ACOUSTIC SCENE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

Daniele Battaglino[*†], *Ludovick Lepauloux*[*] *and Nicholas Evans*[†]

[*] NXP Software
Mougins, France

[†] EURECOM
Biot, France

ABSTRACT

Acoustic scene classification (ASC) aims to distinguish between different acoustic environments and is a technology which can be used by smart devices for contextualization and personalization. Standard algorithms exploit hand-crafted features which are unlikely to offer the best potential for reliable classification. This paper reports the first application of convolutional neural networks (CNNs) to ASC, an approach which learns discriminant features automatically from spectral representations of raw acoustic data. A principal influence on performance comes from the specific convolutional filters which can be adjusted to capture different spectro-temporal, recurrent acoustic structure. The proposed CNN approach is shown to outperform a Gaussian mixture model baseline for the DCASE 2016 database even though training data is sparse.

Index Terms— acoustic scene classification, convolutional neural networks, local binary patterns, spectrogram

1. INTRODUCTION

Acoustic scene classification (ASC) [1] is a recent area of research which aims to categorize an acoustic scene through contextual sounds. A smart objects or device can exploit information extracted from its immediate *soundscape* [2] to adjust system or application parameters or behavior to meet consumer demands for contextualization and personalization. One real example of such technology is an automatic process to increase a ringtone volume when a smart phone is moved from a quieter environment into a noisier one.

Acoustic scenes usually exhibit a high degree of variability, both inter class and intra class. ASC is thus arguably among the most challenging of statistical pattern recognition tasks. Almost all current approaches rely on hand-crafted features chosen specifically to facilitate discrimination between an often-small set of known acoustic classes. With the variability in acoustic scenes being so high, the premise of the research presented here is that hand-crafted features are a bottleneck to ASC performance and that automatically derived features have greater potential.

Deep learning techniques have brought tremendous advances in a huge range of different statistical pattern recognition applications [3] and is now the state of the art in many, if not the majority. These techniques and tools offer one alternative to hand-crafted features and a suite of different approaches to automatic feature learning from raw input data (e.g. images and audio recordings).

This paper reports what is, to the best of the authors' knowledge, the first attempt to harness the power of automatic feature learning for ASC using a deep learning architecture known as a convolutional neural network (CNN). CNNs operate on a raw spectrogram representation of the acoustic data, thereby avoiding reliance on hand-crafted features.

The remainder of this paper is organized as follows. Section 2 summarizes the prior work in ASC. Section 3 presents the new CNN approach to ASC with an emphasis on its adaptation to acoustic data. Section 4 describes specific implementation details, an assessment protocol and experimental results. Conclusions, discussion and suggestions for further work are presented in Section 5.

2. PRIOR WORK

The literature shows that the majority of approaches to ASC utilize features developed for speech and music processing tasks such as speech or genre recognition. Examples include Mel-scaled frequency cepstral coefficients (MFCCs), spectral flatness, spectral centroid, and the zero-crossing rate [4, 5]. Some recent work [6], however, shows that such features may not be sufficiently discriminative for ACR. MFCC features, for example, capture only short term variation but only minimum dynamic information, information which is useful to discriminate between different contexts.

The work in [6] shows the benefit of capturing auto-correlation in the temporal domain through a similarity matrix which reflects recurrent, dynamic structure. The work in [7] showed similar performance gains. Accordingly, advanced time-frequency features for ACR have attracted growing attention. Some of this work has drawn upon methods popular in other two-dimensional problems such as image processing, e.g. local binary pattern (LBP) analysis applied to spectrogram images [8] and histogram of gradient (HOG) approaches [9].

Having been applied so successfully to other, related problems, deep learning techniques [10] are now emerging [11]. Deep neural networks (DNNs) are able to identify and extract optimized, discriminant features from training data and thus offer one alternative to hand-crafted features. Many different architectures and data input representations have been investigated for a host of different applications such as image and speech recognition [12, 13].

While the first investigation of DNN approaches to ASC [11] showed promising results, the work was based upon MFCC input features. The potential benefit of deep learning was thus still curbed by the initial use of hand-crafted features.

While significant progress in ASC has been made in recent times, we argue that the reliance of the past work on hand-crafted features remains a bottleneck. We have thus sought to improve ASC performance through the application of automatic feature learning. This paper reports our first experiments with a particular approach to deep learning involving convolutional neural networks (CNNs).

3. CONVOLUTIONAL NEURAL NETWORK FOR ASC

In continuation of our previous work [8], this paper reports the application of a convolutional neural network (CNN) to the task of

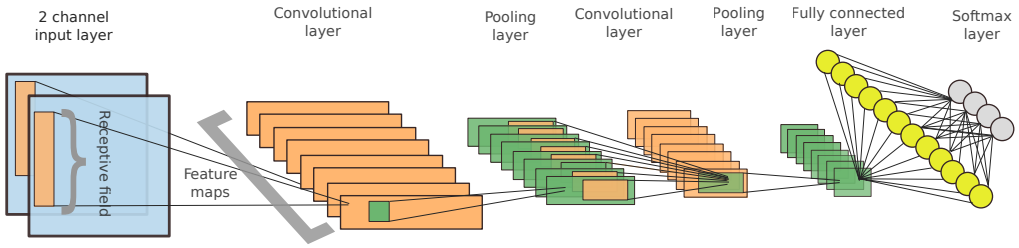

Figure 1: The CNN architecture used in this work. The input is static and dynamic spectrograms. These are followed by two, stacked convolutional and pooling layers. Fully connected and output layers produce the probabilities of the input data belonging to each acoustic class. Convolutional filters are illustrated in light orange while pooling blocks are illustrated in dark green.

ASC. CNNs have been used successfully in a wide variety of related tasks such as speech recognition [14], music analysis [15] and event detection [16]. The motivation behind this the adoption of this approach for ASC lies in (i) the potential to use a raw time-frequency representation as the input, (ii) the replacement of hand-crafted features with automatically learned features and (iii) the potential to capture recurrent, spectro-temporal structure.

3.1. Global architecture

CNNs have a multi-layered, deep network architecture. While is some sense a natural extension of the standard multilayer perceptron model, they exhibit some significant differences: first, CNNs can handle high-dimensional data; second, each hidden unit is connected only to a sub-region of the input data, referred to as the *receptive field*, and thus captures only local structure; third, CNNs can capture recurrent local structure. The architecture proposed in this work is illustrated in Fig. 1. It is composed of an input layer, a stack of convolutional and pooling layers, a fully connected hidden layer and a final output layer.

3.2. Input data

In the application of CNNs to computer vision tasks, individual images are typically split into different color channels (e.g. red, green and blue). Each hidden unit then has access to the corresponding *receptive field* in each color channel [12]. This same idea can be adopted in the application of CNNs to ASC. As illustrated in Fig. 2, input *images* take the form of (i) a static, log-Mel spectrogram and (ii) a separate, dynamic spectrogram representation composed of its first derivatives (Δ). These two representations form a two-channel input to the network so that hidden units can combine static and dynamic information.

One further operation is required in order that the CNN is fed with representations of fixed-size. As also illustrated in Fig. 2, both static and dynamic spectrograms are segmented into smaller components which are treated as independent inputs. As a result, the input data is artificially augmented, the resolution of the CNN is increased whereas the complexity is reduced.

3.3. Convolutional layer

Complex acoustic scenes contain discriminative spectro-temporal recurrent structure, e.g. engine noise and telephone ringtones. These characteristics are referred to here as *local patterns*, namely a recurrent concentration of energy over both frequency and time.

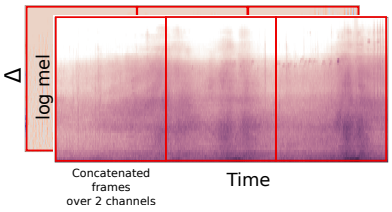

Figure 2: CNN input data is a pair of static (log-Mel) and dynamic (first derivatives, Δ) spectrograms. Each is first segmented into smaller sub-clip illustrated in red, each then forming separate input data.

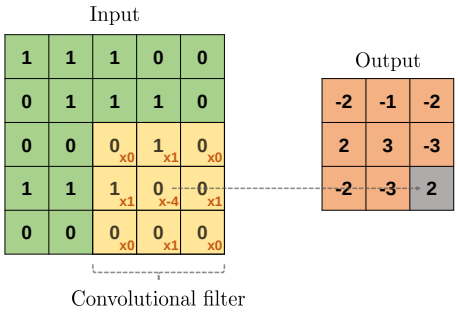

Figure 3: An example convolution between a 3×3 block of input data (the *receptive field*) and a *convolutional filter* with weights W (expressed in the bottom right corner of the image).

Engine noise, for example, is characterized by a predominant local pattern spanning the time axis whereas ringtones may exhibit a recurrent pattern spanning the frequency axis.

Such local patterns can be represented through a convolution operation between an input I and a set of filter weights W which produces an output O:

$$O[i,j] = I[i,j] * W[i,j] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} I[u,v]W[i-u, j-v]$$

where i and j are row and column indices of I, and u and v are row and column indices of W. The two-dimensional convolution operation is illustrated in Fig. 3 for a filter W centered on the lower right component of the input image I.

The convolution is the heart of the convolutional layer in the CNN. Each output of the hidden unit, denoted h_{ij}, is linked to a

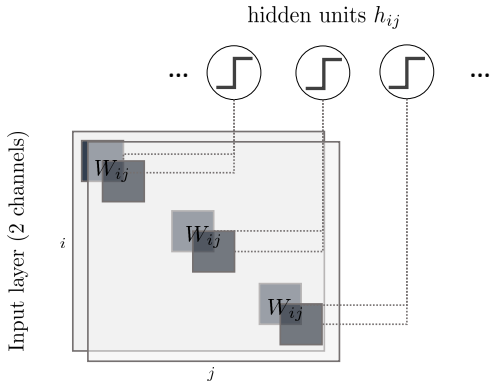

Figure 4: An illustration of the application of CNNs to the input data showing: (1) the local connectivity between each hidden unit and the *receptive field* at coordinates i, j in the input layer; (2) the use of multiple input layers (referred to as *channels*) which maintain the same relationship between receptive fields and hidden units; (3) the use of the same filter weights W which are shared for all input layers and thus capture similar, recurrent patterns in the input data.

local *receptive field* in the input data centred on the i, j coordinates of input data I through filter weights W:

$$h_{ij} = \sigma((W * I)_{ij} + b)$$

where σ is a non-linear activation function and b is a bias. Weights are shared among different hidden layer units in order to reduce the total number of trainable parameters. An example of local hidden units with shared weights W is illustrated in Fig. 4. Hidden unit outputs h_{ij} form new layers of spatially connected neurons which are referred to as *feature maps*. Different *feature maps* can be formed from different combinations of locally connected hidden units, each sharing the same weights applied to different positions of the input space.

3.4. Pooling layer

Pooling layers are applied to the output of each convolutional layer in order to reduce their resolution. Different strategies can be applied. Among the simplest is a max operation whereby a block of values in the pooling layer input are replaced with their single, maximum value. What is effectively an operation of down-sampling has been shown to not only reduce dimensionality, but also to provide invariance to the translation of structure in the input [14].

For the ASC task, this properly provides invariance to small changes in spectro-termporal structure. For example, the same local pattern (e.g. engine noise) centered on a specific frequency may vary only marginally from one recording to another. The pooling operation allows to reduce the frequency or time resolution giving more importance to the pattern rather than its spectro-temporal location.

3.5. Fully connected layer

Convolutional and pooling layers can be replicated in sequence in order to add depth and to produce higher level representations of the input. Classification is finally achieved via fully connected and softmax layers. Inputs are the full set of outputs emerging from the last convolutional or pooling layer. Outputs of the fully connected layer are again fully connected. By definition, fully connected layers do not operate at a local level and instead act to classify the input as one of the output acoustic scenes.

Figure 5: An illustration of the 3×3 filter weights W of the first CNN layer.

Each input segment (of the original, full spectrogram) is classified independently. The most probable acoustic scene or class $\hat{y}$ is identified with a *softmax* function. The classification for the spectrogram as a whole is made according to a majority vote. The objective function used for CNN training (i.e. for the optimization of model parameters W and b) is the minimization of a loss function l between target y and predictions $\hat{y}$ over N input samples:

$$l(\theta = \{W, b\}, N) = -\frac{1}{N} \sum_{n=1}^{N} y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)$$

3.6. Insights

Thus far our experiments with CNNs have investigated the influence of the convolutional filter structure on classification performance. One can readily see that differences in height and width will alter the relative importance of spectral and temporal variation. Relatively square-shaped filters will place similar importance on both spectral and temporal variation. Example filters obtained from the first CNN layer are illustrated in Fig. 5. They show an interesting similarity to uniform local binary patterns [17] used in our previous approach to ASC [8].

Filter patterns different to those in Fig. 5 are obtained by altering the dimensions of the convolutional filters. As reported below, we have found that tall and narrow filter shapes tend to give the best performance. While this observation is illustrative of the delicate balance one must strike between the relative importance of spectral and temporal variation and resolution, clearly both are important to the task of ASC.

4. EXPERIMENTS AND RESULTS

Described here are specific details of our implementation and ASC results for the DCASE 2016 database [18]. Since the later is a standard database used by all challenge participants, it is not described in detail here.

4.1. Implementation details

Audio signals are first treated in the usual way from the application of the discrete Fourier transform to 40ms frames with an overlap

Table 1: ASC performance for the DCASE 2016 development set. Results illustrated for the baseline GMM system and the proposed CNN approach.

Method	accuracy	fold1	fold2	fold3	fold4
GMM	72.6%	67.2%	68.9%	72.3%	81.9%
CNN	**76.0**%	80.6%	67.0%	77.9%	78.7%

of 20ms. Static spectrograms are formed from magnitude spectra which are passed through a bank of 60 log and Mel-scaled filters [19] with a maximum frequency of 22050 Hz. Dynamic Δ spectrograms are calculated in the usual way with a time-window of 9 frames. Each 30s clip of the DCASE database is thus split into 25 sub-clips of 1.2 seconds duration. Each sub-clip is furthermore represented with both static and dynamic spectrogram segments, as illustrated in Fig. 2, resulting in input data of 60 bands $\times$ 60 frames.

The CNN has two stacked pairs of convolution and pooling. The first convolutional layer contains 32 filters each of which spans 57 frequency bands and 6 frames (342 elements). This results in a set of 32 feature maps each of 4 bands and 55 frames (220 elements). On account of the relative dimensions of spectrograms and filters and the overlap inherent to the convolution, the frequency resolution is reduced whereas the time resolution is increased. The pooling layer performs max-pooling over 2 adjacent units in both frequency and time, reducing by one half the dimension of the previous convolutional layer. A second convolutional layer creates 32 feature maps using filters each of which spans 1 band and 2 frames (2 elements). We have used a linear rectifier for the activation function of the convolutional hidden units which is preferred for the modeling of real values [20].

The fully connected layer is comprised of 2000 nodes and is followed by a softmax layer which returns output probabilities for all 15 DCASE classes. Regularization is performed using a 50% dropout [21] both before and after the fully connected layer. Data is treated in batches of 1000 input samples and the network is trained for 100 epochs. The learning rate is set to 0.001 with an initial momentum [22] of 0.9 which is increased linearly to 0.99 for the final epoch.

4.2. Protocols

In order to perform supervised classification, we created an additional validation set which was randomly selected from the training set (i.e. 20% of the training data in each fold, corresponding to 8 files). This validation set is used for CNN learning since model parameters are tuned on the validation loss.

4.3. Results

Classification results are illustrated in Table 1 for both the Gaussian mixture model DCASE baseline system and the new CNN approach reported in this paper. Results are illustrated in terms of average accuracy (second column) and individually for each of the 4 folds (columns 3–6). The average accuracy is seen to improve from 72.6% for the baseline system to 76.0% for the CNN system. Also illustrated is a high degree of variation between folds. This variation is consistent for both GMM and CNN systems and is probably a consequence of limited data.

As discussed in Section 3, the shape of the convolutional filters can be adjusted to balance the relative importance of spectral and temporal information. This effect of filter shape is illustrated in Fig. 6 which illustrates average accuracy independently for each of the 15 DCASE classes for two different convolutional filter shapes. The first filter is the same as that for which results are reported above. These result are illustrated with red bars in Fig. 6. Instead of the 'tall and thin' filter used for all experiments reported above, the second filter is 'short and fat'. This filter places more emphasis on temporal variation. Results for the second filter are illustrated with green bars in Fig. 6. Whereas the first filter outperforms the second, differences serve to illustrate the effect of filter shape; the second filter delivers better performance for classes with greater temporal recurrent structure, such as bus or car noise. These observations warrant investigation in future research.

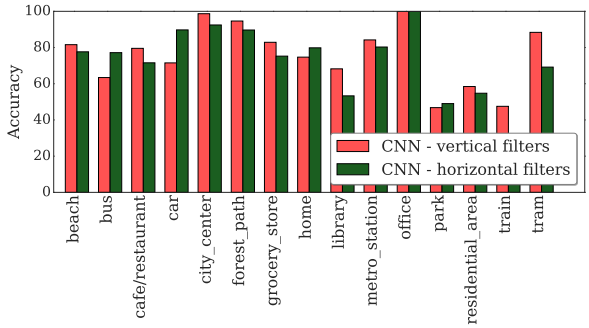

Figure 6: Accuracy per class for two convolutional filter variants: red bars illustrate performance for the proposed method for a 'tall and thin' filter with 57 bands and 6 frames; those in green illustrate performance for a 'short and fat' filter with 6 bands and 57 frames.

5. CONCLUSIONS

This paper describes the first application of convolutional neural networks (CNNs) to acoustic scene classification. In contrast to past work which has used almost exclusively hand-crafted features, the paper shows how CNNs can be used to learn recurrent, local patterns automatically from spectro-temporal representations of raw acoustic data. Local patterns are captured through a combination of convolutional and pooling layers which produce higher level representations of the input data. While different convolutional filter shapes capture different degrees of spectro-temporal structure, tall and thin convolutional filter shapes, which offer better resolution in the frequency domain, give the best performance for the standard DCASE database.

While the CNN approach to ASC proposed in this paper is shown to outperform the baseline system, performance assessed on the evaluation set may yet prove disappointing. Unfortunately, lack of sufficient data may cause a severe bottleneck, offering limited potential to avoid over-fitting. With evaluation restrictions on the use of external data, an assessment of generalization remains a direction for future work.

Nonetheless, external data, albeit unlabeled is available in abundance. The ASC approach reported in this paper is readily adapted to unsupervised feature learning. Only then may the true potential of CNNs for ASC be revealed. Future work should include a thorough investigation of different spectro-temporal representations as CNN input data and of different convolutional filters.

6. 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, May 2015.

[2] R. Schafer, *The Soundscape: Our Sonic Environment and the Tuning of the World*. Inner Traditions/Bear, 1993.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] J. T. Geiger, B. Schuller, and G. Rigoll, "Recognising acoustic scenes with large-scale audio feature extraction and SVM," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[5] W. Nogueira, G. Roma, and P. Herrera, "Sound scene identification based on MFCC, binaural features and a support vector machine classifier," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[6] G. Roma, W. Nogueira, P. Herrera, and R. de Boronat, "Recurrence quantification analysis features for auditory scene classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[7] K. Patil and M. Elhilali, "Multiresolution auditory representations for scene classification," *cortex*, vol. 87, no. 1, pp. 516–527, 2002.

[8] D. Battaglino, L. Lepauloux, L. Pilati, and N. Evans, "Acoustic context recognition using local binary pattern codebooks," in *WASPAA 2015, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 18-21 October 2015, New Paltz, NY, USA*, 10 2015.

[9] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of Time-Frequency Representations for Audio scene detection," *arXiv:1508.04909 [cs]*, Aug. 2015, arXiv: 1508.04909.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," *CoRR*, vol. abs/1404.7828, 2014.

[11] Y. Petetin, C. Laroche, and A. Mayoue, "Deep neural networks for audio scene recognition," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*, Aug 2015, pp. 125–129.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, 2013, pp. 315–320.

[14] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP)*. IEEE, 2012, pp. 4277–4280.

[15] J. Schlter and S. Bck, "Improved musical onset detection with convolutional neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6979–6983.

[16] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept 2015, pp. 1–6.

[17] T. Ojala, M. Pietikinen, and T. Menp, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transacrtions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

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

[19] B. McFee, M. McVicar, C. Raffel, D. Liang, O. Nieto, E. Battenberg, J. Moore, D. Ellis, R. Yamamoto, R. Bittner, and et al., "librosa: 0.4.1," 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.32193

[20] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *Proc. ICML*, vol. 30, no. 1, 2013.

[21] N. Srivastava, "Improving Neural Networks with Dropout," Master's thesis, University of Toronto, Toronto, Canada, January 2013.

[22] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning." *ICML (3)*, vol. 28, pp. 1139–1147, 2013.