for deep learning in computer audition is especially increasing in numerous applications like robotic awareness, environmental monitoring, hearing aids etc. Current solutions in intelligent applications like driverless cars are mostly vision based. But they may not perform well if exposed to conditions where visual information is compromised or is completely absent. The performance of these solutions could be improved drastically if audio information is utilized along with visual cues.

Most of the research utilizing audio information has focussed mainly on speech/music processing. Recently automatic environmental sound classification has received much attention from the research community due to its obvious socially relevant applications. To date, various signal processing as well as machine learning algorithms have addressed the problem of automatic environmental sound classification. Some of them are Non negative Matrix Factorization [1]-[3], Dictionary Learning [4], [5] etc. Recently deep learning techniques like Convolutional Neural Network (CNN) [6], [7] are steadily gaining impetus in this field. Despite their excellent performance CNN poses a challenge in terms of hardware and memory requirements due to its computationally intensive nature. Recent trends in deep learning research focus on reducing the number of parameters in the deep learning framework in order to reduce the computational cost and the network size. State of the art parameter reduction techniques in literature degrades the performance of the network in terms of classification accuracy. In this paper, we put forward a novel CNN architecture with reduced parameters for automatic environmental sound classification. The proposed architecture reduces the computational complexity, training time and enhances the classification accuracy.

II. METHOD

A. Dataset

The proposed method in this work is evaluated on UrbanSound8K dataset [8]. This dataset of 8.75 hours of field recordings contains 8732 sound extracts (≤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 8732 audio slices are prearranged into 10 folds. The audio slices are available in .wav format and the corresponding metadata file is available in .csv format..

B. Preprocessing

As observed from the literature survey, raw audio is not suitable as the input of a classifier even if it automatically learns feature representation like CNN. This is due to the high dimensionality of the audio and due to the fact that perceptually similar audio sounds may not be neighbours in terms of distance. Hence raw audio needs to be converted to a suitable representation that facilitates successful learning.

In this work, we use log scaled mel spectrogram since it allows to use the spectral information in the same way as human hearing. In this work, the raw audio files are read and processed using Librosa - a python package for music and audio analysis. The occurrences in UrbanSound8K vary in their sampling rate, bit depth and number of channels. To deal with variable sampling rate, all the occurrences were

resampled to 44,100 Hz (sampling rate of audio CDs). The fact that 20 kHz is the highest frequency audible by humans makes 44.1 kHz the logical choice for sampling rate. Given an input, it is divided into non overlapping frames of 23ms length (1024 points at a sampling rate of 44.1 kHz) which are then converted to feature space using log scaled melspectrogram. In this work, log scaled melspectrogram is computed with librosa.feature.melspectrogram using a 1024 point fft window and same hop length. The number of mel frequency bands was chosen to be 128 since this is a reasonable size that provides significant dimensionality reduction while preserving most of the original spectral characteristics. For a convenient representation of input to the CNN, 128 T-F patches are selected from the log melspectrogram of each audio signal. The patches are selected from a random initial point in time. Fig. 1 shows exemplary audio signals from the dataset and their corresponding mel spectrograms.

TABLE 1 PARAMETER REQUIREMENT FOR SB_CNN

Layer	Output Dimension	Parameters	MAC Operations
CONV 1	124×124×24	624	399776
POOL 1	31×62×24	0	0
CONV 2	27×58×48	28848	40716
POOL 2	6×29×48	0	0
CONV 3	2×25×48	57648	1300
FC 1	64×1	153664	153664
FC 2	10×1	650	650

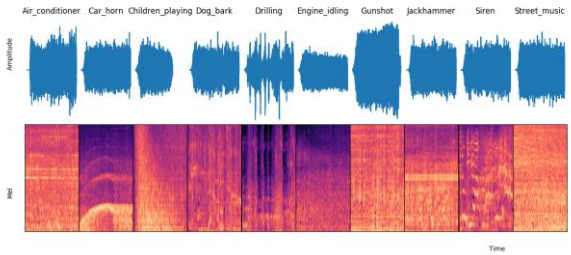

Fig 1 Exemplary audio signals from the dataset and their corresponding log mel spectrograms

C.

Given the input, the network is trained to estimate the function that maps input to a label. The proposed CNN architecture is obtained by applying two parameter reduction strategies to the CNN proposed by J. Salamon et al. (SB _CNN) [6]. SB_CNN consisted of three convolution layers of 24, 48 and 48 5×5 filters of strides (1,1) interleaved with pooling layers of size (4,2) and same strides. We replaced each 5×5 kernel with two stacked 3×3 kernels based on the observation that stacked lower dimensional kernels can extract more complex features with fewer parameters and can contribute to improved accuracy. The second strategy was to tweak the number of filters in each convolutional layer. We experimented with different values for the number of filters in each layer. It was found empirically that reducing the number of filters by a small amount in each convolution layer will not severely degrade accuracy while it contributes to significant parameter reduction. Based on these observations, the final architecture consists of three sets of stacked convolution layers containing 20, 40, and 40 3×3 filters of strides (1, 1) interleaved with pooling layers of size (4,2) and same strides. This is followed by two fully connected layers of 64 and 10 neurons as suggested in SB CNN. To reduce overfitting, dropout is introduced in the last hidden layer with a probability of 0.5. An additional max norm constraint is enforced on the weights of this layer to speed up convergence. For training, the model optimizes categorical cross entropy using Adam. A constant learning rate of 0.001 was used. The training is stopped after 50 epochs. A validation set is used for hyper parameter tuning. The CNN was implemented in Python with Keras.

III. RESULTS AND DISCUSSIONS

Table I shows the complete parameter requirement of SB_CNN. Each convolutional layer kernel has 26 parameters

(25 weights and 1 bias). Thus the parameter count of a convolutional layer is obtained by multiplying the number of parameters with the depth of the output. For example, the depth of Conv 1 layer output in SB_CNN is 24. Hence total number of parameters for Conv 1 layer is 624 obtained by multiplying 26 with 24. Following this calculation, the total number of parameters learned by the network is 241.434K. To calculate the number of MAC (Multiply and Accumulate) operations, the number of parameters for one kernel is multiplied with the product of width and height of the output. For example, the number of MAC operations for Conv 1 layer is obtained by multiplying 26 with the square of 124. Calculation for the entire network suggests that SB CNN computes 596.11K MAC operations. Since pooling layer implements a fixed operation, no learnable parameters as well as MAC operations are associated with it.

Table II shows the complete parameter requirement of the proposed architecture. Each convolutional layer kernel has 10 parameters (9 weights and 1 bias) as opposed to 26 parameters in SB_CNN. Following a similar calculation to SB_CNN, the total number of parameters for Conv 1 layer in the proposed architecture is 200 obtained by multiplying 10 with 20. The total number of parameters learned by the network is 183.1K. Similarly, the number of MAC operations for Conv 1 layer is 158760 obtained by multiplying 10 with the square of 126. The entire network computes 475.87K MAC operations. Similar to SB_CNN, no learnable parameters as well as MAC operations are associated with the pooling layer. Comparing the two architectures, it is observed that the proposed architecture has 183.1K parameters whereas SB CNN has 241.434K parameters. This shows that the proposed architecture offers a reduction of 24.16% in the number of parameters. The proposed architecture has 475.87K MAC operations as opposed to SB CNN with 596.11K MAC operations offering a reduction of 20.17% in the computational effort.

To analyse the impact of parameter reduction on accuracy, the proposed model is evaluated on the Urbansound8K dataset with 10 fold cross validation. We used mean per fold classification accuracy as the evaluation metric. A mean accuracy of 90.85% is obtained. The result is compared with the state of the art approaches for environmental sound classification evaluated on the same dataset. The results of comparison are summarized in Table III. The results indicate that the proposed architecture does not compromise on classification accuracy.

TABLE II PARAMETER REQUIREMENT FOR PROPOSED ARCHITECTURE

Layer	Output Dimension	Parameters	MAC Operations
CONV 1	126×126×20	200	158760
CONV 2	124×124×20	3620	153760
POOL 1	31×62×20	0	0
CONV 3	29×60×40	7240	17400
CONV 4	27×58×40	14440	15660
POOL 2	6×29×40	0	0
CONV 5	4×27×40	14440	1080
CONV 6	2×25×40	14440	500
FC 1	64×1	128064	128064
FC 2	10×1	650	650

TABLE III COMPARISON OF PROPOSED MODEL WITH THE STATE OF THE ART

Model	Parameters	Mean Per Fold Accuracy
SB_CNN [6]	241,434	73%
PiczakCNN [7]	29,115,370	74%
AlexNet [9]	62,378,344	90%
GoogLeNet [9]	6,797,700	93%
Proposed Model	183,094	85%

IV. CONCLUSION

In this paper, an improved CNN with reduced parameters for environmental sound classification is presented. The proposed architecture offered a parameter reduction of 24.16% and reduced the MAC operations by 20.17%. This indicates that the proposed architecture results in reduced computational complexity during hardware deployment without compromise on classification accuracy. In this work, we have not tweaked the fully connected layers of the model even though they contribute most of the parameters. Tweaking the fully connected layers for parameter reduction and replacing cascaded convolutional and pooling layers with new building units for parameter reduction are interesting avenues for future research.

REFERENCES

[1] V. Bisot,R. Serizel, S. Essid, and G. Richard, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *IEEE Intern. Conf. on Acoustics, Speech and Signal Processing* (ICASSP), Shanghai, China, 2016, pp. 6445-6449.

[2] 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 *Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 151-155.

[3] E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic model," in *IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 6450-6454.

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

[5] J. Salamon, and J. P. Bello, "Feature learning with deep scattering for urban sound analysis," in *2015 European Signal Processing Conference (EUSIPCO)*, Nice, France, 2015.

[6] 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, 2017.

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

[8] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM Intern. Conf. on Multimedia (ACMMM14)*, Orlando, FL, USA, 2014.

[9] V. Boddapati, A. Petef, J. Rasmusson and L. Lundberg, "Classifying environmental sounds using image recognition networks," *in Intern. Conf. on Knowledge Based and Intelligent Information and Engineering Systems, (KES2017)*, Marseille, France, 2017.