PROCEEDINGS of the
23rd International Congress on Acoustics
9 to 13 September 2019 in Aachen, Germany

New aspects in birdsong recognition utilizing the gabor transform

Sven HEUER[(1)], Pavel TAFO[(2)], Hajo HOLZMANN[(3)], Stephan DAHLKE[(4)]

[(1)]Philipps University Marburg, Germany, heuersv@mathematik.uni-marburg.de
[(2)]Philipps University Marburg, Germany, tafo@mathematik.uni-marburg.de
[(3)]Philipps University Marburg, Germany, holzmann@mathematik.uni-marburg.de
[(4)]Philipps University Marburg, Germany, dahlke@mathematik.uni-marburg.de

Abstract

In this paper we will be concerned with mathematical methods for birdsong recognition and classification. Current approaches compute the spectrogram of an audio recording using the Gabor transform, which is then used as input for a convolutional neural network (CNN) to classify the recording. While recent work is dedicated to finding the best hyperparameters for training the CNN and for data augmentation, the parameters for the Gabor transform and the signal detection methods receive less attention. We aim to close this gap by evaluating the effect of different window lengths on the overall classification accuracy. Additionally we propose a method for denoising signals based on quantiles to account for different signal to noise ratios in the dataset.

Keywords: bioacoustics, Gabor transform, deep learning

INTRODUCTION

To implement efficient wildlife conservation stategies it is important to accurately observe the conservation-relevant species. As part of this task, we aim to detect birdsongs and to classify the underlying species in a given sound file. The annual BirdCLEF challenge[1] provides a large scale comparison between current methods handling this task. The 2017 edition of this challenge was won by Kahl et al. [6] who made their code publicly available on GitHub[2]. For our approach we mainly adapted their method, but we introduced specific modifications at some crucial points. By proceeding this way, we are able to compare our results more directly to existing state-of-the-art methods.

Our resulting algorithm consists of the following steps. First, we split the given signal into overlapping chunks of five second length. These chunks are then analysed using the Gabor transform, a sampled version of the short-time Fourier transform. We use quantiles and a combination of morphological transformations to denoise our signal chunks dependent on the unknown signal to noise ratio, altering the approach suggested by [9]. In this step we also determine whether the given chunk contains a bird song at all. All the obtained denoised spectrograms are then augmented following [6] and fed into a convolutional neural network for classification. Our approach outperforms the comparative method on our self-assembled dataset.

The paper is organized as follows. In Section 2 we give an overview over the dataset we used for our experiments. Our method is then described in detail in section 3. Section 4 contains the setups for our experiments and our obtained results, which are summarized in Section 5. Here we also give an outlook into the open questions we wish to investigate in the future.

This work originated within the collaborative research project *Natur 4.0 | Sensing Biodiversity*, which aims at developing a modular environmental monitoring system for the high-resolution observation of conservation-relevant species, habitats and processes[3].

[1]https://www.imageclef.org/BirdCLEF2019 for the 2019 edition
[2]https://github.com/kahst/BirdCLEF2017
[3]https://www.uni-marburg.de/de/fb19/natur40

DATASET

Training a deep neural network from scratch requires a large dataset with many diverse classes. The BirdCLEF 2017 dataset [4] contains a total of 36.496 audio recordings for 1500 different bird species, which would make it a natural candidate for our training. However, training on such a large dataset takes a lot of time, which means that we would not be able to test all the different approaches we are interested in. Additionally the recordings in the BirdCLEF dataset contain bird species from South America while our primary goal within the Natur 4.0 project is to detect and identify birds in the Marburg Open Forest test area. The issue of scaling to larger datasets with more classes shall be addressed in a subsequent step.

To account for these problems, we assembled our own dataset containing the 74 bird species we expect to find in our test area. For every species we collected a varying number of audio recordings, resulting in a total of 1276 audio files. Altogether our dataset consists of over 26 hours of audio data. This small dataset turned out to be sufficient for our purpose of analysing different parts of the proposed method.

As with the BirdCLEF dataset, our dataset is built entirely from the Xeno-Canto collaborative database[4], a huge collection of audio recordings shared from all over the world. A copy of the audio files can be obtained on request from the authors.

METHOD

Our overall architecture follows Kahl et al. [6]. We start by splitting each audio file into chunks of five seconds with an overlap of one second. This way we can work with a unified input size for our neural network and we are able to cut out the parts of the recordings where the bird is not audible. In a first step, we generate the spectrogram for each five second chunk. Details are discussed in Section 3.1.

Next, in Section 3.2 we propose an adaptive denoising of the spectrograms using quantiles. In a third step, in Section 3.3 we propose a novel algorithm to remove those chunks which do not contain bird sounds from our training set. Our proposed algorithm also works for general speech detection.

Finally, the denoised spectrograms are fed into a convolutional neural network to classify the species of the birds that can be heard in the original five second chunk. Details are discussed in Section 3.4.

3.1 Generating spectrograms

To obtain time-frequency information about a signal f, we sample the short-time Fourier transform (STFT) on the grid $a\mathbb{Z} \times b\mathbb{Z}$ of $\mathbb{R} \times \mathbb{R}$, where the parameters $a > 0$ and $b > 0$ determine the distance between the grid points. Since our signals are compactly supported in both the time and the frequency domain, we can even store just a finite set of values without losing any information. The goal of the STFT is to find the frequency spectrum of the signal f at any given time x. This is done by localising f to a neighborhood of x using a window function $g \neq 0$ and taking the Fourier transform of the resulting function:

$$V_g f(x,\omega) = \int_{\mathbb{R}} f(t)\overline{g(t-x)}e^{-2\pi i t\omega}\mathrm{d}t, \;\; x,\omega \in \mathbb{R}.$$

Two questions arise from this definition: First, we have to find suitable sampling constants $a,b > 0$ when discetizing the STFT. This problem is easily solved by requiring a fixed spectrogram size to use as input for our neural network. Second, the window function g plays an essential role in the construction of the STFT. Indeed, it is well-known that the shape of the time-frequency spectrum depends quite sensitively on the properties of the window function under consideration. A simple selection might be a characteristic function with window length $L > 0$: $g(t) = 1$ if $t \in [-L/2, L/2]$, and $g(t) = 0$ otherwise. This way, to compute the STFT at time x the signal is simply cut to a neighborhood of length L around x. In fact, this is the choice made by Kahl et al. in [6] with a window length of 0.05 seconds.

[4] https://www.xeno-canto.org/

However, by using a characteristic function as our window function, we obtain an inadequate frequency resolution. We refer to [5] for a discussion of the theoretical background of the STFT and its discretized version, the Gabor transform, as well as the issue of choosing an appropriate window function. Consequently, to enhance the frequency resolution and to avoid artefacts, in this paper we use a smooth function g, for which g together with its Fourier transform $\hat{g}$ both decay rapidly. The Gaussian window function

$$g_a(x) = e^{-\pi x^2/a}$$

with $a > 0$ fulfills this requirement. In fact the time-frequency resolution based on the Gaussian window has certain optimality properties related to the Heisenberg uncertainty principle, see [5].

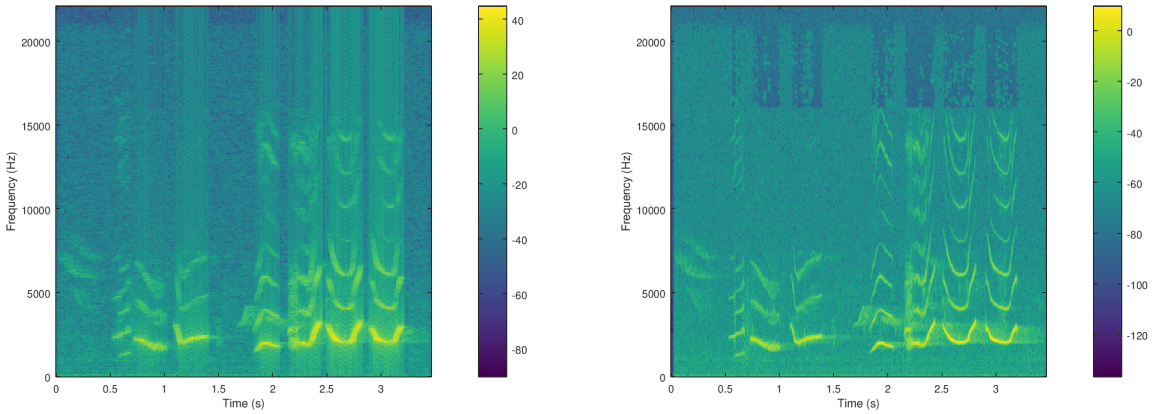

Figure 1. Log-scaled spectrogram of a blackbird song computed with the window function being a characteristic function with a window length of 0.05 seconds (left) and a Gaussian window g_1 (right).

Figure 1 shows the effect of different window functions on the time-frequency resolution of the Gabor transform. The absolute values of the Gabor coefficients were scaled logarithmically to show the differences more clearly. All spectrograms containing the absolute values of the Gabor coefficients were generated with the *ltfatpy* package[5], a partial Python port of the Large Time-Frequency Analysis Toolbox (LTFAT) [7]. This toolbox also provides a convenient tool to calculate the sampling constants we need, so that the spectrogram has the desired size for our neural network input.

3.2 Denoising

It seems obvious that the clearer the input spectrograms are, the easier it is to properly recognize their underlying patterns. Driven by this idea we have tried several methods to algorithmically remove the ambient noise. With the varying environmental conditions in mind we have focused on methods that are able to adapt to different types of noise. Specifically, we expect our signals to consist of clean speech degraded by statistically independent additive noise. The clean speech and the observed noise signal are modeled as realizations of processes with respective power spectra $S(\omega,t)$ and $N(\omega,t)$, as in [10]. As the speech and noise signal are assumed to be additive and independent, the power spectrum of the observed signal is $X(\omega,t) = S(\omega,t) + N(\omega,t)$. Denoising therefore consists of two steps: First we estimate the noise and then we remove it.

To this end we adopt the *back-end processing* approach by estimating the noise directly from the noisy speech signal. *Quantile-based* noise estimation methods make use of the fact that even in speech sections of the input signal not all frequency bands are permanently occupied with speech but rather exhibit high energy levels, see

[5] https://pypi.org/project/ltfatpy/

[10]. In fact, for a significant percentage of time the energy in each frequency band is on noise level. Thus the noise power spectrum for a given chunk is estimated from the observed speech signal by taking the q-th quantile with respect to the time variable, that is $N(\omega,t) = X(\omega,t_{\lfloor qT \rfloor})$. Since we calculate the quantile over each chunk, the computational cost and the memory consumption for estimating $N(\omega,t)$ remain low. The key benefit of quantile-based methods is their ability to adapt not only to the different types of stationary noise but also to non-stationary noise which may result from an abrupt increase in the noise energy. Furthermore, the quantile is not sensitive to outliers.

In the additive noise model, the clean speech can then be estimated by subtracting the estimated noise from the received signal, see [1]:

$$|S(\omega,t)|^2 = \max(0, |X(\omega,t)|^2 - |N(\omega,t)|^2).$$

Figure 2 shows the spectrogram of a whitethroat song before and after denoising.

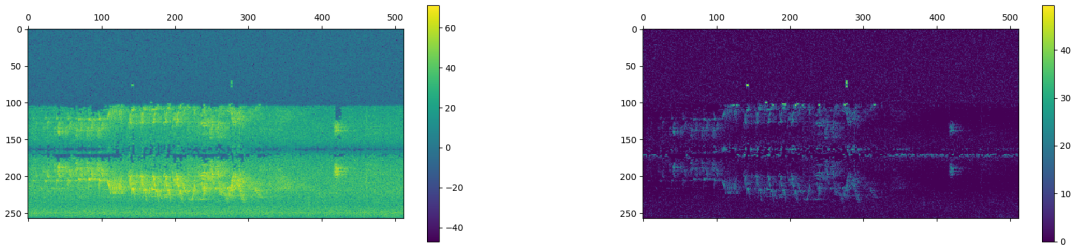

Figure 2. Comparison between a noisy spectrogram (left) and the corresponding denoised spectrogram based on quantiles with $q = 0.8$ (right).

3.3 Birdsong detection

Spectrograms containing birdsong can be viewed as a landscape. The mountainous parts represent changes of the energy level of some frequency at some time, which characterizes speech. The flat parts represent constant energy levels, which mostly characterize noise. Our idea for detecting speech is to get rid of most of the flat or low fluctuation part and to check whether the remaining mountainous part is dense enough. Our novel algorithm to detect chunks that contain birdsongs consists of the following three main steps.

1. For each chunk we create an additional spectrogram by taking $g_{0.01}$ as the window function. This choice results in an improved localization in time at the cost of a reduced frequency resolution. The increased time resolution is legitimated by the fact that we do not expect a bird to sing continuously in the whole chunk but to hit frequencies above the noise level at specific points in time. Due to different birds singing at different frequencies, increasing the frequency resolution has no added value.

2. We denoise the previously created spectrograms by setting $q = 0.7$ for the quantile method. We have run several tests showing that this rather high value for q works for almost all bird species we are dealing with. Although a huge amount of the signal is removed, the part left is still significant enough to perform the upcoming classification. This step generally works fine with values of q between 0.6 and 0.8 depending on how many pixels the noisy spectrogram shows.

3. We apply morphological filtering to detect consistent speech. This is based on the fact that speech regions of the spectrogram are identified on the proximity of high energy regions to neighbouring high energy regions in the three-dimensional space, see [3]. The combined morphological operations[6] of first

[6] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

erosion and then dilation are applied to a binary thresholded version of the degraded spectrogram to highlight probable speech regions. The process of erosion removes those sparse high energy regions. The remaining regions must have been dense before erosion. The process of dilation then tries to restore the neighborhood around these regions. The combination of the two techniques results in a non-linear, time-frequency filter.

Chuncks, where the spectrogram contains at least one pixel after this process, are then classified as containing birdsong. Our method depends on the following tuning parameters: the width a of the Gaussian window function in the Gabor transform and the quantile q for denoising, as well as the kernel size k and the number of iterations i for the morphological operations. Figure 3 illustrates the detection steps.

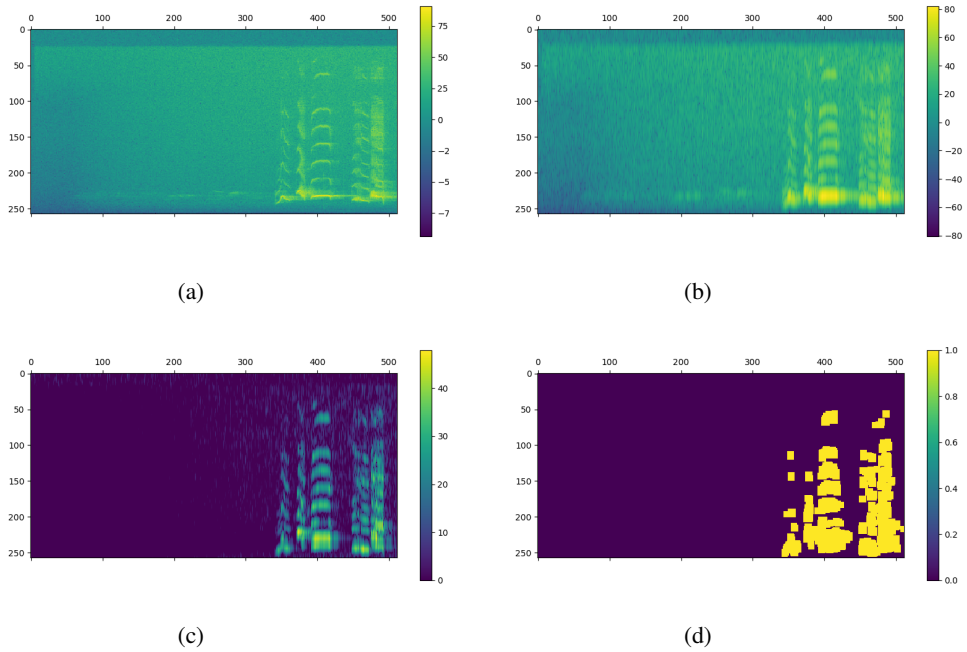

Figure 3. Detection steps: a) The original spectrogram, b) increased time resolution, $a = 0.01$, c) quantile based denoising, $q = 0.7$, d) Morphological filtering: 2x erosion followed by 2x dilation each with a 5×5 kernel.

3.4 Birdsong classification

To keep our results comparable to current state-of-the-art methods, we chose the same hyperparameters for our neural network training as in [6]. We therefore augmented our spectrograms by incorporating a random vertical roll of up to five percent and adding random Gaussian noise as well as pre-chosen noise samples. These samples are the same as the ones used in [6]. The size of the spectrograms as our network input was set to 512×256 pixels. Since the recordings in our dataset do not contain many background species, we chose not to use batch augmentation.

To keep the computation time as short as possible, we used the smallest model (model 3) from [6] as our CNN architecture. The resulting net consists of five convolutional layers with an increasing number of filters (from 32 up to 256) followed by two dense layers with 512 neurons each and the output layer. Since our dataset with 74 different classes is not as diverse as for example the BirdCLEF dataset, this rather small model proved to be sufficient for our cause. Five percent of our training set were used as a validation set.

As in [6], we trained for 55 epochs, decreasing our learning rate from 0.01 to 0.00001. Since we worked on the single-label szenario, we used categorical cross entropy as our loss function and ADAM updates for optimization. Our batch size was 128.

All training was done in *Python* using *Lasagne* [2], a library for training neural networks built on *Theano* [11]. Our generated spectrograms were saved as *NumPy* arrays, while the noise samples were processed using *OpenCV*. For the calculations we used a single NVIDIA Titan Xp graphics card. Our code is available on GitHub[7].

RESULTS

In a first experiment we were interested to see how our approach compares to state-of-the-art methods. First we applied the original method proposed by Kahl et al. [6] to the dataset under investigation in the present paper. We then incorporated our changes – using a Gaussian window, denoising the spectrogram with quantiles and changing the method for detecting bird sounds in the five second cunks – into this method and trained a new network with this altered approach.

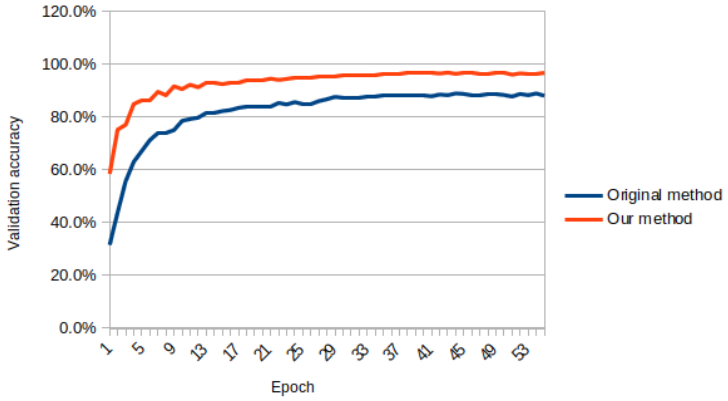

Figure 4. Progression of the accuracy achieved on the validation set during training.

Figure 4 shows how the accuracy on the validation set changed during training for both methods. The comparative method achieved an accuracy of at best 89.0%, while our approach led to an accuracy of up to 96.9%.

A comparison of the signal chunks chosen for training for both methods shows us that with our altered approach more chunks are identified as bird sound, increasing our total amount of spectrograms used for training from 71.470 to 83.014. To check whether the different compositions of the datasets led to different "glass ceilings" for the classification accuracy, we repeated our experiment with the larger model 1 from [6] pretrained on the BirdCLEF 2017 dataset. With this approach we achieved a validation accuracy of 96.6% with the comparative method and 97.7% with our method, indicating that in fact using a Gaussian window instead of a characteristic function and denoising the spectrogram has significant impact on the performance of our neural network.

We then tested different ways to feed the Gabor coefficients into the neural network. In our first run we simply used the absolute value of the coefficients obtained by using a Gaussian window with a time/frequency support ratio of 1. However, when displaying spectrograms, usually the logarithms of the squared absolute values are used to draw a connection to the volume of the signal. These representations show more detailed information to the human eye, so we trained a new neural network on the logarithmically scaled spectrograms. As an additional setup, we used the real and the imaginary part of the coefficients as separate input to also account for the phase instead of only looking at the energy when working with the Fourier transform of the signal.

[7] https://github.com/Heuersv/Birdsong-Classification

In these tests we always used a window with a time/frequency support ratio of 1. Since it was not clear whether a higher resolution in time or frequency might be advantageous, we tested a last setup with three types of spectrograms: One with a time/frequency support ratio of 1, one with a ratio of 0.01 (better time resolution) and one with a ration of 100 (better frequency resolution).

Table 1. Results of different approaches using a Gaussian window function.

Method	best validation accuracy	after ... epochs
Absolute values	96.9%	39
Log-scaled absolute values	95.9%	50
Real and imaginary part	96.4%	50
Abs. values, three resolutions	96.6%	48
Re. and im. party, three resolutions	96.4%	52

The results of all these experiments are summarized in table 1. We see that scaling the absolute values logarithmically does not yield the desired improvement, which is why we omitted this variant when using different resolutions. Apart from that, every approach led to very similar results with the original variant of using just the absolute values of the Gabor coefficients seemingly working best.

However, the differences aren't huge, indicating that a different approach might just work better when confronted with a more difficult dataset. Also, further experiments with the spectrograms generated using a characteristic function as the window function indicated that using a deeper net significantly improves the classification accuracy, though at the cost of longer computation time. We used a 13 layer deep neural network with an architecture inspired by the VGG16 architecture from Simonyan and Zisserman [8] and achieved a validation accuracy of 95.8%, an improvement of 6.8% compared to the more shallow model 3 from [6].

CONCLUSIONS

The results of this paper indicate that a deeper understanding of the mathematical foundation of the short-time Fourier transform can help to improve current benchmark algorithms for birdsong detection and classification. By using different window functions and improved denoising algorithms we were able to outperform current state-of-the-art algorithms in the setup investigated in this paper. Using denoised spectrograms for training our neural network also enabled us to compress our data without loss, saving disk space. We expect our results to extend to similar tasks in audio analysis.

In a next step, we aim to test our methods on large-scale datasets like the BirdCLEF datasets. Especially the soundscape scenario is of particular interest to us. Additionally we wish to examine different window functions for the Gabor transform, in particular those with compact support to reduce computation time. Lastly, using a convolutional neural network on the spectrogram might not be the best machine learning approach for audio analysis. We intend to use different architectures like recurrent neural networks in the future to account for the time dependency of audio signals.

ACKNOWLEDGEMENTS

This work was supported by the LOEWE priority project *Nature 4.0 | Sensing Biodiversity* funded by the Hessian Ministry for Research and Arts, Hesse, Germany.

REFERENCES

[1] Boll,S. Suppression of acoustic noise in speech using spectral subtraction. IEEE transactions on acoustic, speech and signal processing, vol 27, 1979, pp 113-120.

[2] Dieleman, S.; Schlüter, J.; Raffel, C.; Olson, E.; Sønderby, S.; Nouri, D.; others. Lasagne: First release. Zenodo, 2015.

[3] Evans, N.; Mason, J; Roach, M. Noise compensation using spectrogram morphological filtering. Proc. 4th IASTED internat. conf. signal image process, 2002, pp 157-161.

[4] Goëau, H.; Glotin, H.; Vellinga, W.; Planqué, R.; Joly, A. LifeCLEF bird identification task 2017. In CLEF 2017 working notes, 2017.

[5] Gröchenig, K. Foundations of time-frequency analysis, Applied and numerical harmonic analysis, Birkhäuser, 2001.

[6] Kahl, S.; Wilhelm-Stein, T.; Hussein, H.; Klinck, H.; Kowerko, D.; Ritter, M.; Eibl, M. Large-scale bird sound classification using convolutional neural networks. In Working notes on CLEF, 2017.

[7] Průša, Z.; Søndergaard, L. Holighaus, N.; Wiesmeyr, C.; Balasz, P. The large time-frequency analysis toolbox 2.0. Sound, music, and motion, lecture notes in computer science 2014, 2014, pp 419-442.

[8] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2015.

[9] Sprengel E.; Jaggi, M.; Kilchner, Y.; Hofmann, T. Audio based species identification using deep learning techniques. In Working notes on CLEF, 2016.

[10] Stahl, V.; Fischer, A; Bippus, R. Quantile based noise estimation for spectral subtraction and wiener filtering. Proc. (Cat. No.00CH37100) and signal processing 2000 IEEE int conf. acoustics, speech, vol 3, 2000, pp 1875-1878.

[11] Theano Development Team. Theano: A python framework for fast computation of mathematical expressions. arXiv:1605.02688, 2016.