# Convolutional Recurrent Neural Networks for Bird Audio Detection

Emre Cakir
Tampere University of Technology, Finland
Email: emre.cakir@tut.fi

Sharath Adavanne
Tampere University of Technology, Finland
Email: sharath.adavanne@tut.fi

Giambattista Parascandolo
Tampere University of Technology, Finland
Email: giamba92@gmail.com

Konstantinos Drossos
Tampere University of Technology, Finland
Email: konstantinos.drossos@tut.fi

Tuomas Virtanen
Tampere University of Technology, Finland
Email: tuomas.virtanen@tut.fi

***Abstract*—Bird sounds possess distinctive spectral structure which may exhibit small shifts in spectrum depending on the bird species and environmental conditions. In this paper, we propose using convolutional recurrent neural networks on the task of automated bird audio detection in real-life environments. In the proposed method, convolutional layers extract high dimensional, local frequency shift invariant features, while recurrent layers capture longer term dependencies between the features extracted from short time frames. This method achieves 88.5% Area Under ROC Curve (AUC) score on the unseen evaluation data and obtains the second place in the Bird Audio Detection challenge.**

## I. Introduction

Bird audio detection (BAD) is defined as identifying the presence of bird sounds in a given audio recording. In many conventional, remote wildlife-monitoring projects, the monitoring/detection process is not fully automated and requires heavy manual labor to label the obtained data (e.g. by employing video or audio) [1], [2]. In certain cases such as dense forests and low illumination, automated detection of birds in wildlife can be more effective through their sounds compared to visual cues. Besides, acoustic monitoring devices can be easily deployed to cover wide ranges of land. This indicates the need for automated BAD systems in various aspects of biological monitoring. For instance, it can be applied in the automatic monitoring of biodiversity, migration patterns, and bird population densities [2]. BAD systems can be augmented with another classifier to determine the species of the detected birds [3]. Using an automated BAD system as preprocessing/filtering step to determine the bird species would be beneficial especially for remote acoustic monitoring projects, where large amount of audio data is employed.

In this regard, the Bird Audio Detection challenge [4] is organized with an objective to stimulate the research on BAD systems which can work on real life bioacoustics monitoring projects. The challenge provides three bird audio datasets recorded in different acoustic environments. Two of the datasets are provided with bird call annotations to be used as development data. The final dataset consists of recordings from a different physical environment and it is employed as the evaluation data. An extensive review on the recent work on BAD can also be found in [4].

Bird sounds can be broadly categorized as vocal and non-vocal sounds (such as bill clattering, and drumming of woodpeckers) [5]. Since non-vocal bird sounds are harder to associate with birds without any visual cues, the research on BAD has been mostly focused on vocal sounds, as in this work. Vocal sounds can be further categorized as bird calls and bird songs. Bird calls are often short and serve a particular function such as alarming or keeping the flock in contact. Bird songs are typically longer and more complex than bird calls, and they often possess temporal structure which are melodious to human ears [6]. Mating calls can be given as example to bird songs. Vocal bird sounds include distinctive spectral content often including harmonics. Alarm calls tend to be high-pitched with rapid modulations (to get maximum attention), whereas lower frequency calls are common in densely vegetated areas to avoid signal degradation due to reverberation [7]. Furthermore, depending on the environmental conditions (e.g. ambient noise level, vegetation density) and the bird species, bird sounds may exhibit certain local frequency shift variations [7]. Therefore, a BAD system should be able to capture melodic cues in time domain, and also should be robust to local frequency shifts.

Convolutional neural networks (CNN) are able to extract higher level features that are invariant to local spectral and temporal shifts. Recurrent neural networks (RNNs) are powerful in learning the longer term temporal context in the audio signals. In this work, we combine these two approaches in a convolutional recurrent neural network (CRNN) and apply it over spectral acoustic features for the BAD challenge. This method consists of slight modification (temporal max-pooling to obtain file-level estimation instead of frame-level estimation) and hyperparameter fine-tuning for the challenge over the CRNN proposed in [8], where it has provided state-of-the-art results on various polyphonic sound event detection and audio tagging tasks. Similar approaches combining CNNs

ISBN 978-0-9928626-7-1 © EURASIP 2017

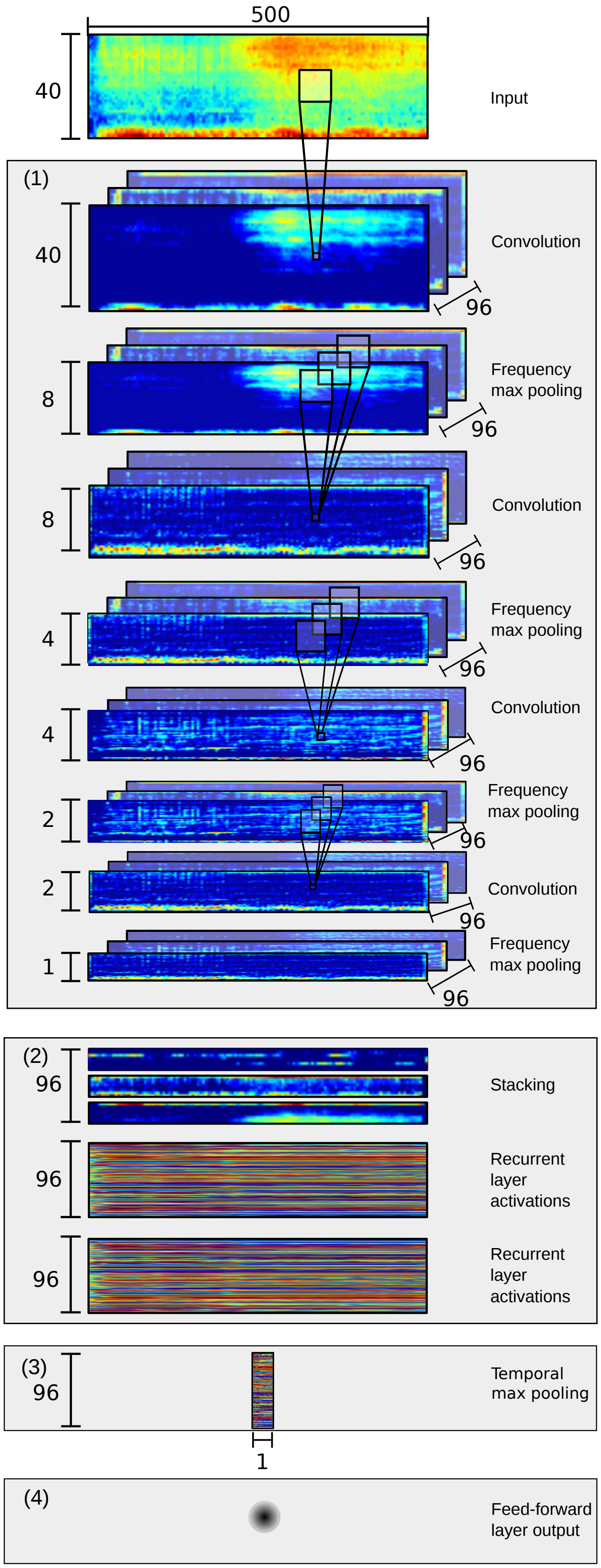

Fig. 1. Illustration of the CRNN architecture proposed for bird audio detection.

and RNNs have been presented recently in ASR [9] and music classification [10].

The rest of the paper is organized as follows. The employed acoustic features and the proposed CRNN for the BAD are presented in Section II. Dataset settings, metrics, and method configuration are reported in Section III. In Section IV are the results and their discussion, followed by the conclusions in Section V.

## II. Method

The proposed method consists of two stages. In the first stage, spectro-temporal features (spectrogram) are extracted from the raw audio recordings to be used as the sound representation. In the second stage, a CRNN is used to map the acoustic features to a binary estimate of bird song presence. CRNN parameters are obtained by supervised learning using material that consists of acoustic features extracted from a training database and the annotations of bird song activity.

### A. Features

The utilized spectro-temporal features are log mel-band energies, extracted from short frames. These features has been shown to perform well in various audio tagging and sound event detection tasks [11], [12], [8]. First, we obtained the magnitude spectrum of the audio signals by using short-time Fourier transform (STFT) over 40 ms audio frames of 50% overlap, windowed with Hamming window. Duration of each audio file in the challenge dataset is 10 seconds, resulting to 500 frames for each file. Then, 40 log mel-band energy features were extracted from the magnitude spectrum. Librosa library [13] was used in the feature extraction process.

Keeping in mind that bird sounds are often contained in a relatively small portion of the frequency range (mostly around 2-8 kHz), extracting features from that range seems like a good approach. However, experiments with features from the whole frequency range (from 0 Hz to Nyquist frequency of 22050 Hz) provided better results, and were therefore utilized in the proposed method.

### B. Convolutional recurrent neural networks

The CRNN proposed in this work, depicted in Figure 1, consists of four parts:

1) convolutional layers with rectified linear unit (ReLU) activations and non-overlapping pooling over frequency axis
2) gated recurrent unit (GRU) [14] layers
3) a temporal max-pooling layer, and
4) a single feedforward layer with a single unit and sigmoid activation, as the classification layer.

A time-frequency representation of the data is fed to the convolutional layers and the activations from the filters of the last convolutional layer are stacked over the frequency axis and fed to the first GRU layer. The extracted representations over each time frame (from the last GRU layer) are used as input to the temporal max-pooling layer. Output of the max-pooling layer is employed as input to the classification layer.

ISBN 978-0-9928626-7-1 © EURASIP 2017

Output of the classification layer is treated as the bird audio probability for the audio file. The aim of the network learning is to get the estimated bird audio probabilities as close as to their binary target outputs, where target output is 1 if any bird sound is present in a given recording, and 0 vice versa.

The network is trained with back-propagation through time using Adam optimizer [15] and binary cross-entropy as the loss function. In order to reduce overfitting of the model, early stopping was used to stop training if the validation data AUC score did not improve for 50 epochs. For regularization, batch normalization [16] was employed in convolutional layers and dropout [17] with rate 0.25 was employed in convolutional and recurrent layers. Keras deep learning library [18] has been used to implement the network.

The proposed method differs from our other submission [19] for the challenge (which came in fifth place) in the following ways: we use a single set of acoustic features, smaller max pool size in frequency domain and no max pooling in time domain in convolutional layers, no maxout activation for the classification layer, and the whole method consists of a single branch with unidirectional GRU. In addition, considering the auxiliary data augmentation and domain adaptation techniques applied in [19], the proposed method is less complex and still performs better in the given BAD challenge.

## III. Evaluation

### A. Datasets

The Bird Audio Detection challenge [4] consists of a development and an evaluation set. The development set consists of *freefield1010* (field recordings gathered by the [1]FreeSound project) and *warblr* (crowd-sourced recordings collected through smartphone app) datasets, and the evaluation set consists of *chernobyl* (collected by unattended recorders in Chernobyl exclusion zone) dataset. Recordings in all the datasets are around 10 seconds long, single channel, and sampled at 44.1 kHz. The annotations for the recordings are binary - bird calls present or absent. The total duration of the available recordings is approximately 68 hours, which makes the dataset a valuable source for detection methods that require large amount of material. The statistics of the datasets are presented in Table I.

From the development set, we create five different splits with 60% training, 20% validation, and 20% testing set distribution. Each split has an equal distribution of birds call present and absent, i.e. 60% of all the development data with present bird call annotation is included in training data, and the same is valid for absent bird call annotations. Different splits are obtained by randomly shuffling the recordings list and re-partitioning the data in given proportions. All development set results are the average performance over the splits. For the challenge submission, the CRNN is trained on single split of 80% training and 20% validation done on development set, with equal distribution of classes.

TABLE I
Bird audio detection challenge [4] dataset statistics

| Dataset | Bird call | | |
|---|---|---|---|
| | Present | Absent | Total |
| freefield1010 | 5755 | 1935 | 7690 |
| warblr | 1955 | 6045 | 8000 |
| chernobyl | ? | ? | 8620 |
| Total | 7710 + ? | 7980 + ? | 24310 |

TABLE II
Final hyperparameters used for the evaluation based on the validation results from the hyperparameter grid search.

| | Hyperparameters |
|---|---|
| # convolutional layers | 4 |
| Filter shape | 5-by-5 |
| pool size | (5,2,2,2) |
| # recurrent layers | 2 |
| # feature maps/hidden units | 96 |
| # Parameters | 806K |

### B. Evaluation Metric and Configuration

The BAD system output is evaluated from the receiver operating characteristic (ROC) using the AUC measurement. AUC is calculated from the area under the ROC curve that shows the true positive rate against false positive rate over various binarization threshold values.

In order to obtain the optimal hyperparameters for the given task, we run a hyperparameter grid search over the validation set. The grid search covers each of the combinations of the following hyperparameter values: the number of CNN feature maps/RNN hidden units (the same amount for both) $\{96, 256\}$; the number of recurrent layers $\{1, 2, 3\}$; and the number of convolutional layers $\{1, 2, 3 ,4\}$ with the following frequency max pooling arrangements after each convolutional layer $\{(4), (2, 2), (4, 2), (8, 5), (2, 2, 2), (5, 4, 2), (2, 2, 2, 1), (5, 2, 2, 2)\}$. Here, the numbers denote the number of frequency bands at each max pooling step; e.g., the configuration (5, 4, 2) pools the original 40 bands to one band in three stages: 40 bands $\rightarrow$ 8 bands $\rightarrow$ 2 bands $\rightarrow$ 1 band. The final network configuration is selected as the one with the best average validation set AUC score over the five splits, and the resulting parameters are given in Table II.

### C. Baseline

In this work, we trained a CNN to be used as a baseline and also to understand the benefit of using recurrent layers after the convolutional layers. Based on the information given after the challenge, most of the submissions also use CNN as their classifier, and therefore it can be deemed as an appropriate baseline for the proposed method. The optimal parameters for CNN is found with a similar grid search as explained in Section III-B, the only difference is that we replace the recurrent layers with feedforward layers. Each feedforward layer had shared weights between timesteps.

For comparison, we also provide the scores from the top three submissions for the challenge. Both methods use CNN as classifier (therefore labeled as *CNN** [20] and *CNN*** [21]),

[1] http://freesound.org/

ISBN 978-0-9928626-7-1 © EURASIP 2017

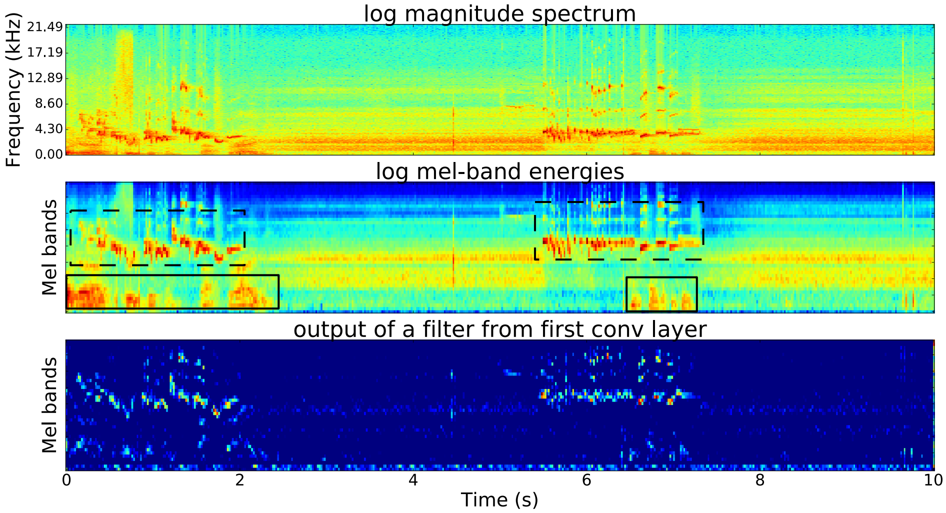

Fig. 2. Log magnitude spectrum (top), log mel-band energies (middle) and a single filter output from first convolutional layer (bottom) for *000a3cad-ef99-4e5e-9845.wav*. Dashed boxes mark the components due to bird sounds, and solid boxes mark the components due to two people speaking.

they use mel spectrogram as input features, and they apply frequency and time shift as data augmentation techniques. Both methods apply pseudo-labeling (i.e. including the very confident detections from the test set into training set) and they further apply model ensembling over the networks.

## IV. Results and discussion

AUC scores for the baseline CNN and the proposed CRNN methods on development and evaluation sets are presented in Table III. AUC for development set is obtained from the mean test AUC of the five splits. Although the performance difference between CNN and CRNN is minimal for the development data, CRNN performs significantly better for the evaluation data. Considering that the evaluation data includes recordings from different environmental and recording conditions than the development data, one can say that CRNN does a better job of generalizing over bird sounds in different conditions. For both methods, the validation data AUC score reaches to about 92% in the very first epoch and reaches its peak in about 20 epochs. To compare with the other top submissions, CNN* reaches 88.7% AUC and CNN** obtains 88.2% on the evaluation data.

In order to provide some insight on the features and network outputs, one of the recordings from the evaluation set (namely *000a3cad-ef99-4e5e-9845.wav*) has been specifically investigated. The top panel represents the magnitude spectrum (in log scale) for the recording, the middle panel shows the normalized log mel band energies which are used as input for the network, and the bottom panel represents the output from one of the filters in the first convolutional layer before max-pooling. When we compare the top two panels, we notice that with log mel band energies, the frequency components due to speech and bird sounds become very distinguishable. In addition, by looking at the filter outputs in the bottom panel, one can say that this filter has learned to react to the bird sound components and mostly ignore the rest for the given audio recording. The trained CRNN outputs a probability of 94.7% for a bird sound in this recording.

TABLE III
AUC scores on development and evaluation sets

| Dataset | Method | |
|---|---|---|
| | CNN | CRNN |
| Development | 95.3 | **95.7** |
| Evaluation | 85.5 | **88.5** |

Since the amount of available material is quite large (about 68 hours), we did not further experiment on various data augmentation techniques. For the challenge submission, we experimented with a model ensemble method: 11 networks with the same architecture and different initial random weights (obtained by sampling from different random seeds) were trained and the estimated probabilities from each network were averaged to obtain the ensemble output. Although this method improved the prior AUC results (calculated from a small portion of the evaluation data) from 88.3 to 89.4, it performed worse in the final results (88.5 vs. 88.2). The authors do not have a clear reasoning for this contradiction, other than the possibility that the prior evaluation data does not sufficiently represent the data distribution of the whole evaluation dataset.

## V. Conclusion

In this work, we propose using convolutional recurrent neural networks for bird audio detection as a part of a research challenge. The proposed method shows robustness for the local frequency shifts and is able to utilize longer term temporal information. Both of these features are essential for a generalized, context independent BAD system. The method achieves 88.5% AUC score and obtains the second place in Bird Audio Detection challenge.

ISBN 978-0-9928626-7-1 © EURASIP 2017

## Acknowledgment

The research leading to these results has been conducted with the funding from the European Research Council under the European Unions H2020 Framework Programme through ERC Grant Agreement 637422 EVERYSOUND. The authors also wish to acknowledge CSC-IT Center for Science, Finland, for computational resources.

## References

[1] R. T. Buxton and I. L. Jones, "Measuring nocturnal seabird activity and status using acoustic recording devices: applications for island restoration," *Journal of Field Ornithology*, vol. 83, no. 1, pp. 47–60, 2012.

[2] T. A. Marques, L. Thomas, S. W. Martin, D. K. Mellinger, J. A. Ward, D. J. Moretti, D. Harris, and P. L. Tyack, "Estimating animal population density using passive acoustics," *Biological Reviews*, vol. 88, no. 2, pp. 287–309, 2013.

[3] M. Graciarena, M. Delplanche, E. Shriberg, and A. Stolcke, "Bird species recognition combining acoustic and sequence modeling," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 341–344.

[4] D. Stowell, M. Wood, Y. Stylianou, and H. Glotin, "Bird detection in audio: a survey and a challenge," in *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2016, pp. 1–6.

[5] S. N. Howell and S. Webb, *A guide to the birds of Mexico and northern Central America*. Oxford University Press, 1995.

[6] P. Ehrlich, D. Dobkin, and D. Wheye, "Birds of stanford essays." [Online]. Available: http://web.stanford.edu/group/stanfordbirds/text/uessays/essays.html

[7] E. P. Derryberry, "Ecology shapes birdsong evolution: variation in morphology and habitat explains variation in white-crowned sparrow song," *The American Naturalist*, vol. 174, no. 1, pp. 24–33, 2009.

[8] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[9] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4580–4584.

[10] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," *arXiv preprint arXiv:1609.04243*, 2016.

[11] "Detection and classification of acoustic scenes and events (DCASE)," 2016. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-real-life-audio

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

[13] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, 2015.

[14] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014.

[15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *arXiv:1412.6980 [cs.LG]*, 2014.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," in *Journal of Machine Learning Research (JMLR)*, 2014.

[18] F. Chollet, "Keras," github.com/fchollet/keras, 2015.

[19] S. Adavanne, D. Konstantinos, E. Cakir, and T. Virtanen, "Stacked convolutional and recurrent neural networks for bird audio detection," in *European Signal Processing Conference (EUSIPCO)*, 2017, submitted.

[20] T. Grill, "Source code for the BAD challenge submission, user: bulbul," https://jobim.ofai.at/gitlab/gr/bird_audio_detection_challenge_2017/tree/master, 2017.

[21] T. Pellegrini, "Source code for the BAD challenge submission, user: topel," github.com/topel/bird_audio_detection_challenge, 2017.

ISBN 978-0-9928626-7-1 © EURASIP 2017