URBAN ACOUSTIC SCENE CLASSIFICATION USING RAW WAVEFORM CONVOLUTIONAL NEURAL NETWORKS

Technical Report

Daniele Salvati, Carlo Drioli, Gian Luca Foresti

Department of Mathematics, Computer Science and Physics
University of Udine, Italy
{daniele.salvati, carlo.drioli, gianluca.foresti}@uniud.it

ABSTRACT

We present the signal processing framework and the results obtained with the development dataset (task 1, subtask A) for the detection and classification of acoustic scenes and events (DCASE 2019) challenge. The framework for the classification of urban acoustic scenes consists of a raw waveform (RW) end-to-end computational scheme based on convolutional neural networks (CNNs). The RW-CNN operates on a time-domain signal segment of 0.5 s and consists of 5 one-dimensional convolutional layers and 3 fully connected layers. The overall classification accuracy with the development dataset of the proposed RW-CNN is 69.7%.

***Index Terms*—** Urban acoustic scene classification, end-to-end learning, raw waveform , convolutional neural network.

1. INTRODUCTION

Acoustic scene classification (ASC) aims at identifying the environment in which an audio recording has been produced. Applications that can benefit from ASC include the design of context-aware services, intelligent wearable devices, robotics navigation systems, and audio archive management [1].

Since many decades, machine learning and neural network models have been successfully employed in a wide range of audio processing applications, such as automatic speech recognition [2], audio forensic [3], music information retrieval [4], sound classification [5], array signal processing [6], and ASC [7]. Moreover, since the new computational and performance advances brought by the recent developments in the field of deep neural networks (DNNs) research, the use of this particular computation model is now being investigated in a variety of acoustic applications.

Recently, there have been various attempts to directly use the raw waveform (RW) to learn feature representation with DNNs [8, 9, 10, 11]. In [8], it is shown that the RW and mel-frequency cepstral coefficients (MFCCs) with convolutional neural network (CNN) have comparable performance for the estimation of phoneme class conditional probabilities. The ASR performance with RW and DNN has been analyzed in [9] with promising results that however present a slight worse gap with the MFCC features. In [10], it is shown that RW matches the ASR performance of log-mel filterbank energies using convolutional long short-term memory DNNs. The RW-DNN has been demonstrated to outperform a DNN that uses log-mel filterbank magnitude features using a multichannel system for ASR [11].

In this report, we present a signal processing framework based on RW-CNN model for the DCASE 2019 challenge [12]. The framework has an end-to-end computational scheme, in which the raw audio signal is used as the input of the network, without any further acoustic front-end processing. We design accurately the CNN model by analyzing the convolutional and pooling effects on raw signals. This analysis suggests that using the same convolutional filter size in all layers is the best choice. The network consists of 5 one-dimensional convolutional layers and 3 fully connected layers. We use a network scheme to handle variable-length signals by designing the CNN input using a signal frame of 0.5 s.

2. RAW WAVEFORM CNN ACOUSTIC MODEL

We aim at designing a nonlinear function $F(\cdot, \mathbf{\Theta})$ using a CNN ($\mathbf{\Theta}$ are the learned parameters during the training), which maps the input raw waveform $\mathbf{x}(t)$ of the n-the acoustic scene to the output prediction class n

$$n(t) = F(\mathbf{x}(t), \mathbf{\Theta}). \tag{1}$$

The input signal is a vector of length L

$$\mathbf{x}(t) = [x(t - L + 1), x(t - L + 2), \ldots, x(t)]. \tag{2}$$

The overall structure of the one-dimensional convolution CNN network is made of several convolution layers, followed by fully-connected layers and a classification layer. The data undergoes a filtering and activation detection step operated through the one-dimensional convolutional layer, as

$$\mathbf{h}^l = \sigma(\mathbf{w}^l * \mathbf{h}^{l-1} + b^l), \tag{3}$$

where $\mathbf{h}^l$ and $\mathbf{h}^{l-1}$ are feature maps in two consecutive layers, $\mathbf{w}^l$ is a trained kernel, b^l is a bias parameter, $\sigma(\cdot)$ is the activation function, and * denotes convolution. The rectified linear unit (ReLU) [13] is a common operation for generating the output of the convolutional layer. It computes the function $f(x) = \max(0, x)$. The bias guarantees that every node has a trainable constant value. The kernels are computed through a stochastic gradient descent method [14], which minimizes a loss function measuring the discrepancy between the CNN predictions and the targets. The loss function for classification is the cross entropy [15].

The output of the convolutional layers is then used as the input of one or more fully connected layers, in which each neuron is connected to all neurons of the previous layer. A fully connected layer multiplies the input by a weight matrix and then adds a bias vector

$$\mathbf{h}^l_{\text{FC}} = \sigma(\mathbf{W}^l \mathbf{h}^{l-1}_{\text{FC}} + \mathbf{b}^l). \tag{4}$$

To operate a dimensionality reduction and yield more robust features, a pooling layer following the ReLU layer is typically used with an averaging or maximizing operation with respect to the dimension of the feature [16]. The pooling operation, which performs a downsampling of the data, consists in dividing the input into pooling regions and computing the average or the maximum of each region. Another operation that is commonly used in CNNs to prevent overfitting is the dropout [17], which randomly sets input elements to zero with a given probability. To speed up the training of CNNs and reduce the sensitivity to network initialization, the batch normalization is used to normalize the data across a mini-batch, back-propagating the gradients through the normalization parameters [18]. The activation function used in classification layer is the softmax function [19].

The features extraction operation from the raw waveform consists mainly of two processes: convolution and max (or average) pooling. It is interesting to underline the effect of these operations on a time-domain audio signal. First, the convolution theorem states that the Fourier transform of a convolution of two signals is the point-wise product of their Fourier transforms. The convolution, which is typically computed with small kernel size, computes a filter transformation of the input with low frequency resolution. For example, assuming a sampling rate of 48 kHz and a kernel size of 32, the frequency resolution will be 1500 Hz. In this case, through the optimization algorithm, the kernel learns to emphasize the input data on 16 sub-bands with resolution of 1500 Hz. The max pooling performs down-sampling. Considering an operation with size 2 and stride 2, an input of length L will be $L/2$ after the pooling. Basically, a sub-sampling is computed reducing the sampling frequency by half. If the sampling rate is 48 kHz, it will be 24 kHz after the pooling. We can note that this operation introduces aliasing in the transformed signal. The aliasing frequency components are not lost, but they are projected in the spectrum as aliasing frequencies. After successive convolutional and pooling layers, the audio signal is again filtered and down-sampled. This results in an effective features extraction operation. The convolution emphasizes frequency components of the input with few sub-bands, and the pooling reduces the input size without loss of information since high frequency components are projected as aliasing frequencies in the spectrum. A small filtered version of the audio input will be more robust for the neural network processing if it represents the input with essential characteristics. For this reason, it is particularity important to appropriately set the kernel size taking into account that the pooling operation has to reduce and to filter the input size preserving prominent frequency components.

3. ACOUSTIC SCENE CLASSIFICATION

We use a network scheme suitable for handling variable-length signals by designing the CNN input using a short signal frame L. This strategy makes the network flexible for the audio segment to analyze due to the frame setting.

The ASC based on the raw waveform CNN acoustic model is computed using a segment of the signal composed of B frames of length L. The sequence of input vectors is $\mathbf{x}(t + bR)$, $b = 0, 1, \ldots, B - 1$, where R is the overlap step. Each input vector $\mathbf{x}(t + bR)$ of size L is processed by the CNN, which estimates B predictions for the identification.

Table 1: The architecture of the proposed RW-CNN.

l	**Layer**	**Description**	**Output Size**
1	Input	raw waveform	24000×1
2	Convolution	1×16, 32 filters	24000×32
3	Batch normalization		24000×32
4	ReLU		24000×32
5	Max pooling	1×4, stride 4	6000×32
6	Convolution	1×16, 64 filters	6000×64
7	Batch normalization		6000×64
8	ReLU		6000×64
9	Max pooling	1×4, stride 4	1500×64
10	Convolution	1×16, 128 filters	1500×128
11	Batch normalization		1500×128
12	ReLU		1500×128
13	Max pooling	1×4, stride 4	375×128
14	Convolution	1×16, 256 filters	375×256
15	Batch normalization		375×256
16	ReLU		375×256
17	Max pooling	1×4, stride 4	93×256
18	Convolution	1×16, 512 filters	93×512
19	Batch normalization		93×512
20	ReLU		93×512
21	Max pooling	1×4, stride 4	23×512
22	Fully connected	output size: 512	1×512
23	ReLU		1×512
24	Dropout	probability: 0.5	1×512
25	Fully connected	output size: 512	1×512
26	ReLU		1×512
27	Dropout	probability: 0.5	1×512
28	Fully connected	output size: 10	1×10
29	Softmax classification	predicted class probabilities	1×10

The classification is calculated as

$$\hat{n} = \underset{n}{\operatorname{argmax}} \Big[\sum_{b=0}^{B-1} p_n(t + bR) \Big], \tag{5}$$

where $p_n(t + bR)$ is the prediction output of the input vector $\mathbf{x}(t + bR)$ for the classification of the n-th acoustic scene. The B outputs whose sum correspond to the largest value provides the acoustic scene class predicted for the audio input signal.

4. CNN ARCHITECTURE

The architecture of the proposed RW-CNN consists of 5 one-dimensional convolutional layers, 3 fully connected layers, and a classification layer with softmax function. We carefully tune the kernel size and the number of filters of convolutional layers, the max pooling operation, and the output of intermediate fully connected layers to obtain an optimized performance. After each convolutional layer, the batch normalization and the activation with the ReLU are computed. Then a max pooling layer operates a dimensionality reduction. Each kernel of the convolutional layers has dimension 1×16 with stride of 1 adding zero padding to have the same size of the output as the input. In the first convolutional layer, the number of filters is 32, and it is doubled for each subsequent convolutional layer. The max pooling layers have dimensions 1×4 with stride 4. To enhance nonlinearity and to reduce overfitting, 3 fully connected layers are used with two dropout layers between

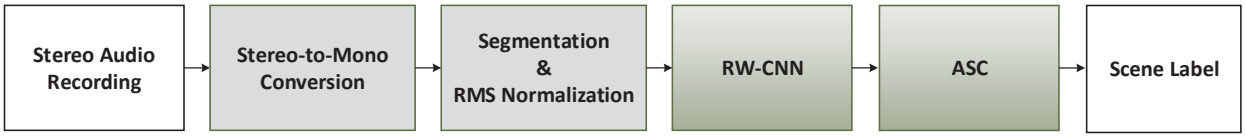

Figure 1: Schematic diagram of the proposed system.

them. The dropout layer is set with a probability of 0.5. The first and the second fully connect layers have an ouput size of 512 neurons. The last fully connect layer has $N = 10$ output neurons. The network is thus composed of 29 layers (1 input, 5 convolutional, 5 max pooling, 5 batch normalization, 7 ReLU, 3 fully connect, 2 dropout, 1 softmax classification).

In this study, we use a length frame L of 24000 samples (0.5 s) with a sampling rate of 48 kHz. The size of convolutional kernels is the same for all layers. This setting allows the increment of the filter resolution at each next convolutional layer due to the downsampling operated by the max pooling. A kernel of size 16 corresponds to a filtering operation with frequency resolution of 3000 Hz in the first convolutional layer, of 750 Hz in the second convolutional layer, and so on. The last convolutional layer has a frequency resolution of 11.72 Hz. The size of the feature maps is hence 23 samples with 512 filters. Table 1 shows the architecture of the proposed RW-CNN.

The training of the CNN is computed through a stochastic gradient descent method [14], which minimizes a cross entropy loss function measuring the discrepancy between the CNN prediction and the target. The learning rate is set to 0.001 using a mini-batch size of 32. The number of epochs is 100. The total number of learnable parameters is 9053546.

5. EXPERIMENTAL RESULTS

We present the experimental results on the development dataset (task 1, subtask A). The development dataset contains only stereo material recorded with the same device, containing 40 hours of audio, balanced between classes. The data comes from 10 of the 12 cities (the evaluation dataset contains data from all 12 cities). The training subset contains recordings from only 9 of the cities, to test the generalization properties of the systems. The training and the test subsets consist of 9185 and 4185 segments, respectively. Each segment has a duration of 10 s. The system setup is implemented with the following parameters:

- sampling rate: 48 kHz;
- CNN input size: $L = 24000$ samples;
- hop size: $R = 24000$ samples;
- number of frames for classification: $B = 20$.

The signal processing framework has been implemented using Matlab R2018a. The overall processing workflow is summarized in Figure 1. In both training and testing phase, the stereo audio input recordings are first converted into mono signals, and then each segment are divided in $B = 20$ frames of 0.5 s ($L = 24000$ samples) duration. Each segment is transformed with a root mean square (RMS) normalization. The 20 normalized frames are analyzed with the RW-CNN, and the acoustic scene classification is finally computed using equation (5).

Table 2: Classification results (accuracy %) of the development dataset (task 1, subtask A).

Scene Label	Baseline	RW-CNN
Airport	48.4	73.9
Bus	62.3	76.9
Metro	65.1	64.0
Metro station	54.5	65.1
Park	83.1	90.7
Public square	40.7	39.3
Shopping mall	59.4	49.9
Street pedestrian	60.9	72.7
Street traffic	86.7	90.5
Tram	64.0	75.7
Overall	62.5	69.7

Table 2 shows the DOA classification results of the baseline and the proposed RW-CNN for the development dataset. The baseline system is based on a CNN, where log mel-band energies are first extracted for each 10-second signal. The network consists of two CNN layers and one fully connected layer. The proposed RW-CNN has an accuracy of 69.7%, and improves of 7.2 % the baseline system.

6. CONCLUSIONS

A signal processing framework based on a RW-CNN model for the DCASE 2019 challenge has been presented. We described the framework that is based on an end-to-end computational scheme, in which the raw audio signal is used as the input of the network. We showed the results with the development dataset, and we showed that the proposed RW-CNN outperforms the baseline system with an accuracy increment of 7.2 %.

7. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] J. P. Campbell Jr., "Speaker recognition: a tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[3] X. Lin, J. Liu, and X. Kang, "Audio recapture detection with convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1480–1487, 2016.

[4] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic

music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2016.

[5] 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.

[6] D. Salvati, C. Drioli, and G. L. Foresti, "A weighted MVDR beamformer based on SVM learning for sound source localization," *Pattern Recognition Letters*, vol. 84, pp. 15–21, 2016.

[7] T. Zhang and J. Wu, "Constrained learned feature extraction for acoustic scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1216–1228, 2019.

[8] D. Palaz, R. Collobert, and M. Magimai-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Proceedings of the Conference of the International Speech Communication Association*, 2013.

[9] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Proceedings of the Conference of the International Speech Communication Association*, 2014.

[10] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proceedings of the Conference of the International Speech Communication Association*, 2015.

[11] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4624–4628.

[12] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2018)*, 2018, pp. 9–13.

[13] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the International Conference on Machine Learning*, 2010, pp. 807–814.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams., "Learning internal representations by error propagation." in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*. MIT Press, 1986, pp. 318–362.

[15] P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.

[16] M. Ranzato, F. J. Huang, Y. L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[19] J. S. Bridle, *Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition*. Springer, 1990, pp. 227–236.