Deep Convolutional Recurrent Neural Network for Rare Acoustic Event Detection

Shahin Amiriparian[1,2], Sahib Julka[1], Nicholas Cummins[1], Björn Schuller[1,3]
[1] ZD.B Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany
[2] Machine Intelligence & Signal Processing Group, Technische Universität München, Germany
[3] GLAM – Group on Language, Audio & Music, Imperial College London, U. K.
shahin.amiriparian@tum.de

Abstract

Rare acoustic event detection, as evidenced by the recent IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2017), is a growing field of acoustic classification research. Rare audio events often possess unique spectral and temporal structures which can aid their identification. In this regard, we investigate the advantages of a hybrid combination of convolutional neural network and a recurrent neural network to classify rare occurring sound events in audio streams. Our developed system uses log-Mel spectrograms fed into convolutional layers to first extract high-level, shift- invariant spectral features. Recurrent layers are then used to learn the long-term temporal context from obtained high-level features. Finally, using a feed forward neural network with sigmoid activations, a sequence of probability estimations is used to predict the onset and presence of the rare sounds. We develop and test our system on the Detection of Rare Sound Events task of the DCASE 2017 challenge. Key results presented indicate that our proposed approach outperforms the challenge baseline, improving the overall detection error rate from 0.63 to 0.29 on the evaluation dataset.

1 Introduction

Monitoring systems using audio sensors, in addition to video sensors, are becoming increasingly popular [1]. Audio is especially useful when video fails to effectively detect an event. In situations where video is occluded, audio event detection is more effective to use, given that the event has an audio characteristic. Rare sound event detection (RSED) is a newly proposed task that aims to automatically detect certain emergency sounds in acoustic signals with a high degree of accuracy. Such a system has many benefits in surveillance and smart home systems, including gunshot and intrusion detection or baby-cry monitoring.

This need has given rise to research interest in developing better techniques for audio event detection, both monophonic [2] and polyphonic sound events [3, 4]. Monitoring systems need to be able to focus on the specific alarm of interest with high accuracy. Since these sounds rarely occur simultaneously, it is useful to explore monophonic sound event detection (SED) techniques for such a task. As for the algorithms applied, SED has seen use of non-negative matrix factorisation (NMF) [5] for source separation, and hidden Markov models [2] and support vector machines (SVMs) [6] for acoustic modelling. However, recent approaches using deep learning have been more effective [3, 7]. In this paper, we utilise a convolutional recurrent neural network (CRNN) deep learning approach for precise onset detection of emergency sound events. CRNNs were first proposed for document classification [8] and can be regarded as state-of-the-art in many audio tasks [9]. Convolutional neural networks (CNNs) themselves are well known for their ability to learn robust, task specific feature representation and have been successfully applied in SED [10, 11], speech recognition [12] and audio analysis [13, 14]. At the same time, recurrent neural networks (RNNs) are well known for their strengths in modelling temporal sequences. Long short-term memory (LSTM) RNNs have also been used in related tasks of event detection [4], scene classification [15], and sound classification [7]. In this work, we use a combination of a CNN and an RNN, along with a feed forward network (FFN) to classify and detect the sound class of interest with a relatively precise time of onset. This hybrid combination allows for the global temporal context to be taken into account, while efficiently extracting features [9], and thus reducing the network complexity.

2 Approach

A high-level overview of our deep learning approach is given in Figure 1. Our system is composed of five main components: a) augmented mixture generation, b) extraction of log-Mel spectrogram, c) feature extraction with a CNN, d) temporal modelling with an LSTM-RNN, and e) final predictions using a FFN and post-processing.

2.1 Pre-processing

Recently, several researches have demonstrated the advantage of using log-Mel energies for SED [16, 3]. A major advantage borne by these features in comparison to simple spectrograms is the filtering of frequency components with log scale filter banks similar to human ears. Exploring the advantages of Mel-spectrogram based features, we use them in this study. First, we extract frame-wise spectrogram with window size of 46 ms, and then apply 128 Mel-filters to frequency component of each of the frame. We also apply a logarithm on the amplitude. We then break the Mel-spectrogram into chunks with timestep (τ) to be fed into the convolutional neural network.

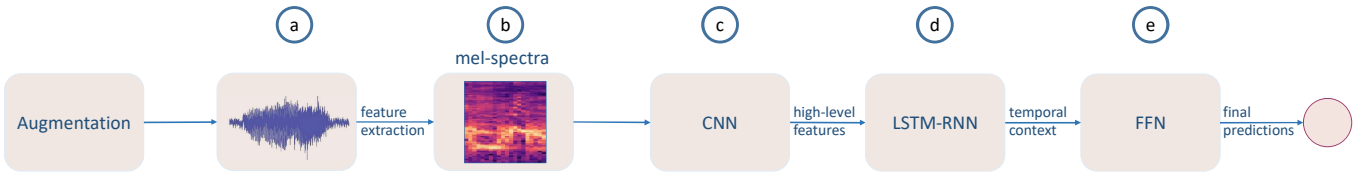

Figure 1: Illustration of the proposed CRNN approach composed of convolutional and recurrent neural networks for feature extraction and a feed forward network to generate the final predictions. A detailed account of the procedure is given in Section 2.

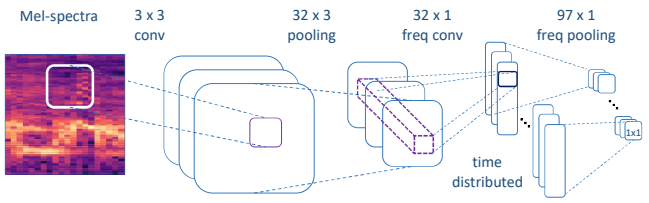

Figure 2: The structure of the 2D convolutional neural network applied for extracting high-level features from the input Mel-spectrograms.

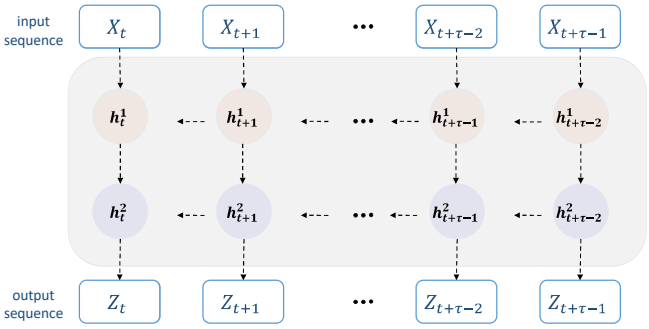

Figure 3: Two backward RNN-LSTM layers with 128 hidden units (h)each. Outputs (Z) are returned for all inputs (X) during the timestep (τ).

2.2 Convolutional Layers

The features, after extraction as chunks, are fed into a convolutional layer with 2D filters. As depicted in Figure 2, the frequency time convolution is followed by non-overlapping pooling to ensure no shrinking in time. Subsequently, a 1D convolution along the spectral domain is applied, which is then followed by max pooling along the frequency domain. Rectified linear unit (ReLU) activation is used in the convolutional layers, and batch normalisation (BN) [17] is applied between them. Furthermore, a dropout of 30 % is used for all layers to add regularisation and also to minimise potential overfitting problems caused by non-overlapping max pooling [18]. Convolutional layers, however, are capable of effectively capturing only short-term temporal context, often in the range < 200 ms [16]. To achieve longer temporal modelling, we pass the outputs from the CNN to a LSTM network.

2.3 Recurrent Layers

The activations emerging from the CNN are passed to a network comprising of two RNN layers (cf. Figure 3). Each RNN layer consists of 128 hidden LSTM units, applied in the reverse direction, in contrast to the traditional bidirectional or unidirectional RNN layers. Our preliminary results indicated that reverse RNN layers work better for this specific problem, possibly because it is more effective to first detect the peak, in order to detect the onset precisely. We apply hyperbolic tangent as the activation function and a dropout of 30 % for each of the layer. At the end, a total of 128 features are obtained for each timestep, which are passed on to the fully connected layer to obtain prediction results.

2.4 Fully Connected Layer

The features returned for each timestep from the LSTM-RNN layers are fed into a fully connected FFN comprising of a single layer with 128 hidden units, matching the depth of the input features. We apply BN to the output, so that the mean is close to 0 and standard deviation is close to 1. The activation function used is ReLU, which adds the desired non-linearity to activations. The updated features are further fed into a time distributed output layer with one sigmoid unit to obtain probabilities at each timestep.

2.5 Post-processing

Sound events typically occur for a period of few hundred milliseconds to a few seconds. Therefore, when a particular frame shows activity presence, it is also likely that the frames around it have higher chances of activity. Additionally, if an activity is detected for n number of frames, then not detected for two further frames and then detected again, it is likely that the predictions from those two frames are noisy. In order to overcome such an issue, we use the technique of *sliding ensemble* to average the overlapping predictions and obtain smoother outputs. A window size equal to the number of timesteps and with a hop size of 1 is used to obtain temporal probability sequence. We then apply fixed thresholding to estimate the presence of an event and the onset time. A threshold of 0.8 for *babycry* and *glassbreak* and 0.5 for *gunshot* is applied for event presence in an entire audio clip. If an event is present in an audio clip, the peak is then calculated and a certain number of frames is checked before the peak; the first frame with $p > 0.5$ is determined to be the onset. Figure 4 illustrates the process of applying the thresholds on an example prediction.

Table 1: Contrast between the positive and negative labels, illustrating the data imbalance.

Target Event	Positive	Negative
Babycry	4.13%	95.87%
Glassbreak	2.47%	97.53%
Gunshot	2.47%	97.53%

Table 2: Final models comprising of weighted average ensembles. t(n) signifies number of timesteps (τ) used for input chunks.

Target Event	Timestep Ensemble
Babycry	$(t5 + t9 + t50)/3$
Glassbreak	$t3$
Gunshot	$(t3 + t5)/2$

3 Experimental Settings

3.1 Database

The DCASE 2017 [19] challenge task 2 dataset has been used for experiments in this work. It consists of samples from 15 different everyday acoustic scenes (home, park, train, cafe, etc.), some of which are mixed with isolated recordings from one of the three different target sound event classes, namely, *babycry*, *glassbreak*, and *gunshot*. The isolated recordings are divided into segments, and relevant target classes are selected by a human annotator. Mixing is performed by adding a segment to the 30-second long background acoustic scene sample with a random time offset. The mean duration of the events is $< 2.5\,s$, thus enforcing the idea of 'rare'. There is also a big gap between positive and negative labels in the dataset (cf. Table 1).

3.2 Configuration

The hyperparameters adjusted for our CRNN are given in Table 3. The total duration of the training set is 25 hours, which is twice the size of the development set with default parameters (event presence probability: 0.5, mixtures per class: 500, event to background ratio: -6, 0, 6). For the training, mixtures were created with parameters from the table. For the validation, the default train set was used. And for testing, a pre-combined test set which contains 1 500 audio clips (500 per event class) was used.

4 Results and Discussion

The evaluation of the classification performance is done using event-based error rate (ER) [20], which involves calculating the true positives, false positives and false negatives. These metrics were computed using the SED toolbox provided in the challenge [19]. Table 2 shows the timesteps combined and averaged for an ensemble. Table 4 shows the results of our approach (CRNN) for the three subtasks and compares the performance with

Table 3: Hyperparameters for our CRNNs for each subtask, together with their approach identifiers. N_c : total convolutional layers; N_r : total recurrent layers; lr : learning rate; lrd : learning rate decay. Each network is trained for atleast 100 epochs and then stopped using early stopping with patience of 15 epochs.

Parameters	Babycry	Glassbreak	Gunshot
N_c	2	2	2
N_r	2	2	2
timesteps (τ)	5, 9, 50	3	3, 5
lr	0.001	0.001	0.001
lrd	0.01	0.01	0.01

Table 4: Comparison of our proposed CRNN and CNN+FFN approaches with the challenge baseline (FFN) [19]. The CNN used here has the same hyperparameters as the CRNN.

ER	Babyc.	Glassb.	Guns.	Average	
Model	test	test	test	dev	test
CRNN	**0.21**	**0.19**	**0.41**	0.17	**0.29**
CNN	0.48	0.27	0.56	0.35	0.43
FNN	0.80	0.38	0.53	0.53	0.63

a CNN and an FNN from the challenge baseline [19]. It can be clearly seen that the CRNN outperforms other networks by a wide margin. The event *glassbreak* has the best performance regardless of the network configuration. It is due to the nature of the event that the frequency component, at the moment when the glass breaks, is impulsive and distinct in comparison to the background. Therefore, short timestep is effective for this problem. However, sometimes other events with similar onset frequencies get confused. The event *gunshot* is also an impulsive sound and requires short timestep analysis. But in a gunshot, there are usually several vibrations between the onset and the offset, hence relatively longer timestep works effectively for this task. In the case of *babycry*, the event lasts for longer periods and so requires the use of longer timestep frames.

5 Conclusions and Future Work

To find the right parameters, several experiments were conducted with different features and timesteps. Based on our experiments, we observed that data augmentation using synthetically created mixtures and frame concatenation timesteps are important hyperparameters in this task. Therefore, it could be concluded that this is a data driven approach and the context window sizes in this task have direct effect on the performance of the trained network.

Our results indicate that the post-processing step (cf. Section 2.5) has strong impact on the value of predictions, as it helps to remove unwanted noise. We show that the applied CRNN lead to better results in compar-

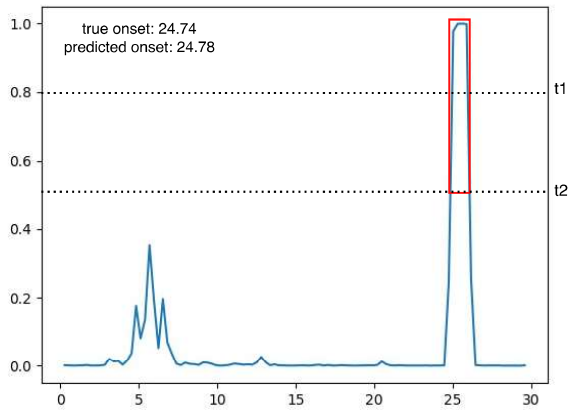

Figure 4: An example of prediction on mixture_devtest_babycry_057.wav. Threshold $t1$ detects event presence in the audio clip, while threshold $t2$ detects the onset time accurately.

ison to CNN and FFN approach. In addition, CRNNs have less complexity compared to RNNs because of the fact that CNNs provide abstraction and reduce the overall number of trainable parameters. Finally, based on our experiments, we also conclude that using ensembles of different timesteps leads to stronger predictions.

As part of future work, it would be of high interest to experiment with more timestep ensembles to obtain robust predictions. We observed that in some cases, sound events can get confused with one another. Hence, it could be also worthwhile to use source separation techniques as a preprocessing step. In the post-processing, the step of sliding ensemble can be improved as well. Finally, we want to explore the benefits of collecting further data from social multimedia using our purpose built software [21] to train the CRNN with more real-world recordings.

References

[1] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

[2] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Signal Processing Conference, 2010 18th European*. IEEE, 2010, pp. 1267–1271.

[3] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–7.

[4] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6440–6444.

[5] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 151–155.

[6] A. Temko and C. Nadeu, "Classification of acoustic events using svm-based clustering schemes," *Pattern Recognition*, vol. 39, no. 4, pp. 682–694, 2006.

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

[8] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[9] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2392–2396.

[10] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 559–563.

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

[12] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[13] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, S. Pugachevskiy, and B. Schuller, "Bag-of-deep-features: Noise-robust deep feature representations for audio analysis," in *Proceedings 31st International Joint Conference on Neural Networks (IJCNN)*, IEEE. Rio de Janeiro, Brazil: IEEE, July 2018, 8 pages, to appear.

[14] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, and B. Schuller, "Snore Sound Classification Using Image-based Deep Spectrum Features," in *Proceedings INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*, ISCA. Stockholm, Sweden: ISCA, August 2017, pp. 3512–3516.

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

[16] E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 1744–1748.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[18] 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.

[19] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

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

[21] S. Amiriparian, S. Pugachevskiy, N. Cummins, S. Hantke, J. Pohjalainen, G. Keren, and B. Schuller, "CAST a database: Rapid targeted large-scale big data acquisition via small-world modelling of social media platforms," in *Proc. 7th biannual Conference on Affective Computing and Intelligent Interaction (ACII 2017)*, AAAC. San Antonio, TX: IEEE, October 2017, pp. 340–345.