doi:10.20944/preprints201811.0509.v1

Article

Recognition of urban sound events using deep context-aware feature extractors and handcrafted features

Theodoros Giannakopoulos [1,*] and Stavros Perantonis [2]

[1] NCSR Demokritos; tyiannak@gmail.com
[2] NCSR Demokritos; sper@iit.demokritos.gr
* Correspondence: tyiannak@gmail.com; Tel.: +30 6977157480

Abstract: This paper proposes a method for recognizing audio events in urban environments that combines handcrafted audio features with a deep learning architectural scheme (Convolutional Neural Networks, CNNs), which has been trained to distinguish between different audio context classes. The core idea is to use the CNNs as a method to extract context-aware deep audio features that can offer supplementary feature representations to any soundscape analysis classification task. Towards this end, the CNN is trained on a database of audio samples which are annotated in terms of their respective "scene" (e.g. train, street, park), and then it is combined with handcrafted audio features in an early fusion approach, in order to recognize the audio event of an unknown audio recording. Detailed experimentation proves that the proposed context-aware deep learning scheme, when combined with the typical handcrafted features, leads to a significant performance boosting in terms of classification accuracy. The main contribution of this work is the demonstration that transferring audio contextual knowledge using CNNs as feature extractors can significantly improve the performance of the audio classifier, without need for CNN training (a rather demanding process that requires huge datasets and complex data augmentation procedures).

Keywords: audio recognition; context-aware; deep learning

1. Introduction

Soundscape audio recordings capture the sonic environment of a particular time and location and it can be conceived as an "auditory landscape", either in an individual or common level. With the advances in audio signal processing and machine learning, it has become possible to automatically predict the content, the context [1–4] and even the quality [5] of the soundscape recordings. Automatic recognition of soundscapes is rather important in many emerging applications, such as surveillance, urban soundscape monitoring and noise source identification.

An important problem in automatic soundscape classification is the diversity between different datasets and benchmarks, in terms of class taxonomy, dataset size and audio signal quality. In this work, we utilize the ability of deep neural networks to learn patterns in large datasets, capturing both spectral and temporal relations in audio signals. Towards this end, a deep learning network architecture is trained to distinguish between contextual classes (e.g., park, restaurant, library, metro station, etc). Then, this network is used as a supervised audio feature extractor in an urban classification task of urban audio events and in combination with handcrafted audio features. Extensive experimentation proves that this transfer of knowledge from an audio context domain, using deep neural networks, can boost the performance of the soundscape classification procedure, when handcrafted features are combined with the deep context-aware features.

The main contribution of this work is the experimental proof that audio contextual knowledge can be "transferred" through a CNN, which is trained in a scene-specific dataset, and that this scheme can be used to boost the performance of audio event classification based on typical handcrafted audio

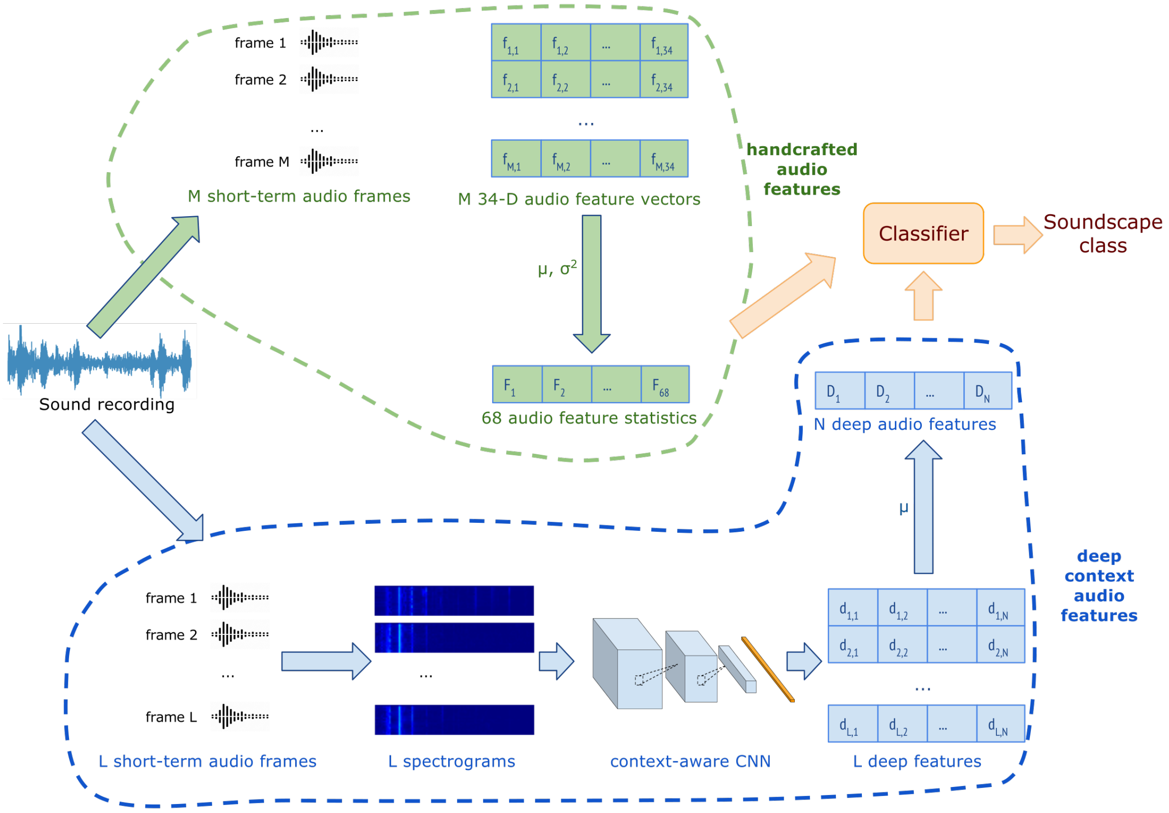
frame 1
frame 2
...
frame M
M short-term audio frames
$f_{1,1}$ $f_{1,2}$ ... $f_{1,34}$
$f_{2,1}$ $f_{2,2}$ ... $f_{2,34}$
...
$f_{M,1}$ $f_{M,2}$ ... $f_{M,34}$
M 34-D audio feature vectors
μ, σ^2
F_1 F_2 ... F_{68}
68 audio feature statistics
handcrafted audio features
Sound recording
Classifier
Soundscape class
D_1 D_2 ... D_N
N deep audio features
μ
deep context audio features
frame 1
frame 2
...
frame L
L short-term audio frames
...
L spectrograms
context-aware CNN
$d_{1,1}$ $d_{1,2}$ ... $d_{1,N}$
$d_{2,1}$ $d_{2,2}$ ... $d_{2,N}$
...
$d_{L,1}$ $d_{L,2}$ ... $d_{L,N}$
L deep features

Figure 1. Conceptual diagram of the proposed method. Two separate steps are adopted during in the analysis pipeline: hand-crafted-audio features are computed on the raw signal, as well as a supervised CNN trained to distinguish between different context classes is used as a feature extractor.

features. This has been experimentally demonstrated using two widely adopted benchmarks, even with a baseline classification approach and a standard early-fusion feature combination scheme.

2. GENERAL SCHEME

As illustrated in the general conceptual diagram of Figure 1, we propose classifying unknown soundscape recordings using two distinct feature representation steps:

- **Handcrafted audio Features (HaF)** According to this widely-adopted approach, each signal is represented by a series of statistics computed over short-term audio features, either from the time or the frequency domain, such as signal energy, zero crossing rate and the spectral centroid. These features aim to represent the audio signal in a space that is discriminative with regards to the involved audio classes. This baseline audio representation methodology is described in more detail in Section 3.
- **Context-aware deep Features (CadF)** A supervised convolutional neural network is trained to discriminate between different audio urban context classes (such as park, restaurant, etc), based on spectrograms of short-term segments. The output of the last fully connected layer of this network is used as feature extractor in the initial soundscape classification task. This methodology is described in detail in Section 4.

The two different audio feature representations are then combined in an early-fusion scheme and classified using a standard Support Vector Machine classifier. The main idea behind this feature combination procedure is the fact that the Context-aware deep Features (CadF) are extracted based on a deep neural network that has been trained to distinguish between different context classes and can therefore introduce a diverse and complementary content representation to the Handcrafted audio

Table 1. Handcrafted short-term audio features. In total, 34 audio features extracted from the time, spectral and cepstral domains are computed per short-term frame. This leads to a series of 34-dimensional feature vectors for each audio segment. At a second stage segment-level statistics (mean and standard deviation) are computed for the whole audio segment, leading to a 68-dimensional representation.

Index	Name	Description
1	Zero Crossing Rate	Rate of sign-changes of the signal during the duration of a particular frame
2	Energy	(normalized) sum of squares of the signal values
3	Entropy of Energy	Entropy of sub-frames' normalized energies
4	Spectral Centroid	Center of gravity of the spectrum
5	Spectral Spread	Second central moment of the spectrum
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames
7	Spectral Flux	Squared diffs between the normalized magnitudes of the spectra of the two successive frames
8	Spectral Rolloff	Frequency below which 90% of the magnitude distribution of the spectrum is concentrated
9-21	MFCCs	Mel Frequency Cepstral Coefficients
22-33	Chroma Vector	A 12-element representation of the spectral energy using a semitone spacing
34	Chroma Deviation	Standard deviation of the 12 chroma coefficients

Features (HaF). In various machine learning applications, it has been proven that combining diverse and complementary features (or individual classification decisions) in meta-classification schemes, leads to classification performance boosting [6]. This is also proven experimentally in this work, as described later in the experiments section.

3. Audio classification based on handcrafted audio features

As a baseline methodology of automatic characterization of audio segments, a short-term feature extraction workflow of widely-adopted *handcrafted* audio features has been adopted. Traditional audio classification, regression and segmentation utilizes handcrafted features in order to represent the corresponding audio signals in a feature space that is able to discriminate an unknown sample between the involved audio classes. This process of extracting features from the initial audio signal is therefore essential in all audio analysis methodologies.

In order to achieve audio feature extraction, each audio signal is first divided to either overlapping or non-overlapping short-term windows (frames). Widely accepted short-term window sizes are 20 to 100 ms. Additionally, a widely adopted methodology in audio analysis is the processing of the feature sequence on a "mid-term basis", according to which the audio signal is first divided into mid-term windows (segments), which can be either overlapping or non-overlapping. For each segment, the short-term processing process, described above, is carried out and the feature sequence from each mid-term segment, is used for computing feature statistics (e.g. the average value of the zero crossing rate). Therefore, each mid-term segment is represented by a set of statistics. Typical values of the mid-term segment size can be 1 to 10 seconds [7], [8], [9].

Table 1 shows the adopted handcrafted audio features. In this work, two mid-term statistics have been adopted, namely the average value and the standard deviation of the respective short-term features. This means that the final signal representation using these handcrafted feature statistics is a $34 \times 2 = 68$ feature vector. The pyAudioAnalysis library has been adopted for implementing these audio features [10]. Each unknown audio file is therefore represented using the aforementioned procedure and it is classified using a Support Vector Machine classifier with an RBF kernel. More details on the classification scheme of the handcrafted audio features are given in the experiments section. pyAudioAnalysis has been widely used in several audio classification tasks in the bibliography

doi:10.20944/preprints201811.0509.v1

(e.g., [11,12]) and it implements most basic audio features. In addition, in this paper it has been chosen for its Pythonic implementation that makes it easier to combine with the deep learning - related experiments implemented in Keras and Tensorflow.

4. Context-aware deep learning

4.1. Convolutional neural networks

As with most machine learning application domains, audio analysis has significantly benefited by the recent advances that *deep learning* has offered. Most of the research efforts towards this direction have focused on employing audio features in deep learning schemes for speech recognition [13,14], generic audio analysis [15] and music classification [16]. Also, inspired by the outstanding results of convolutional neural networks (CNNs) in image classification [17], a few research efforts have also focused on particular audio analysis tasks by representing the audio signal as a 2-D time-frequency representation, mainly for musical signal analysis applications [18–22] and speech emotion recognition [23]. In general, such deep-architectures and especially deep CNNs are well known for their ability to autonomously learn highly-invariant feature representations, extracted from complex images.

CNNs have been widely adopted as deep architectures. They are actually a subcategory of traditional neural-networks (ANNs). However, in CNNs, (convolutional) neurons in one or more layers are applied to a small "region" of the layer input, emulating the response of an individual neuron to visual stimuli. These layers are called convolutional and are usually deployed in the beginning of the architectural scheme. CNNs in general use one or more convolutional layers, usually followed by a subsampling step and at the end by one or more fully connected layers, similar to the ones used in traditional multilayer neural networks. CNNs have been proven to achieve the training of computationally large model with very robust feature representations especially in difficult multi-class image classification tasks.

4.2. Context-aware deep audio feature extractors

4.2.1. Signal representation

In this work, we propose using CNNs as estimators of contextual audio classes. Towards this end, instead of using handcrafted audio features, the audio signal is first represented by its **spectrogram**: in particular, a Short-Term FFT (applied on the hanning-windowed version of each raw audio signal) is used to estimate the audio signal's spectrum, using a short-term window of 20 ms with 50% overlap (i.e. the short-term window's size is 10 ms). It has to be noted, that before the spectrogram calculation, the signal is resampled to 16KHz and stereo signals are converted to single-channel. In addition, after the spectrogram calculation, only the first 100 frequency bins are kept. Since each short-term window is 20 ms long, this actually means that only frequencies up to 5000 Hz are kept, i.e. 62.5% of the total spectral distribution. The spectrogram process calculation is performed for each 200 ms mid-term segment. Therefore, for each individual audio segment the respective time-frequency spectrogram representation corresponds to an image of 19×100 resolution. Figure 2 illustrates the adopted process that leads to the signal representation used by the convolutional neural network. Each individual 19×100 spectrogram representation (corresponding to a 200 mseconds segment) is fed as input to the context-aware CNN described in the next section.

4.2.2. CNN architecture

The aforementioned 19x100 spectrogram image is fed as input to the proposed CNN architecture. As shown in figure 3, the convolutional neural network that is trained to recognize audio context classes has first a convolutional layer that takes the initial spectrogram image (19x100) and uses 64 convolutional nodes (3x3 each). This is also followed by a batch normalization and a RELU activation

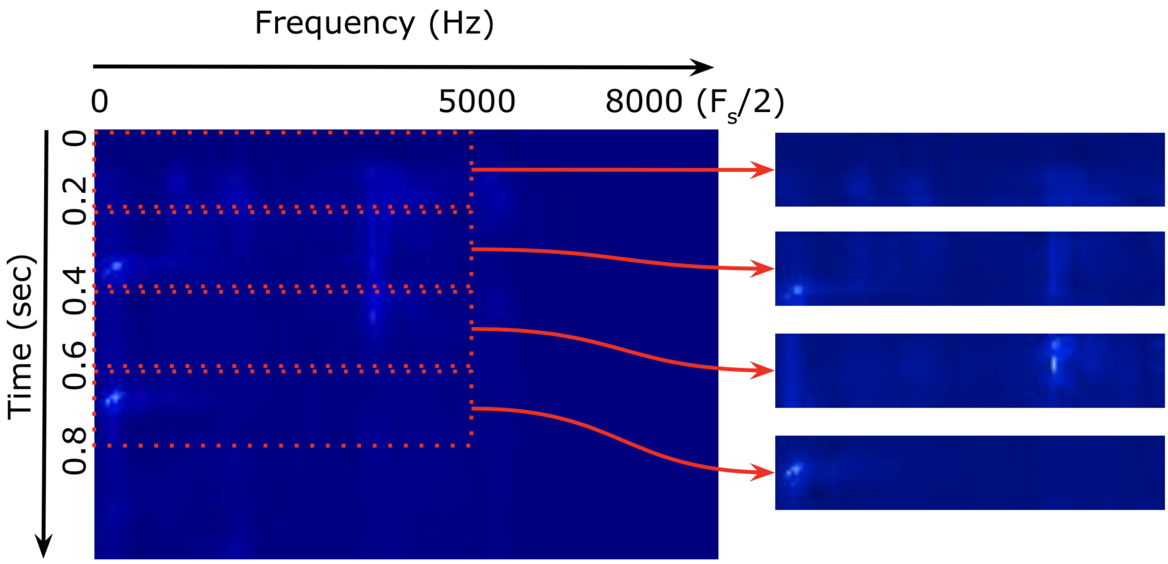

Figure 2. Signal representation adopted in the deep learning scheme: each 19 × 100 spectrogram representation (corresponding to a 200 mseconds segment) is fed as input to the context-aware CNN model

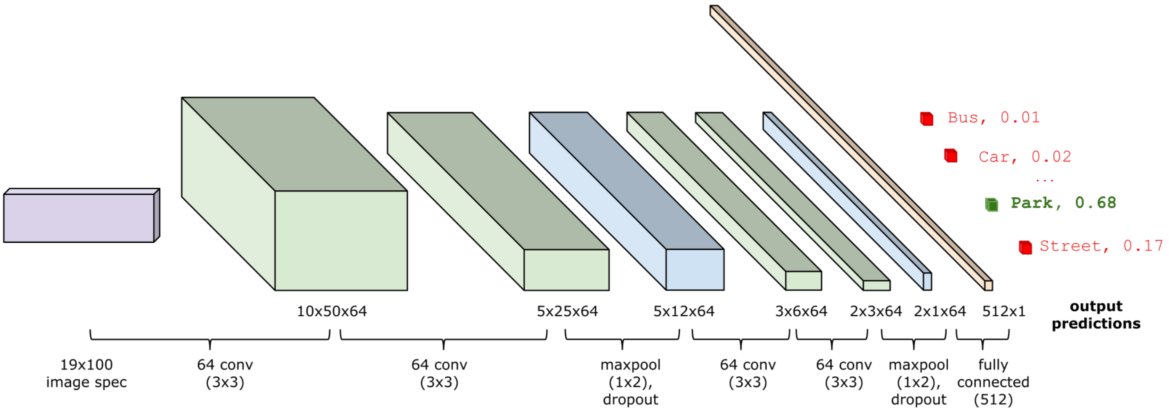

Figure 3. Adopted CNN architecture. In general, two groups of two convolutional layers are used. Each convolutional layer has 64 nodes, while after each group a maxpooling procedure is applied. The final dense layer has 512 nodes

stage [24]. Then a similar 2nd convolutional layer is used with the same number of nodes, followed by a maxpooling and a dropout step. Maxpooling is actually a sub-sampling layer that changes the resolution of its input, in order to facilitate the discovery of more abstract features and avoid overfitting. In this architecture, maxpooling size is set equal to 1x2, in order to only subsample the representation dimensions that correspond to the frequency domain. Dropout layers are used in an additional effort to avoid overfitting [25]. According to the dropout technique, at each training stage, individual nodes are "dropped out" of the net with a particular probability. Incoming and outgoing edges to a dropped-out node are also removed. Dropout is set equal to 0.2 in our system.

The group of these two convolutional layers and the maxpooling layer is then repeated at a second group of layers. Finally, a fully connected (dense) layer that maps the (flat) representation is generated by the last pooling layer to a high-dimensional (512) flat representation. The final output of the network is the prediction of the adopted audio scene classes as described in the sequel. The adopted CNN architectural scheme has resulted after a parameter optimization procedure where the number of convolutional layers, the number of convolutional neurons at each layer and the size of the dense layer have been used as parameters, where performance measures from the TUT dataset have been used (not the final evaluation datasets).

4.2.3. CNN training procedure

In this work, it is our goal to extract audio context-aware knowledge through the utilization of a CNN that has been trained to distinguish between acoustic scene classes. Towards this end, the aforementioned network has been trained using the TUT Acoustic scene classification dataset [26]. This is a widely adopted benchmark of almost 5000 audio segments of 10 seconds each. The segments have been annotated to 15 scene classes that describe the recording context such as: bus, car, city center, home, metro station, residential area, train etc.. The adopted CNN scheme has been trained on several non-overlapping 200-mseconds spectrograms of the aforementioned audio segments of the TUT dataset. In total, around 200 thousand spectrograms have been used to train the CNN. The resulting CNN is used as a feature extractor, i.e. the last output layer is omitted in the overall system.

Both the network and the training procedure has been implemented using the Tensorflow [27] and Keras (keras.io) frameworks. The training procedure has been carried out in a Linux workstation equipped with a Tesla K40c GPU, which achieved a 15x speed boosting, compared to the CPU-based training procedure.

4.2.4. Using CNN as a feature extractor

As soon as the CNN is trained as described above, the input values of its last dense layer (i.e. the 512 values of the respective fully connected layer) are adopted as deep features for each 0.2 seconds audio segment. These 512 feature sequences are then used to produce long-term averages. Note, that this corresponds to the statistic calculation substage of the handcrafted feature extraction described in Section 3. However, in the case of the CNN-based feature extractor, only means of the short-term features are extracted, due to the fact that the CNN training phase has also taken into consideration the temporal evolution of the audio signal (through one of the two dimensions of the spectrogram), so there is no obvious need for further using standard deviation or some other statistic, apart from the mean value.

Therefore, during the audio event recognition phase, each audio recording is represented by 512 averages of the respective 512 short-term sequences of the deep context-aware features. Finally, these 512 features are merged with the 68 handcrafted audio feature statistics, leading to a early fusion feature vector of 580 dimensions in total, for each audio recording.

Table 2. Classification accuracy for all audio feature methods: baseline, handcrafted audio features, context-aware deep features and combination. The performance boosting offered by the combination classification method has been found to be statistically significant ($p < 0.05$)

Method	ESC50	U-8K
BL	44	68
HaF	46.6 ± 2.5	69.4 ± 6.6
CadF	44.2 ± 1.8	65.1 ± 5.2
HaF + CadF	52.2 ± 2	73.1 ± 6.2

5. Experimental evaluation

5.1. Sound Event Datasets

The TUT dataset described in Section 4.2.3 has been used to train the adopted CNN scheme to distinguish between audio scenes that characterize the "context" of the audio signal. In order to evaluate the performance of the final classification approach, two audio event datasets have been utilized:

- The Urban Soundcape Dataset 8K (U-8K) [28], contains 8732 labeled sound excerpts (all less or equal to 4s of length) of urban sounds from 10 audio event classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The average classification accuracy obtained is 68% for the baseline audio classification method.
- The ESC-50 dataset [29] has also been used in various audio classification papers as a benchmark. This is a public labeled dataset of 2000 environmental recordings with 50 audio event classes, 40 sound files per class, 5 seconds per file. The number of classes is much higher in this dataset, therefore the classification accuracy of the reported baseline method is around 44%.

5.2. Experimental Results

Table 2 presents the classification results for all three methods on both datasets: baseline (BL, as described in the corresponding dataset descriptions, i.e., [28] and [29]), Handcrafted audio features (HaF), context-aware deep features (CadF) and combination (HaF + CadF). Note that the HaF results correspond to a tuning procedure in terms of short-term window size and step. The best performance has been reported for a 40 msec frame of 50% overlap (i.e. 20 msec step) and a non-overlapping 2-second segment size. The C parameter of the SVM classifier has been selected in the context of a cross-validation pipeline from the range 0.01 to 50. The selected C value was 10 for both tasks and the respective training errors were 65% and 82% which does not indicate a significant overfitting state. Also, it has to be noted that melgrams have also been evaluated as an alternative signal representation method and they did not lead to performance improvement (they were on average 1% less accurate for both methods).

It can be seen that the HaF method slightly outperforms the CadF feature extraction approach, however the two feature methodologies combined lead to a relative performance boosting of 11% and 5% for the ESC50 and the U-8K datasets respectively. This performance boosting, despite the fact that the CadF method alone hardly outperforms the baseline approach, is a rather important finding. Note that all classification results presented here are for the best Support Vector Machine classifier, with a RBF kernel, where the C parameter has been tuned in a cross-validation procedure. Other widely used classifiers such as random forests and extra trees have been also evaluated but achieved lower classification rates, when used both in the individual feature representation feature methods and in their combinations.

The goal of this paper is to demonstrate the ability of the "context-aware" CNNs to provide an alternative feature representation that boosts the performance of handcrafted audio features. The aforementioned results prove that, even with a baseline classification and fusion approach the combination of the two feature representation methodologies lead to significant performance boosting. Despite the fact that a simple workflow has been used in the classification stage, the overall method achieves comparable results for the U-8K dataset (74% in [30] and 75% in [3]), even if such methods adopt complex deep learning classification schemes that usually require laborious data augmentation procedures and respective parameter tuning [31].

6. Conclusions

In this paper we have demonstrated the utilization of Convolutional Neural Networks that have been trained to distinguish between acoustic scene (context) classes, in a framework for classifying urban audio events. Towards this end, handcrafted audio features as well as features extracted from the proposed CNN are combined in an early fusion approach and classified using a baseline classifier. Extensive experimentation has proven that this combination leads to a relative performance boosting of up to 11%, despite the fact that the performance of the CNN-generated features alone is hardly baseline-equivalent. This is due to the fact that the CNN introduces highly diverse representation, which is not modelled in the handcrafted features.

The contribution of this work is focused in the fact that it experimentally proves that the transfer of contextual knowledge using a CNN trained in a scene-specific dataset can lead to significant performance boosting when combined with typical handcrafted (manually designed) audio features. However, this performance boosting has been demonstrated using a very simple classification scheme (i.e. SVMs performed on simple long-term feature statistics). Our ongoing and future work aims to combine this contextual knowledge in the context of a deep learning framework that will replace the simple feature merging (early fusion) and the (meta)classification technique adopted in this paper.

Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER grant number XXX." and and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding, any errors may affect your future funding.

References

1. Khunarsal, P.; Lursinsap, C.; Raicharoen, T. Very short time environmental sound classification based on spectrogram pattern matching. *Information Sciences* **2013**, *243*, 57–74.
2. Ye, J.; Kobayashi, T.; Murakawa, M. Urban sound event classification based on local and global features aggregation. *Applied Acoustics* **2017**, *117*, 246–256.
3. Huzaifah, M. Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks. *arXiv preprint arXiv:1706.07156* **2017**.
4. Thorogood, M.; Fan, J.; Pasquier, P. Soundscape Audio Signal Classification and Segmentation Using Listeners Perception of Background and Foreground Sound. *Journal of the Audio Engineering Society* **2016**, *64*, 484–492.
5. Giannakopoulos, T.; Siantikos, G.; Perantonis, S.; Votsi, N.E.; Pantis, J. Automatic soundscape quality estimation using audio analysis. Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments. ACM, 2015, p. 19.
6. Kuncheva, L.I.; Whitaker, C.J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning* **2003**, *51*, 181–207.
7. Giannakopoulos, T.; Pikrakis, A. *Introduction to Audio Analysis: A MATLAB® Approach*; Academic Press, 2014.
8. Hyoung-Gook, K.; Nicolas, M.; Sikora, T. *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*; John Wiley & Sons, 2005.
9. Theodoridis, S.; Koutroumbas, K. *Pattern Recognition, Fourth Edition*; Academic Press, 2008.
10. Giannakopoulos, T. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. *PloS one* **2015**, *10*, e0144610.

11. Subramaniam, A.; Patel, V.; Mishra, A.; Balasubramanian, P.; Mittal, A. Bi-modal first impressions recognition using temporally ordered deep audio and stochastic visual features. European Conference on Computer Vision. Springer, 2016, pp. 337–348.
12. Mesaros, A.; Heittola, T.; Diment, A.; Elizalde, B.; Shah, A.; Vincent, E.; Raj, B.; Virtanen, T. DCASE 2017 challenge setup: Tasks, datasets and baseline system. DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events, 2017.
13. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; others. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* **2012**, *29*, 82–97.
14. Dahl, G.E.; Yu, D.; Deng, L.; Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing* **2012**, *20*, 30–42.
15. Lee, H.; Pham, P.; Largman, Y.; Ng, A.Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. Advances in neural information processing systems, 2009, pp. 1096–1104.
16. Scardapane, S.; Comminiello, D.; Scarpiniti, M.; Uncini, A. Music classification using extreme learning machines. 2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA). IEEE, 2013, pp. 377–381.
17. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, pp. 1097–1105.
18. Schlüter, J.; Böck, S. Cnn-based audio onset detection mirex submission.
19. Schlüter, J.; Böck, S. Musical onset detection with convolutional neural networks. 6th International Workshop on Machine Learning and Music (MML), Prague, Czech Republic, 2013.
20. Grill, T.; Schluter, J. Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. Signal Processing Conference (EUSIPCO), 2015 23rd European. IEEE, 2015, pp. 1296–1300.
21. Choi, K.; Fazekas, G.; Sandler, M. Explaining Deep Convolutional Neural Networks on Music Classification. *arXiv preprint arXiv:1607.02444* **2016**.
22. Zhang, C.; Evangelopoulos, G.; Voinea, S.; Rosasco, L.; Poggio, T. A deep representation for invariance and music classification. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 6984–6988.
23. Huang, Z.; Dong, M.; Mao, Q.; Zhan, Y. Speech emotion recognition using CNN. Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014, pp. 801–804.
24. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks* **2015**, *61*, 85–117.
25. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **2014**, *15*, 1929–1958.
26. Mesaros, A.; Heittola, T.; Virtanen, T. TUT database for acoustic scene classification and sound event detection. Signal Processing Conference (EUSIPCO), 2016 24th European. IEEE, 2016, pp. 1128–1132.
27. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; others. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* **2016**.
28. Salamon, J.; Jacoby, C.; Bello, J.P. A dataset and taxonomy for urban sound research. Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014, pp. 1041–1044.
29. Piczak, K.J. ESC: Dataset for environmental sound classification. Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015, pp. 1015–1018.
30. Piczak, K.J. Environmental sound classification with convolutional neural networks. Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on. IEEE, 2015, pp. 1–6.
31. Salamon, J.; Bello, J.P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters* **2017**, *24*, 279–283.