End–to–end Convolutional Neural Networks for Sound Event Detection in Urban Environments

Pablo Zinemanas, Pablo Cancela, Martín Rocamora
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
{pzinemanas,pcancela,rocamora}@fing.edu.uy

***Abstract*—We present a novel approach to tackle the problem of sound event detection (SED) in urban environments using end-to-end convolutional neural networks (CNN). It consists of a 1D CNN for extracting the energy on mel–frequency bands from the audio signal based on a simple filter bank, followed by a 2D CNN for the classification task. The main goal of this two-stage architecture is to bring more interpretability to the first layers of the network and to permit their reutilization in other problems of same the domain. We present a novel model to calculate the mel–spectrogam using a neural network that outperforms an existing work, both in its simplicity and its matching performance. Also, we implement a recently proposed approach to normalize the energy of the mel–spectrogram (per channel energy normalization, PCEN) as a layer of the neural network. We show how the parameters of this normalization can be learned by the network and why this is useful for SED on urban environments. We study how the training modifies the filter bank as well as the PCEN normalization parameters. The obtained system achieves classification results that are comparable to the state–of–the–art, while decreasing the number of parameters involved.**

I. Introduction

In recent years there has been an increasing interest in the development of technologies for monitoring and diagnosing urban sound environments, which can facilitate the planning and management of the city in order to control noise pollution [1], [2]. The proposed systems are usually based on distributed sensor networks over Internet of Things (IoT) technologies, that provide sound pressure level estimates throughout the city in real time [1], [3], [4]. Based on the application of signal processing and machine learning techniques, some recent works [1], [5], [6], [7] have addressed the automatic generation of high–level descriptions of the sound environment, including the detection of sound sources. This information can help city councils to implement corrective policies or develop monitoring plans.

Environmental sound classification and detection can be tackled in different ways [8]. A possible approach, aims at identifying just the predominant sound source at each time of the audio signal. A more complex approach, called sound event detection (SED), is defined as the task of finding individual sound events, by indicating the onset time, the duration and a text label describing the type of sound.

Under real conditions, the SED problem in an urban sound environment can be very challenging. The acoustic features of each class can have a great diversity, given by the intrinsic variability of sound sources of the same type (e.g. cars) and the influence of the acoustic environment (e.g. reverberation, distance). Besides, the temporal overlapping of the sound events makes the classification task harder. There are other issues, such as the influence of the microphone's response [8].

Most works on SED use the energy on mel bands as input features, in conjunction with recurrent neural networks (RNN) [9], [10], [11], [12], convolution networks (CNN) [13], [14] or convolutional-recurrent networks (CRNN) [15], [16]. Mel–frequency Cepstral coefficients (MFCC) have also been used as features, with Gaussian mixture models [17], decision trees [18], [5], [7] and deep neural networks [19], [20], [21], [22].

Yet, to the best of our knowledge, the SED problem in urban environments has never been addressed by using end–to–end neural networks. In end–to–end neural networks the input is the raw signal (e.g. audio, image) and the output is a classification vector. They have been applied to speech recognition [23], [24], [25], speaker recognition [26], automatic music labeling [27], [28], music audio tagging [29], and automatic notes transcription [30]. But, despite the fact that end–to–end image processing has brought excellent results to the image classification task (e.g. AlexNet [31], VGG [32], and GoogLeNet [33]), the results yielded by end–to–end audio processing are not better than those of the models whose input is a time–frequency representation [34].

In end–to–end neural networks feature extraction is usually done by the first convolutional layers. Generally, these models are used as a "black box", with the goal of making the network learn the acoustic features that better discriminate the classes. However, it is possible to use domain knowledge to tailor the feature–extraction layers of the network to a particular problem. The mel–spectrogram transformation (MST) model is an example of this approach [34], in which the input is one second of the audio signal and the target output is the log–mel–spectrogram. If this model is concatenated with a neural network whose input is a time–frequency representation, it forms an end–to–end neural network. The first layers of the network can still be trained to adapt the feature extraction to a particular problem, but starting at initial condition that has proved to be effective for the problem domain. As a result, the training may also need a smaller amount of data and less training epochs. On top of this, the first layers of the network (either with the initial values or after training) could be applied to other similar tasks in audio processing.

Similarly, an acoustic trainable front–end to normalize the energy of the frequency bands of the mel–spectrogram, called per channel energy normalization (PCEN), has recently been

ISSN 2305-7254

proposed for improving the far–field speech recognition [35]. The normalization function used is differentiable, hence its parameters can be learned with gradient based training processes. A set of parameter values have been proposed based on asymptotic analysis that is a good initial condition for training [36], but no training experiments have been reported.

This work aims to apply the end–to–end approach to the SED problem using domain knowledge. To achieve this, we propose a novel scheme, named SMel, to compute the energy of the mel–spectrogram using a neural network architecture. We show that the proposed SMel scheme yields better results than those of the MST model [34]. Then, we concatenate the SMel model with a state–of–the–art CNN for urban sound event detection [13], to form the end–to–end architecture. A similar approach based on end–to–end neural networks to tackle the SED problem was proposed in [37]. It uses a learned time–frequency representations as the input of a convolutional–recurrent network. However, in that work the feature extraction network is implemented by calculating the real and the imaginary parts of the discrete Fourier transform. Besides, the focus of that work is not urban environments.

In this work, we also implement PCEN as a neural network layer and we study its applicability to the SED task. We show how the network can learn the PCEN parameter values and how the mel–frequency filter bank changes after training.

II. Calculating the log-mel-spectrogram

In this section, we present a novel model to calculate the log–scaled mel–spectrogram based on a simple mel filter–bank. In first place we present a recently published work on this topic and then we explain our proposed model. Finally, we compare both architectures. The input of both models are one–second length non–overlapping slices of the audio signal, and the output is the log–scaled mel–spectrogram.

A. MST model

MST model is a CNN architecture devised to calculate the log–scaled mel–spectrogram of an audio slice [34]. The architecture is formed by three convolutional layers of 512, 256 and N_{mels} filters respectively, where N_{mels} is the number of mel bands. Fig. 1a shows the diagram of the MST model for $N_{mels} = 128$, window length of 1024 points, a hop of 512 points, and sampling rate $f_s = 22050$ Hz.

B. Proposed model (SMel)

We propose a simpler approach, namely the SMel model, which is based on the fact that all the steps to calculate the mel–band energy are differentiable functions. Therefore, those steps can be implemented as layers of a neural network. The input of our network is a matrix whose columns are the frames of the audio signal multiplied by a Hann window. The first layer of the network is a time-distributed (TD) convolution of N_{mels} filters and it is initialized with a mel filter bank. Therefore, the output of the first layer is the result of the filter bank applied to each signal frame. In the next layers, the energy of each band is calculated by an element-wise square function; a mean value function; and a logarithmic function to convert energy values to decibels (see Fig. 1b).

The mel filter bank is formed by N_{mels} filters with triangular frequency response centered in the mel–scale frequencies and overlapped by half of their bandwidth. Therefore, the frequency response of filter l is:

$$H_l(f) = \Lambda\left(\frac{f - f_l}{\Delta f_l}\right) \text{ for } f \geq 0, \tag{1}$$

where f_l is the central frequency and $\Delta f_l = f_{l+1} - f_l$ is half bandwidth. Therefore, we design impulse responses for each filter as follows:

$$h_l(t) = 2\Delta f_l \text{sinc}^2(t\Delta f_l)\cos(2\pi f_l t)w(t), \tag{2}$$

where $w(t)$ is a Hann window.

C. Comparing models

To compare both models, we train them with the same dataset and parameters. The dataset used is the URBAN–SED, that contains audio files with urban sound events. This dataset includes 6000 files of 10 seconds for training, and 2000 files for validation and test [13]. It is devised for polyphonic urban SED and includes ten classes from mechanical (e.g. air conditioner, engine idling), human (e.g. children playing), musical (e.g. street music), and natural (e.g. dog bark) categories.

We down–sample the audio files to a sampling rate of 22050 Hz, just to decrease the computational cost, assuming this provides a sufficient bandwidth for the problem at hand. We process the audio signal in short–time windows of length $N = 1024$ samples and using a hop size of 512 samples. The target function (ground truth of the mel–spectrogram) is calculated using *librosa* [38] with 128 mel bands from 0 Hz to 11025 Hz.

To train the MST model we use the same strategy proposed by its authors [34]. To train our model it is especially important to carefully choose the learning rate because the logarithmic function has a large gradient close to zero. We use the gradient descent equation to estimate the learning rate for the worst case. As the *librosa* function used to convert power to decibels saturates on -100 dB (power equal to 10^{-10}), that is our worst case. So, we estimate the learning rate to have a small relative change on $x = 10^{-10}$.

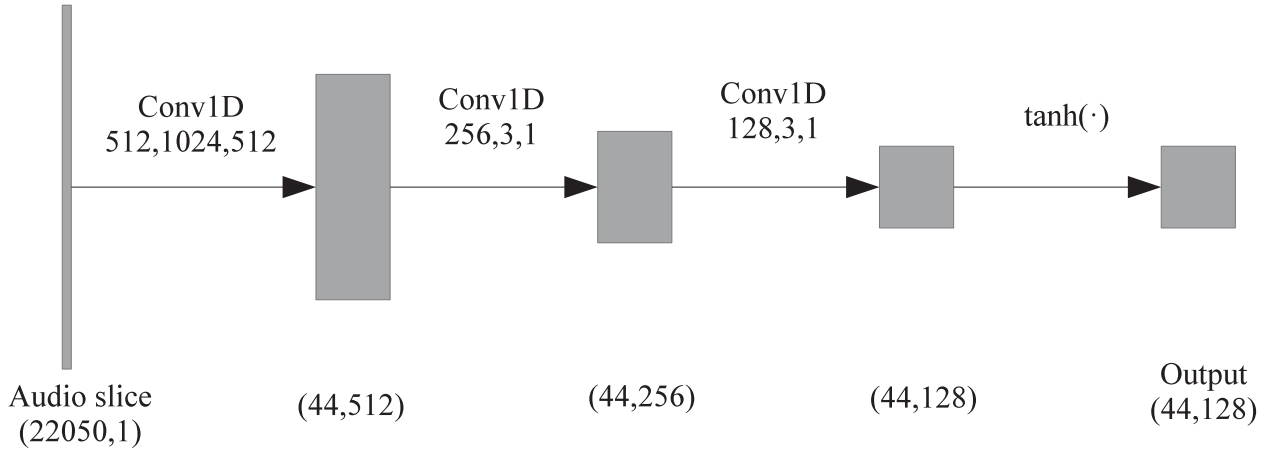
We train both models for 100 epochs using *Adam* optimizer and a mean squared loss function. Although theoretically, with the initialized parameters, SMel extracts the mel–spectrogram almost exactly, it is interesting to see the variation of the loss function for each model, as shown in Fig. 2. It is clear that the approximation of our model is better than MST. Furthermore, due to the initialization of the filters, the convergence of the proposed model is faster. Fig. 3 shows the output of each model for a randomly selected file of the validation set. Note that the output of MST model is more blurry, particularly at high frequencies.

III. Energy Normalization

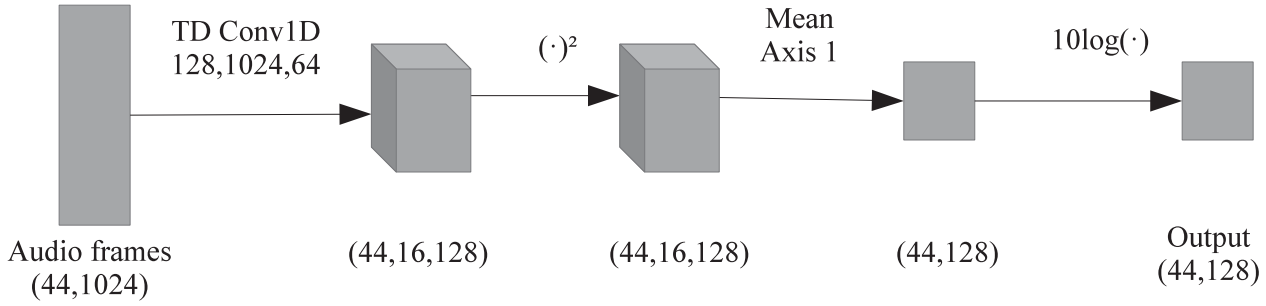
Generally, the mel–spectrogram, $E[i, l]$, is converted to a decibel scale as follows:

$$E_{dB}[i, l] = 10\log_{10}(E[i, l]), \tag{3}$$

where i is the hop index and l the frequency channel (mel filter index). This kind of normalization is very common in the SED



(a) MST model.



(b) SMel model.

Fig. 1. Block diagram of (a) MST model and (b) the proposed model for $N_{mels} = 128$, $f_s = 22050$, window length of 1024 points and hop of 512 points. In convolution layers the parameters showed are the number of filters, the size of the kernels and the stride value in that order. Above the arrows are shown the signals' dimensions

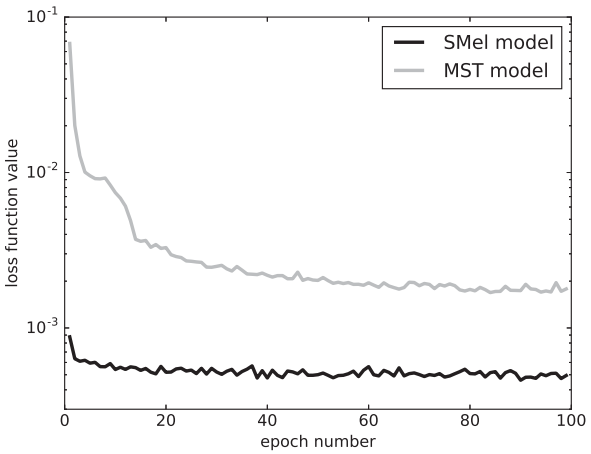


Fig. 2. Variation of the loss function value for SMel and MST in training

problem [13], [15]. A new approach, called per channel energy normalization (PCEN), was recently proposed to increase the robustness to loudness variations on speech detection systems [35]. The static logarithmic function is replaced with a dynamic range compression (DRC) and an adaptive gain control (AGC) with temporal integration. This integration is performed with a low-pass filter ϕ_T, in a temporal scale of T, as follows:

$$E_{PCEN}[i,l] = \left(\frac{E[i,l]}{(\epsilon + M[i,l])^{\alpha}} + \delta\right)^{r} - \delta^{r}, \qquad (4)$$

where $M[i,l] = (E^t * \phi_T)[i,l]$, while α, ϵ, r and δ are positive constants [36]. The low-pass filter is implemented as a first order IIR filter as follows:

$$M[i,l] = (1-s)M[i-1,l] + sE[i,l], \qquad (5)$$

where s is a smoothing coefficient [35].

Fig. 4 shows, for an example audio file from the dataset, the comparison of the logarithmic scale versus the PCEN using the parameter values suggested in [35].

Values for the PCEN parameters have been proposed according to asymptotic studies [36], but it is interesting to note that the function is differentiable, thus, those parameters could be learned by neural network models [35].

We implement PCEN with neural network layers. $M[i,l]$ is calculated using a recurrent layer that implements the IIR filter that has been proposed in [35]. The rest of the operations are implemented in a layer that has two inputs $E[i,l]$ and $M[i,l]$. The parameters of this normalization are frequency dependent (i.e. $\alpha[l]$). Fig. 5 shows the diagram of this implementation.

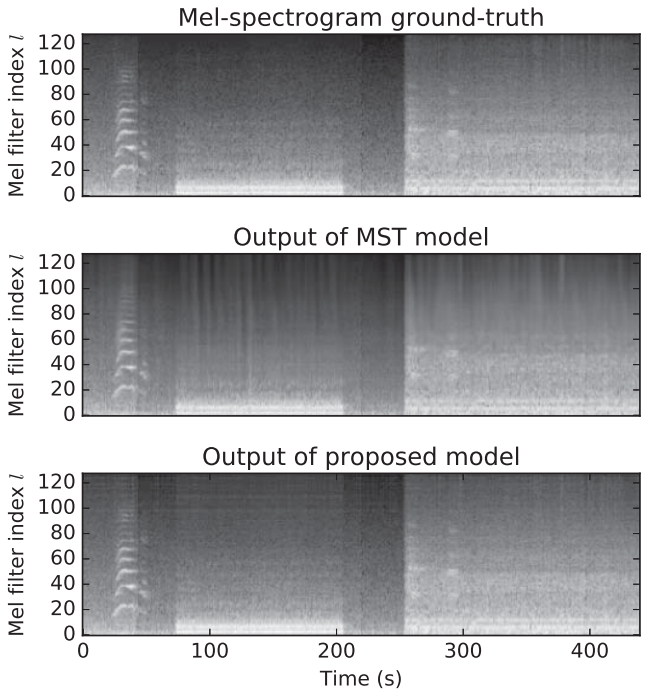


Fig. 3. The mel–spectrogam ground–truth calculated using *librosa* implementation, and the outputs of the MST and SMel models for an example file from the validation set

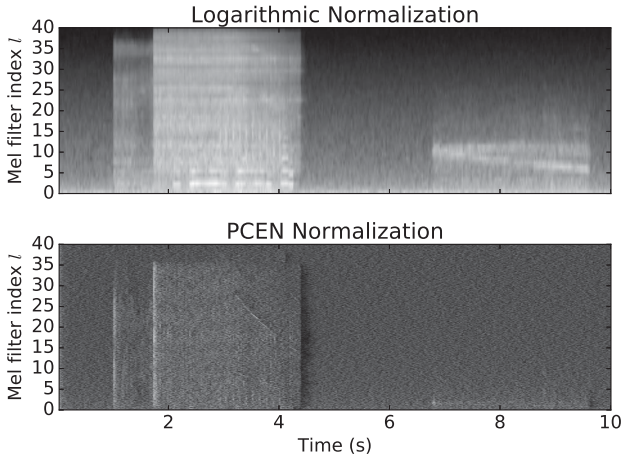


Fig. 4. Comparison of energy normalization with logarithmic function (top) and using PCEN (bottom). PCEN is calculated using *librosa* implementation with parameters $\alpha = 0.8$, $\delta = 10$, $r = 0.25$, $T = 0.06$, $\epsilon = 10^{-6}$.

In the next section, we show the benefits of using the SMel model and the PCEN normalization for sound event detection in urban environments.

IV. Experiments and Results

In order to show how it is possible to use the proposed model, we concatenate it to a network that uses the mel–spectrogram as input. We use the CNN described on URBAN–SED article [13] as the baseline. This network has three 2D convolutional layers followed by three fully–connected layers. The final layer is a sigmoid of 10 units that perform the classification task.

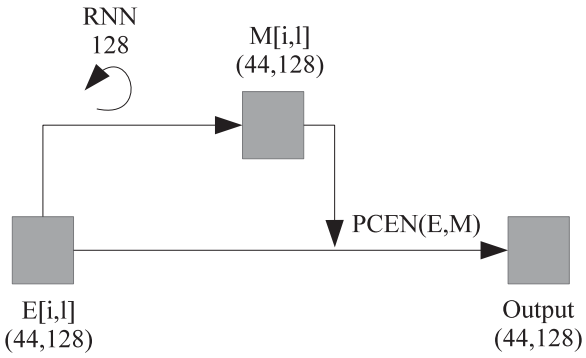


Fig. 5. Diagram of PCEN implementation with neural network

We train three networks: (CNN) the baseline (mel–spectrogram from *librosa* as input) [13]; (MST+CNN) the MST model concatenated to the baseline CNN; and (SMel+CNN) the proposed model also concatenated to the CNN. All models are implemented using *keras* [39] library with Tensor Flow as backend. This scheme implies that the temporal resolution of the detected sound events is one second.

A. Training strategy

In order to train the CNN, we use the same strategy as presented in [13] and the parameter values presented in Section III. To train the MST+CNN and SMel+CNN networks we use a strategy inspired by Branch Training for hierarchical classification [40]. The training process is performed with two loss functions; mean squared for mel–spectrogram (l_0) and binary cross entropy for classification (l_1). The final loss function (l) is a weighted sum of the two losses:

$$l = w_0 l_0 + w_1 l_1, \tag{6}$$

where $[w_0, w_1]$ is a pair of weights. We set w_0 with a small value, as a way to regularize the mel–spectrogram training, and $w_1 = 1 - w_0$.

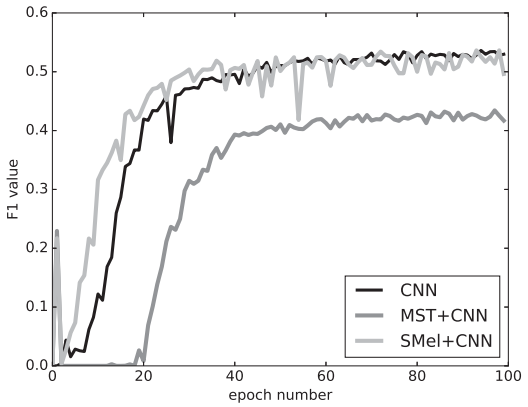


Fig. 6. Variation of the F1 value in the validation set for each model

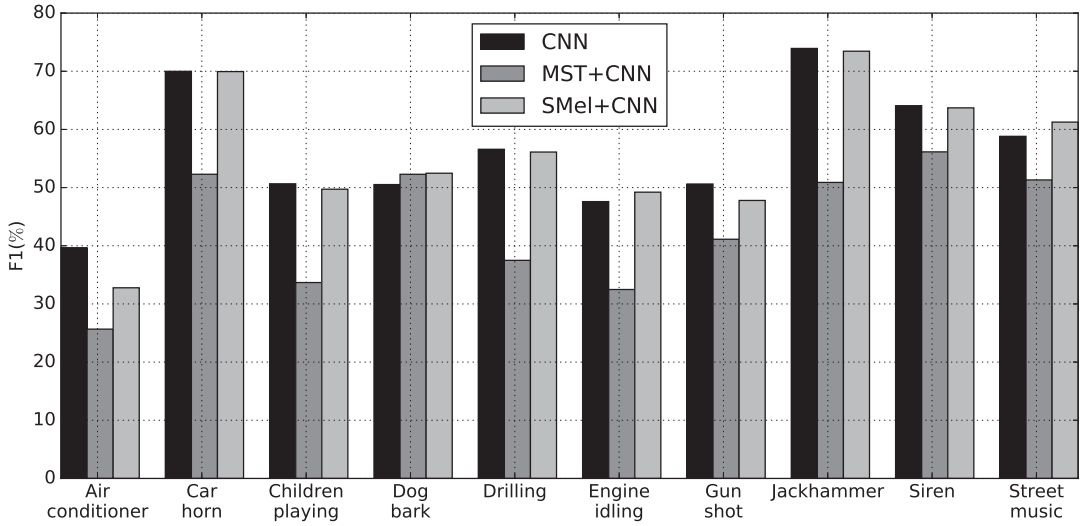


Fig. 7. F1 values per class on test set for each model

B. Evaluation metrics

The F–score ($F1$) and the error rate (ER) are the performance measures usually reported for SED systems [13], [9], [15], compared with ground-truth annotations on one–second length segments. For each segment, the False Positive (FP), False Negative (FN) and True Positive (TP) rates are calculated; and then the precision and recall are computed as follows:

$$P = \frac{TP}{TP + FP},\ R = \frac{TP}{TP + FN}.$$

$F1$ is the harmonic mean of P and R:

$$F1 = \frac{2PR}{P + R}. \tag{7}$$

The ER is calculated in terms of insertions (I), deletions (D) and substitutions (S). A substitution is defined as the case when the system detects an event on a segment, but with the wrong label. This is equivalent to have a FP and a FN in the same segment. After counting substitutions, the rest of FP are counted as insertions and the rest of FN as deletions. The ER measure is calculated as the integration of this values on the total number of segments K, as follows:

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)}, \tag{8}$$

where $N(k)$ is the number of active classes in the ground–truth at the segment k [17], [41].

C. Classification results

We train the three networks for 100 epochs using the strategies proposed previously and using *Adam* optimizer. Fig. 6 shows the variations of $F1$ value on the validation set for the three networks. We save the network's weights on the epoch for which the F1 value on the validation set is maximum. Fig. 7 shows the best attained F1 values per class on the test set for each model. Table I shows the overall detection results.

TABLE I. RESULTS OF $F1$ AND ER VALUES ON THE TEST SET (IN BOLD THE BEST RESULTS).

Network	$F1(\%)$	ER
CNN	56	0.53
MST+CNN	43	0.61
SMel+CNN	**57**	**0.50**

D. Energy normalization

In this section, we present the experiments related to PCEN. The SMel model whose outputs are normalized with PCEN is called SMel_P. Firstly, the CNN is trained with the input data also normalized with PCEN using the *librosa* implementation and the same parameters used in section III. This model is called CNN_P. Then, the concatenated network SMel_P+CNN_P is trained with the same loss function of equation (6). In this experiment, we study how do the filters $H_l[k]$ change, thus w_0 is set to zero to avoid regularization. PCEN parameters are initialized with the same values as in CNN_P. Fig. 8 shows the parameter values after the training process. The only parameter that changed significantly is r which defines how the DRC works. For high frequencies, the r value decreases, and the compression increases [36]. Analogously, the compression is small for low frequencies. This could be because in this dataset most meaningful information for classification is in the lower frequencies.

TABLE II. RESULTS OF $F1$ AND ER ON TEST SET FOR NETWORKS CNN, CNN_P AND SMEL_P+CNN_P.

Network	$F1(\%)$	ER
CNN	56	0.53
CNN_P	54	0.56
SMel_P+CNN_P	51	0.60

Table II shows the performance results for CNN, CNN_P and SMel_P+CNN_P networks. Note that PCEN normalization does not improve the performance of CNN. Also, training SMel_P in conjunction with CNN_P, does not improve the

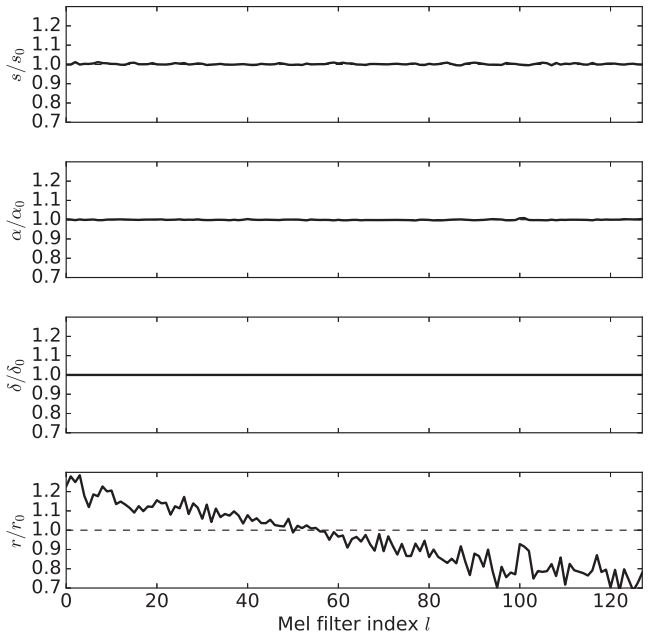


Fig. 8. PCEN parameter as a function of the channel (band frequency l) referenced to initial values trained with SMel_P+CNN_P. Dash lines mark initial values

results, but it is interesting to see in the Fig. 9 how the $H_l[k]$ filters change.

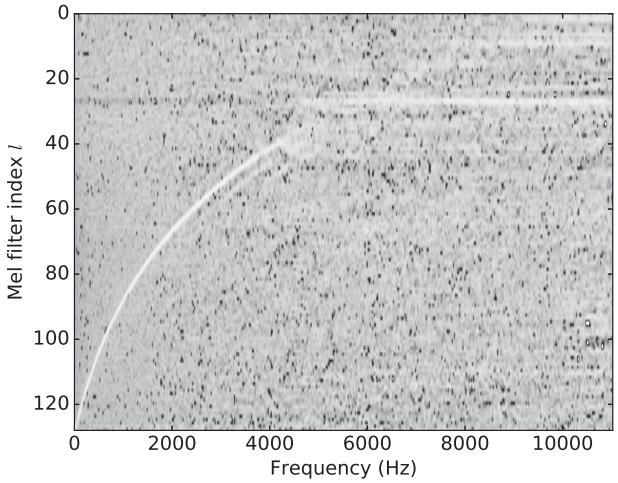


Fig. 9. $H_l[k]$ filters learned by SMel P+CNN P network

For high frequencies, the result seems to be very noisy, in particular above 4000 Hz. This suggests that the information in this frequency band does not contribute substantially in the classification. It is interesting to note that this result is similar to the one reported in [37]. To corroborate this finding, we explored re–sampling the dataset at 8000 Hz and calculating the energy on 64 mel bands in the range from 0 to 4000 Hz. Notice that effectively the results of ER and F1 do not change significantly working with a sampling rate of 8000 Hz, while the number of parameters decreases considerably (see Table III). This result could have been obtained by other means, for instance, changing the dataset sampling rate by trial and error. However, the proposed feature extraction network can learn task specific filters when faced to other problems.

TABLE III. RESULTS OF $F1$ AND ER TEST SET AND NUMBER OF PARAMETERS OF NETWORKS CNN_P AND SMEL_P+CNN_P TRAINED WITH URBAN-SED DATASET RE-SAMPLED TO 22050 HZ AND 8000 HZ.

$f_s(Hz)$	Red	$F1(\%)$	ER	# params (M)
22050	CNN_P	54	0.56	~ 2.48
	SMel_P+CNN_P	51	0.60	~ 2.55
8000	CNN_P	52	0.55	~ 0.99
	SMel_P+CNN_P	49	0.56	~ 1.01

V. CONCLUSION AND FUTURE WORK

This work presents a novel approach for sound event detection in urban environments using end–to–end neural networks, that is obtained by concatenating two networks: one for feature extraction and another one for classification. This two–stage architecture facilitates the introduction of domain knowledge and improves the interpretability of what the networks learn.

For the first network, that is devised to extract the mel–spectrogram, we propose a simple approach based on a mel–frequency filterbank. We show that the proposed model achieves better results (smaller loss function value) than those of the recently proposed MST model. We also show that the classification results of the concatenated end–to–end network are similar to those of a state–of–the–art CNN. However, the proposed model offers a better interpretability regarding the output of the first layers of the network.

Also, we implement the recently proposed PCEN energy normalization as a neural network layer and we train its parameters in conjunction with those of the rest of the network. We find that the only parameter that significantly changes in the training is r, that determines the amount of dynamic range compression of the signal.

We also study how the filters of the first layer change with the training. The results suggest that for the URBAN–SED dataset, the most relevant information is below 4000 Hz and this is confirmed by obtaining similar results for a 8000 Hz sub-sampled version of the dataset. We conclude that models with less parameters could be used.

As future work we aim to apply these end–to–end architectures to other audio related problems. Also, further comparisons with state–of–the–art systems will be conducted.

REFERENCES

[1] J. P. Bello, C. Mydlarz, and J. Salamon, *Computational Analyis of Sound Scenes and Events*. Springer, 2017, ch. 13 Sound Analysis in Smart Cities.

[2] D. K. Daniel Steele and C. Guastavino, "The sensor city initiative: cognitive sensors for soundscape transformations," in *Geoinformatics for City Transformations*. Technical University of Ostrava, January 2013, pp. 243–253.

[3] M. B. Charlie Mydlarz, Charles Shamoon and M. Pimpinella, "The design and calibration of low cost urban acoustic sensing devices," in *EuroNoise 2015*. European Acoustics Association, 2015, pp. 2345–2350.

[4] C. Mydlarz, S. Nacach, A. Roginska, T. H. Park, E. Rosenthal, and M. Temple, "The implementation of mems microphones for urban sound sensing," in *Audio Engineering Society Convention 137*, Oct 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=17466

[5] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 171–175.

[6] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22st ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014.

[7] J. Salamon and J. P. Bello, "Feature learning with deep scattering for urban sound analysis," in *2015 European Signal Processing Conference (EUSIPCO)*, 2015, pp. 724–728.

[8] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analyis of Sound Scenes and Events*. Springer, 2017, ch. 1 Introduction to Sound Scene and Event Analysis.

[9] S. Adavanne, G. Parascandolo, P. Pertila, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," in *Detection and Classification of Acoustic Scenes and Events 2016*, 2016.

[10] T. H. Vu and J.-C. Wang, "Acoustic scene and event recognition using recurrent neural networks," in *Detection and Classification of Acoustic Scenes and Events 2016*, 2016.

[11] A. Gorin, N. Makhazhanov, and N. Shmyrev, "Dcase 2016 sound event detection system based on convolutional neural network," in *Detection and Classification of Acoustic Scenes and Events 2016*, 2016.

[12] J. Zhou, "Sound event detection in multichannel audio LSTM network," DCASE2017 Challenge, Tech. Rep., September 2017.

[13] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello., "Scaper: A library for soundscape synthesis and augmentation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, october 2017.

[14] I.-Y. Jeong, S. Lee, Y. Han, and K. Lee, "Audio event detection using multiple-input convolutional neural network," DCASE2017 Challenge, Tech. Rep., September 2017.

[15] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *Transactions on Audio, Speech and Language Processing: Special issue on Sound Scene and Event Analysis*, vol. 25, no. 6, pp. 1291–1303, June 2017.

[16] S. Adavanne and T. Virtanen, "A report on sound event detection with different binaural features," in *Detection and Classification of Acoustic Scenes and Events 2017*, 2017.

[17] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *In 24rd European Signal Processing Conference 2016 (EUSIPCO 2016), Budapest, Hungary*, 2016.

[18] B. Elizalde, A. Kumar, A. Shah, R. Badlani, E. Vincent, B. Raj, and I. Lane, "Experimentation on the dcase challenge 2016: Task 1 - acoustic scene classification and task 3 - sound event detection in real life audio," in *Detection and Classification of Acoustic Scenes and Events 2016*, 2016.

[19] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, "Deep neural network baseline for dcase challenge 2016," in *Detection and Classification of Acoustic Scenes and Events 2016*, 2016.

[20] D. Wei, J. Li, P. Pham, S. Das, S. Qu, and F. Metze, "Sound Event Detection for Real Life Audio DCASE Challenge," in *Detection and Classification of Acoustic Scenes and Events 2016*, 2016.

[21] C.-H. Wang, J.-K. You, and Y.-W. Liu, "Sound event detection from real-life audio by training a long short-term memory network with mono and stereo features," DCASE2017 Challenge, Tech. Rep., September 2017.

[22] R. Lu and Z. Duan, "Bidirectional GRU for sound event detection," DCASE2017 Challenge, Tech. Rep., September 2017.

[23] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for lvcsr," in *INTERSPEECH*, 2014.

[24] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux, "End-to-end speech recognition from the raw waveform," in *Interspeech*, 2018.

[25] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, B. Li, E. Variani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, *Raw Multichannel Processing Using Deep Neural Networks*. Springer International Publishing, 2017.

[26] G. Valenti, A. Daniel, and N. Evans, "End-to-end automatic speaker verification with evolving recurrent neural networks," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 335–341.

[27] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, 2018.

[28] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 05 2014, pp. 6964–6968.

[29] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *19th ISMIR Conference*, September 2018.

[30] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *5th International Conference on Learning Representations - ICLR 2017*, 2017.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Oct 2015.

[34] T. M. S. Tax, J. L. D. Antich, H. Purwins, and L. Maaløe, "Utilizing domain knowledge in end-to-end audio processing," in *31st Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017.

[35] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017.

[36] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, "Per-Channel Energy Normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, Jan. 2019.

[37] E. Cakr and T. Virtanen, "End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input," in *International Joint Conference on Neural Networks (IJCNN)*, 2018.

[38] B. McFee, M. McVicar, C. Raffel, D. Liang, O. Nieto, E. Battenberg, J. Moore, D. Ellis, R. Yamamoto, R. Bittner, D. Repetto, P. Viktorin, J. F. Santos, and A. Holovaty, "librosa: 0.4.1," Oct. 2015. [Online]. Available: https://doi.org/10.5281/zenodo.32193

[39] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[40] X. Zhu and M. Bain, "B-CNN: branch convolutional neural network for hierarchical classification," 2017, unpublished. [Online]. Available: http://arxiv.org/abs/1709.09890

[41] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.