Neural Computing and Applications (2020) 32:915–927
https://doi.org/10.1007/s00521-018-3626-7

S.I. : DEEP LEARNING FOR MUSIC AND AUDIO

Bioacoustic detection with wavelet-conditioned convolutional neural networks

Ivan Kiskin[1] · Davide Zilli[1,2] · Yunpeng Li[1] · Marianne Sinka[3] · Kathy Willis[3,4] · Stephen Roberts[1,2]

Received: 1 December 2017 / Accepted: 11 July 2018 / Published online: 1 August 2018

Abstract
Many real-world time series analysis problems are characterized by low signal-to-noise ratios and compounded by scarce data. Solutions to these types of problems often rely on handcrafted features extracted in the time or frequency domain. Recent high-profile advances in deep learning have improved performance across many application domains; however, they typically rely on large data sets that may not always be available. This paper presents an application of deep learning for acoustic event detection in a challenging, data-scarce, real-world problem. We show that convolutional neural networks (CNNs), operating on wavelet transformations of audio recordings, demonstrate superior performance over conventional classifiers that utilize handcrafted features. Our key result is that wavelet transformations offer a clear benefit over the more commonly used short-time Fourier transform. Furthermore, we show that features, handcrafted for a particular dataset, do not generalize well to other datasets. Conversely, CNNs trained on generic features are able to achieve comparable results across multiple datasets, along with outperforming human labellers. We present our results on the application of both detecting the presence of mosquitoes and the classification of bird species.

Keywords Convolutional neural networks · Spectrograms · Short-time Fourier transform · Wavelets · Acoustic signal processing · Classification and detection

1 Introduction

The timely and accurate detection of animals, birds and insects is of critical importance for conservation, ecology and epidemiology. We consider the effective analysis of the natural soundscape as a constituent component of this analysis. In this paper, we focus on bioacoustic classification, with a particular emphasis on mosquito detection. As part of showcasing the methods developed for this application, we describe how they can also, with minimal alteration, offer robust results in other bioacoustic classification domains.

Mosquitoes are responsible for hundreds of thousands of deaths every year due to their capacity to vector lethal parasites and viruses, which cause diseases such as malaria, lymphatic filariasis, zika, dengue and yellow fever [51, 52]. Their ability to transmit diseases has been widely known for over a hundred years, and several practices have been put in place to mitigate their impact on human life. Examples of these include insecticide-treated mosquito nets [7, 33] and insect sterilization

✉ Ivan Kiskin
ikiskin@robots.ox.ac.uk

Davide Zilli
dzilli@robots.ox.ac.uk

Yunpeng Li
yli@robots.ox.ac.uk

Marianne Sinka
marianne.sinka@zoo.ox.ac.uk

Kathy Willis
kathy.willis@zoo.ox.ac.uk

Stephen Roberts
sjrob@robots.ox.ac.uk

[1] Department of Engineering Science, University of Oxford, Oxford, UK

[2] Mind Foundry Ltd., Oxford, UK

[3] Department of Zoology, University of Oxford, Oxford, UK

[4] Royal Botanic Gardens, Kew, London, UK

techniques [4]. However, further progress in the battle against mosquito-vectored disease requires a more accurate identification of species and their precise location—not all mosquitoes are vectors of disease, and some non-vectors are morphologically identical to highly effective vector species. Current surveys rely either on human-landing catches or on less effective light traps. In part, this is due to the lack of cheap, yet accurate, surveillance sensors that can aid mosquito detection. Acoustic monitoring of mosquitoes proves compelling, as the insects produce a sound both as a by-product of their flight and as a means for communication and mating. Detecting and recognizing this sound is an effective method to locate the presence of mosquitoes and even offers the potential to categorize by species. Nonetheless, automated mosquito detection presents a fundamental signal processing challenge, namely the detection of a weak signal embedded in noise. Current detection mechanisms rely heavily on domain knowledge, such as tuning models to likely fundamental frequency and harmonics, and often extensive handcrafting of features, frequently similar to traditional speech representation methods. Over the last decade, there have been increasingly impressive performance gains achieved by the paradigm shift to deep learning, including bioacoustics [29]. An opportunity hence emerges to exploit and expand upon these advances to tackle our application problem.

Deep learning approaches, however, tend to be effective only once a critical number of training samples has been reached [9]. Consequently, data-scarce problems are not well suited to this paradigm. As with many other domains, the task of data labelling is expensive in both time requirement for hand labelling and associated ambiguity—namely that multiple human experts will not be perfectly concordant in their labels. Furthermore, recordings of free-flying mosquitoes in realistic environments are scarce [37] and hardly ever labelled.

This paper presents a novel approach for classifying events from acoustic data using scarce training data. Our approach is based on a convolutional neural network classifier conditioned on wavelet representations of the raw data. By exploiting the high sample rates of audio recordings, we are able to create sufficient training data for deep learning to remain highly effective. The network architecture and associated hyperparameters are, however, still strongly influenced by constraints in dataset size. We compare our methods to well-established classifiers, trained on both handcrafted features and the short-time Fourier transform (STFT), as well as human-made labels of mosquito audio recordings. We show that wavelet-conditioned CNN classifications are consistently made more accurately and confidently than on the STFT. The majority of our algorithms are able to more reliably detect mosquitoes (with accuracy above 90%) than human labellers, where only 70% of labels are in full agreement amongst four labellers.

Furthermore, without additional hyperparameter tuning we demonstrate that our approach scales well to different data domains, a transfer that traditional handcrafted features or classifiers struggle to make. Highlighting the generic nature of the solution we propose, we show that the CNN is also able to extract feature representations that allow it to distinguish between nine species of birds with reliably high accuracy (over 90%), similarly from very little data.

The remainder of this paper is structured as follows. Section 2 addresses related work, explaining the motivation and benefits of our approach. Section 3 details the method we adopt, providing insight into the relative strengths of wavelet transforms. Section 4 describes the experimental setup. Section 5 highlights the value of the method. We visualize and interpret the predictions made by our algorithm on unseen data in Sect. 6 to help reveal informative features learned from the representations and verify the method. Finally, we suggest further work and conclude in Sect. 7.

2 Related work

The use of artificial neural networks in acoustic detection and classification of species dates back to at least the beginning of the century, with the first approaches addressing the identification of bat echolocation calls [40]. Both manual and algorithmic techniques have subsequently been used to identify insects [10, 53], elephants [15], delphinids [39] and other animals. The benefits of leveraging the sound animals produce—both actively as communication mechanisms and passively as a result of their movement—is clear: animals themselves use sound to identify prey, predators and mates. Sound can therefore be used to locate individuals for biodiversity monitoring, pest control, identification of endangered species and more.

This section will therefore briefly review the use of machine learning approaches in bioacoustics. We describe the traditional feature and classification approaches to acoustic signal detection. In contrast, we also present the benefit of feature extraction methods inherent to current deep learning approaches. Finally, we narrow our focus down to the often overlooked wavelet transform, which offers significant performance gains in our pipeline.

2.1 Applications

The employment of artificial neural networks has proven successful for over a decade. In Chesmore and Ohya [10], a neural network classifier was used to discriminate four species of grasshopper recorded in northern England, with accuracy surpassing 70%. Other classification methods include Gaussian mixture models [41, 44] and hidden Markov models [34, 53], applied to a variety of different features extracted from recordings of singing insects. The work of Chen et al. [9] attributes the stagnation of automated insect detection accuracy to the sole use of acoustic devices, which are often not capable of producing a signal sufficiently clean to be classified correctly. In their work, they replace microphones with optical sensors, recording mosquito wingbeat through a laser beam hitting a phototransistor array—an extension of the method proposed by Moore et al. [36]. In a real-world setting, the resultant signals have a higher signal-to-noise ratio than those recorded acoustically. We regard these approaches and acoustic sensors as complementary, rather than competitors, and note that approaches which work well for acoustic detection can also be used to perform detection in other datasets, including optically sensed data, as well as other bioacoustic problems.

Whichever technique is used to record a mosquito wingbeat frequency, the need arises to be able to identify the insect's flight in a (more or less) noisy recording. The following section therefore reviews recent achievements in feature representation and learning, in the broad context of practical acoustic signal classification.

2.2 Feature representation and learning

The process of automatically detecting an acoustic signal in noise typically consists of an initial preprocessing stage, which involves cleaning and de-noising the signal itself, followed by a feature extraction process, in which the signal is transformed into a format suitable for a classifier, followed by the final classification stage. Historically, audio feature extraction in signal processing employed domain knowledge and intricate understanding of digital signal theory [28], leading to handcrafted feature representations.

Many of these representations often recur in the literature. A powerful, though often overlooked, technique is the wavelet transform, which has the ability to represent multiple time-frequency resolutions [2, Chapter 9]. An instantiation with a fixed time-frequency resolution thereof is the Fourier transform. The Fourier transform can be temporally windowed with a smoothing window function to create a short-time Fourier transform (STFT). Mel-frequency cepstral coefficients (MFCCs) create lower-dimensional representations by taking the STFT, applying a nonlinear transform (the logarithm), pooling and a final affine transform. A further example is presented by linear prediction cepstral coefficients (LPCCs), which pre-emphasise low-frequency resolution and thereafter undergo linear predictive and cepstral analysis [1].

Detection methods have fed generic STFT representations to standard classifiers [42], but more frequently complex features and feature combinations are used, applying dimensionality reduction to combat the curse of dimensionality [32]. Complex features (e.g., MFCCs and LPCCs) were originally developed for specific applications, such as speech recognition, but have since been used in several audio domains [35]. Humphrey et al. [28] argue that using features specifically developed for a prior application is unsustainable and has contributed to the stagnation in the field of audio event recognition.

On the contrary, the deep learning approach usually consists of applying a simple, general transform to the input data and allowing the network to both learn a feature representation and perform classification. This enables the models to learn salient, hierarchical features from raw data. The automated deep learning approach has recently featured prominently in the machine learning literature, showing impressive results in a variety of application domains, such as computer vision [31] and speech recognition [32]. However, deep learning models such as convolutional and recurrent neural networks are known to have a large number of parameters and hence typically require large data and hardware resources. Despite their success, these techniques have only recently received more attention in the time series signal processing literature.

A prominent example of this shift in methodology is the BirdCLEF bird recognition challenge. The challenge consists of the classification of bird songs and calls into up to 1500 bird species from tens of thousands of crowd-sourced recordings. The introduction of deep learning has brought drastic improvements in mean average precision (MAP) scores. The best MAP score of 2014 was 0.45 [23], which was improved to 0.69 the following year when deep learning was applied, outperforming the closest scoring handcrafted method by 19% [29]. The impressive performance gain came from the utilization of well-established convolutional neural network practice from image recognition. By transforming the signals into STFT spectrogram format, the input is represented by 2D matrices, which are used as training data. The following year saw a further jump to 0.71 [46] by utilizing transfer learning of the Inception-v4 deep convolutional neural network which was highly successful in ImageNet. Alongside this example, the most widely used base method to transform the input signals is the STFT [26, 43, 45].

An alternative feature transformation can be obtained with wavelets. Gaining popularity in the late 1990s, wavelets have been applied successfully to efficient image compression [12, JPEG 2000], de-noising [20], and have shown an ability to form efficient multi-resolution representations [17]. These properties have led to the use of wavelets in deep learning in two general ways. In one, wavelets are used as a preprocessing step to form noise-robust representations of time series, while in the second wavelets are employed to replace neurons to form wavelet neural networks. An example application of the former used Haar wavelets for stock price time series forecasting with recurrent neural networks [27]. In the latter scenario, wavelet neural networks have seen some success in time series prediction [8], signal classification and compression [30], but a lack of standard representations and general frameworks has prevented wider adoption [3].

As a result, to the best of our knowledge, the wavelet transform is rarely used as the representation domain for a *convolutional* neural network. In the following section, we present our method, which leverages the benefits of the wavelet transform demonstrated in the signal processing literature, as well as the ability to form hierarchical feature representations for deep learning.

3 Methods

We present a novel wavelet transform-based convolutional neural network architecture for the detection of events in noisy audio recordings. As our results of Sect. 5 indicate superior performance when training on wavelet representations of the data, we describe in depth the wavelet transform to provide insight into its benefits over the conventional STFT. We explain the wavelet transform in the context of the algorithm, thereafter describing the neural network configurations and a range of traditional classifiers against which we assess performance. The key steps of the feature extraction and classification pipeline are given in Algorithm 1.

3.1 The wavelet transform

We begin by discussing the details of the transform used in Step 2 of Algorithm 1 as a base to further extract features. A well-established approach in signal processing is the Fourier transform, which can be used to express any signal with an infinite series of sinusoids and cosines. Its main disadvantage is the provision only of frequency resolution, meaning one can identify all the frequencies present in a signal, but not their occurrence in time. To overcome this, common approaches include cutting the signal into sections of time and treating each segment separately. This action, however, smears out frequencies, especially in the case of short windows. A wide window is able to provide better frequency resolution at the sacrifice of time resolution. Choosing a window function therefore limits one to a fixed time-frequency resolution. The uncertainty in time-frequency is referred to as the Heisenberg-Gabor limit [6] which is derived from the notion that the product of the precision in time and frequency is limited.

The wavelet transform employs a fully scalable modulated window which provides a principled solution to the windowing function selection problem [49]. The window is slid across the signal, and for every position a spectrum is calculated. The procedure is then repeated at a multitude of scales, providing a signal representation with multiple time-frequency resolutions. This allows the provision of good time resolution for high-frequency events, as well as good frequency resolution for low-frequency events, which in practice is a combination best suited to real signals.

We choose to use the continuous wavelet transform (CWT) due to its successful application in time-frequency analysis [18]. The CWT is particularly well suited over the discrete wavelet transform to time-frequency analysis as redundancy makes information available in peak shape and peak composition more visible and easier to interpret [21]. The CWT can be written in the time domain as:

Algorithm 1 Detection Pipeline

```
1: Load $N$ labelled microphone recordings $x_1(t), x_2(t), \ldots, x_N(t)$.
2: Take transform with $h_1$ features such that we form a feature tensor $\mathbf{X}_{\text{train}}$ and corresponding label vector $\mathbf{y}_{\text{train}}$:
       $$\mathbf{X}_{\text{train}} \in \mathbb{R}^{N_S \times h_1 \times w_1}, \mathbf{y}_{\text{train}} \in \mathbb{R}^{N_S \times n},$$
   where $N_s$ is the number of training samples formed by splitting the transformed recordings into 2D ‘images’ with dimensions $h_1 \times w_1$, and $n$ is the number of classes.
3: Train classifier on $\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}$.
4: For test data, $\mathbf{X}_{\text{test}}$, neural network outputs a prediction $\hat{y}_i$ for each class $C_i$:
       $$0 \leq \hat{y}_i(\mathbf{x}) \leq 1, \quad \text{such that} \quad \sum_{i=1}^{n} \hat{y}_i(\mathbf{x}) = 1.$$
```

$$a(s,\tau) = |s|^{-1/2} \int_{-\infty}^{\infty} f(t)\psi^*\left(\frac{t-\tau}{s}\right) \mathrm{d}t, \tag{1}$$

or equivalently in the frequency domain as:

$$a(s,\tau) = |s|^{1/2} \int_{-\infty}^{\infty} F(\omega)\Psi^*(s\omega)e^{i\omega\tau} \,\mathrm{d}\omega, \tag{2}$$

where s is the scale factor, τ is the translation factor, $|s|^{-1/2}$ is the energy normalization factor, and $*$ denotes complex conjugation. The wavelets are generated by scaling and translating a single mother wavelet $\psi(t)$. Through continuous dilation in τ, the resulting CWT coefficients $a(s,\tau)$ can be assembled for a multitude of scales to either reconstruct the signal with an inverse transform or to create a spatial representation, called the scalogram. An equivalent representation in Fourier space requires the continuous application of 1-D Fourier transforms with windows that are translated in time. We can illustrate this by substitution of $\psi^*_{s,\tau}(\frac{t-\tau}{s}) = e^{-i\omega t}$ in Eq. 1. Essentially, this is equivalent to using a fixed basis with $s = 1$ and ignoring the dilation in τ. We thereby emphasise the more principled solution employed with the CWT, eliminating the need to choose and parameterize the window function necessary for STFT representations. Furthermore, working with the CWT one is free to choose a wavelet function with properties and characteristics that best suit the data, given knowledge of the signal being analysed. A popular choice of wavelet function for time-frequency analysis is given by the bump wavelet [50], expressed in the Fourier domain as:

$$\Psi(s\omega) = \exp\left(1 - \frac{1}{1-(s\omega-\mu)^2/\sigma^2}\right)\mathbb{I}[(\mu-\sigma)/s, (\mu+\sigma)/s], \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator function. Valid values for μ, σ are [3, 6], [0.1, 1.2], respectively. Smaller values of σ result in a wavelet function spanning a narrower frequency bandwidth (Fig. 1a), which results in superior frequency localization but poorer time localization. The bump wavelet is symmetric in frequency and has a direct relationship between wavelet scale and centre frequency, which we illustrate in Fig. 1b. As a result, we can create spectrograms in frequency which retain clear interpretability (which becomes important for Sects. 4, 6).

The spatial features thus created are then passed to the classifiers in the next step of the algorithm. We discuss neural network and more traditional implementations separately in the upcoming sections.

3.2 Neural network configurations

In this subsection, we start by providing definitions for the layers and parameters used in our convolutional neural network model. Thereafter, we describe how they were used in experimental setting.

A convolutional layer $H_{\text{conv}}\colon \mathbb{R}^{h_1 \times w_1 \times c} \to \mathbb{R}^{h_2 \times w_2 \times N_k}$ with input tensor $\mathbf{X} \in \mathbb{R}^{h_1 \times w_1 \times c}$ and output tensor $\mathbf{Y} \in \mathbb{R}^{h_2 \times w_2 \times N_k}$ is given by the sequential application of N_k learnable convolutional kernels $\mathbf{W}_p \in \mathbb{R}^{k \times k}, p < N_k$ to the input tensor. Given our single-channel ($c = 1$) input representation of the signal $\mathbf{X} \in \mathbb{R}^{h_1 \times w_1 \times 1}$ and a single kernel $\mathbf{W}_p$, their 2D convolution $\mathbf{Y}_k$ is given by [24, Chapter 9]:

$$\mathbf{Y}_k(i,j) = \mathbf{X} * \mathbf{W}_p = \sum_{i'}\sum_{j'} \mathbf{X}(i-i', j-j')\mathbf{W}_p(i',j'). \tag{4}$$

The N_k individual outputs are then passed through a non-linear function ϕ and stacked as a tensor $\mathbf{Y}$. Conventional choices for the activation ϕ include the sigmoid function, the hyperbolic tangent and the rectified linear unit (ReLU).

The data size constraint results in an architecture choice (Fig. 2) of few layers and free parameters. Our network consists of an input layer connected sequentially to a single convolutional layer and a fully connected layer, which is connected to the two output classes with dropout [48] with probability p. ReLU activations are employed based on their desirable training convergence properties [31]. Finally, we perform grid search over potential candidate hyperparameters using tenfold cross-validation on a subset of the mosquito training data. We show the results of these in Sect. 4.2. The combination of cross-validation and dropout helps avoid overfitting to our scarce data environment. This is shown by the excellent performance transfer with no hyperparameter re-tuning in Sect. 5.

3.3 Traditional classifier baseline

As a baseline, we compare the neural network models with more traditional classifiers that typically require explicit feature design. We choose three candidate classifiers widely used in machine learning with audio: random forests (RFs), naïve Bayes' (NBs) and support vector machines using a radial basis function kernel (RBF-SVMs). Their popularity stems from ease of implementation, reasonably quick training and competitive performance [47], especially in data-scarce problems. For brevity, we present results with the best-performing of these only, namely the SVM.

We selected ten features to encode the observed raw data: STFT spectrogram slices with 256 coefficients (created with a Hanning window and 256 samples of overlap), 13 MFCCs, entropy, energy entropy, spectral entropy, flux, roll-off, spread, centroid, and the zero crossing rate (for a detailed explanation of these features, see for example the open-source audio signal analysis toolkit by

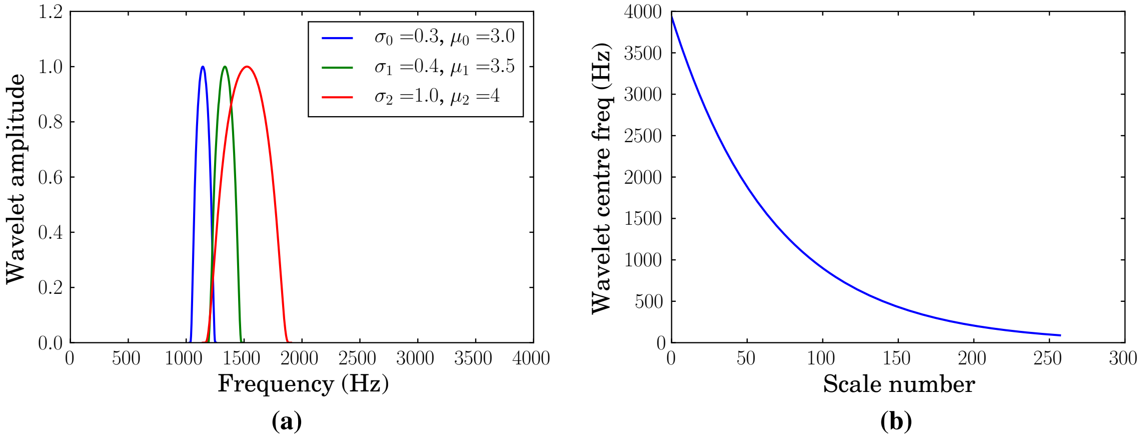


Fig. 1 Illustration of key properties of the bump wavelet, constructed from Eqs. 2 and 3. **a** Bump mother wavelets of fixed scale, s_{10}, with varying values of μ, σ, constructed from Eq. 3. **b** By converting wavelet scale to frequency, $f = (\frac{1}{s}\frac{\mu}{2\pi})$, we can illustrate the tiling of the frequency plane with the bump wavelet

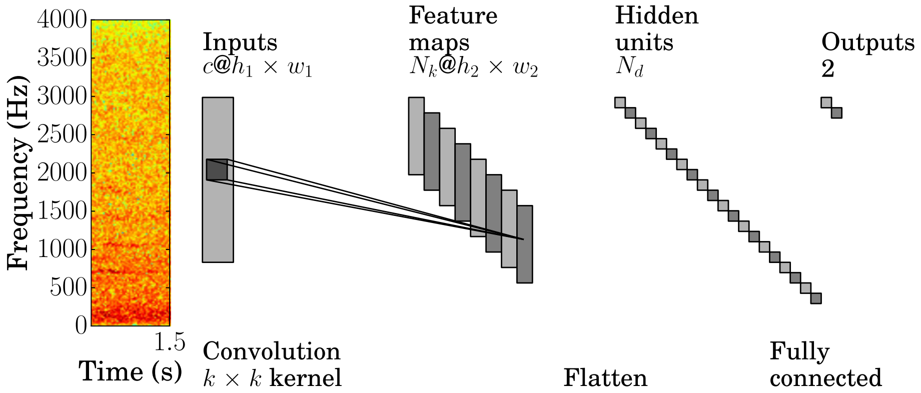


Fig. 2 The CNN pipeline. 1.5 s spectrogram of mosquito recording is partitioned into images with $c = 1$ channels, of dimensions $h_1 \times w_1$. This serves as input to a convolutional network with N_k filters with kernel $\mathbf{W}_p \in \mathbb{R}^{k \times k}$. Feature maps are formed with dimensions reduced to $h_2 \times w_2$ following convolution. These maps are fully connected to N_d units in the dense layer, fully connected to 2 units in the output layer

Giannakopoulos [22]). We note that our choice of feature parameters is based on past literature [19, STFT], [38, MFCCs], as well as empirical evidence. Prior parameterization of the feature space is necessary to some extent, as the number of feature and classifier parameters grows combinatorially to the point where joint optimization of all possible variables is infeasible. We select certain aspects of classifier-feature pipelines by cross-validation as detailed in Sect. 4.2.

4 Experimental details

4.1 Datasets

The mosquito data used here were recorded in January 2016 within culture cages containing both male and female *Culex quinquefasciatus*. Mosquito wingbeat sounds commonly have a fundamental frequency in the range of 150–750 Hz [14]. In noisy recording conditions, higher harmonics are less audible due to the sharper fall-off of shorter wavelength waves. Furthermore, the signals are sampled with inexpensive smartphone microphones to allow widespread deployment at low cost. Given the quality of these microphones, we observe empirically that sound emitted by mosquitoes mostly disappears in noise for frequencies higher than the third harmonic. We therefore choose to sample at $F_s = 8$ kHz. Figure 3 shows a frequency domain excerpt of a particularly faint recording in the windowed frequency domains. For comparison, we also illustrate the wavelet scalogram taken with the same number of scales as frequency bins, h_1, in the STFT. We plot the logarithm of the absolute value of the derived coefficients against the spectral frequency of each feature representation. Figure 3 (lower) shows the classifications within $y_i = \{0, 1\}$: absence, presence of mosquito, as

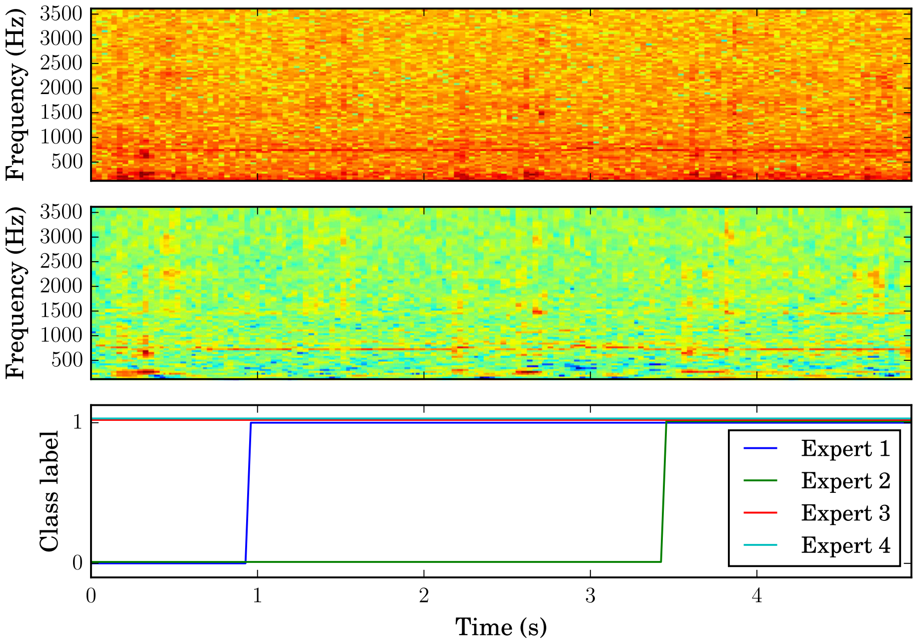


Fig. 3 STFT (top) and wavelet (middle) representations of signal with $h_1 = 256$ frequency bins and wavelet scales, respectively. Corresponding varying class labels (bottom) as supplied by human labellers. The wavelet representation shows greater contrast in horizontal constant frequency bands that correspond to the mosquito tone

labelled by four individual human researchers. These labels are created in version 2.2.2 of Audacity [5] with access to the recording audio and a matching spectrogram visualization. Of these, one particularly accurate label set created with great care under ideal conditions is taken as a gold standard reference to both train the algorithms and benchmark with the remaining experts. The classifications are restricted to only 2 classes due to the absence of labelled data in a multi-species scenario.

4.2 Cross-validated parameter search

In this section, we describe the experiment design and choice of hyperparameters, optimized to maximize F_1 score over a cross-validation sample. The available 57 mosquito recordings were split into 50% training and 50% held out data. The training data was then further split tenfold to perform cross-validation, creating approximately 3000–30,000 training samples, for window widths $w_1 = 10$ and $w_1 = 1$ samples, respectively. The neural networks were trained with a batch size of 256 for 20 epochs, according to validation accuracy in conjunction with early stopping criteria.

For fair comparison, we partition the data (choose window length w) to the strengths of each individual classifier. When evaluating cross-validation performance over the label interval, our hyperparameter optima ($w_1 = 10$ for the CNN, and $w_1 = 1$ for the SVM) given in bold in Table 1 suggest stacking the windows together creates feature vectors that lead to performance degradation for the SVM. Therefore, for each CNN input image with height $h_1 = 256$ and width $w_1 = 10$, our SVM will use 10 training samples with $w = 1, h = 256$ instead. Optimum window widths may vary with the dynamics of the signal, so it is important to consider this parameter systematically. In particular, should significantly more data be available, the drawbacks due to the use of larger windows (decrease in training samples the classifier sees) would be mitigated by the higher number of training samples at disposal. Conversely, the advantage of longer windows lies with supplying a longer temporal context.

The traditional classifiers are cross-validated with principal component analysis (PCA), and recursive feature elimination [25, RFE], with the number of components controlled by n and m, respectively. The best-performing feature set for all traditional classifiers is the set extracted by cross-validated RFE, outperforming all PCA reductions for every classifier-feature pair. The highest scoring hyperparameter, $m = 27$, defines a feature set, that we denote as RFE_{88}, which retains 88 dimensions from the ten original features spanning 304 dimensions ($F_{10} \in \mathbb{R}^{304}$).

4.3 Computational details

We consider computational complexity by splitting the pipelines into three processing stages: feature transformation, classifier training and classifier prediction. The overall compute time is thus the sum of all three. We further break this down for individual pipelines, noting that various software libraries can differ significantly in processing time for the same transformation.[1] We offer some insights from the figures given in Table 2 in this subsection.

[1] We provide our data and implementation on http://humbug.ac.uk/kiskin2018/.

Table 1 Results for grid search over hyperparameter values

Classifier	Features	Parameter grid	F_1 score
CNN	STFT	$w_1 \in \{1, \mathbf{10}, 100\}$, $k \in \{2, \mathbf{3}, 4, 5\}$, $N_k \in \{8, 16, 32, \mathbf{64}, 128, 256, 1028, 2056\}$, $N_d \in \{64, \mathbf{128}, 256, 512, 1024\}$	0.861 ± 0.066
CNN	Wavelet	$w_1 \in \{1, \mathbf{10}, 100\}$, $k \in \{2, 3, 4, \mathbf{5}\}$, $N_k \in \{8, 16, 32, \mathbf{64}, 128, 256, 1028, 2056\}$, $N_d \in \{64, \mathbf{128}, 256, 512, 1024\}$	0.915 ± 0.023
NB, RF, SVM	$F_{10} \in \mathbb{R}^{304}$	$w_1 \in \{\mathbf{1}, 10, 100\}$, $\text{PCA} \in \mathbb{R}^N, N \in 0.8^n \times 304$, $n \in \{0, 1, \ldots, 12\}, \text{RFE} \in \mathbb{R}^M, M \in 304 - 8m$, $m \in \{0, 1, \ldots, \mathbf{27}, \ldots, 35\}$	0.880 ± 0.055

The cross-validated F_1 score is reported for optimal hyperparameters found (in bold)

Table 2 Execution time given in seconds for feature-classifier pipelines trained on 15 min (900 s) and evaluated over 15 min of audio data sampled at 8000 Hz

Pipeline	Dimensions	Feature transform (s)	Classifier training (s)	Classifier prediction (s)
SVM MFCC	13	1.5	17.0	3.6
SVM RFE_{88}	88	5.2	42.9	8.4
SVM STFT	256	1.4	121.1	32.7
SVM Wavelet	256	170.6	131.4	28.9
CNN MFCC	13	1.5	16.0	1.0
CNN RFE_{88}	88	5.2	24.0	1.0
CNN STFT	256	1.4	37.5	4.0
CNN Wavelet	256	170.6	44.8	7.5

Run times are an average of three passes on a mid-range desktop with an Intel i7-4790k CPU with 16 GB DDR4 RAM and an NVIDIA GTX 970 GPU

The native training complexity of the RBF-SVM is stated as $O(n_{\text{SV}} d)$, where d is the input dimension and n_{SV} is the number of support vectors [13]. Table 2 shows that an increase in d coupled with the already large number of training samples (leading to a large n_{SV}) causes a significant slowdown in both training and prediction with the SVM. A feature dimension reduction, as encountered with the MFCC or RFE approaches, while slightly more costly as a preprocessing step, speeds up the training and prediction significantly.

The CNN was trained in Keras [11], with an NVIDIA 970 GTX GPU. This allows quick training and prediction, resulting in much shorter computation times than those of the `scikit-learn` SVM (running on a CPU) when working with a large feature space.

Furthermore, the CWT is highly redundant and so incurs a greater computational cost. Its computational complexity increases linearly with number of wavelet scales (provided sufficient RAM). Despite this, the sum of feature transformation and training time is well under the length of the audio recordings, suggesting real-time detection to be perfectly feasible given appropriate hardware. A significant reduction in the CWT processing time can be achieved by calculating each wavelet scale in parallel, due to the independence of each computation per scale. Further considerable speed-up can be achieved by utilizing a discrete wavelet transform or a fast wavelet transform [16]. While not the focus of this paper, this may be worth considering when transferring algorithms to embedded devices and specialized hardware in future work.

5 Classification performance

The performance metrics are defined at the resolution of the supplied label interval (0.1 s granularity) and presented in Table 3 for the mosquito dataset, and in Table 4 and Fig. 4 for the BirdCLEF subset. We highlight three key results in both applications.

5.1 Mosquito detection

Firstly, both the traditional and deep learning algorithms accurately and reliably detect mosquitoes, far surpassing human labellers in both F_1 score and precision-recall (PR) area. Since human labels were supplied as absolute (either $\hat{y}_i = 1, \hat{y}_i = 0$), an incorrect label incurs a large penalty on

Table 3 **Mosquito detection**: summary classification metrics reported as means ± the standard deviation from $n = 30$ random hold out dataset splits with 50% training data, and 50% test data

Classifier	Features	F_1 score	TPR	TNR	PR area
CNN	MFCC	0.895 ± 0.022	0.89 ± 0.04	0.90 ± 0.03	0.963± 0.012
SVM	MFCC	0.880 ± 0.020	0.88 ± 0.02	0.88 ± 0.02	0.951 ± 0.013
CNN	**$\mathbf{RFE_{88}}$**	**0.922 ± 0.019**	**0.93 ± 0.02**	**0.91 ± 0.04**	**0.980 ± 0.007**
SVM	RFE_{88}	0.904 ± 0.020	0.91 ± 0.02	0.90 ± 0.02	0.963 ± 0.013
CNN	STFT	0.883 ± 0.031	0.86 ± 0.05	0.91 ± 0.02	0.939 ± 0.017
SVM	STFT	0.858 ± 0.031	0.80 ± 0.05	0.91 ± 0.02	0.889 ± 0.036
CNN	Wavelet	0.913 ± 0.020	0.92 ± 0.02	0.91 ± 0.02	0.962 ± 0.012
SVM	Wavelet	0.897 ± 0.020	0.90 ± 0.02	0.90 ± 0.02	0.944 ± 0.012
Labeller 1	Audacity	0.819 ± 0.018	0.89 ± 0.02	0.85 ± 0.02	0.843 ± 0.006
Labeller 2	Audacity	0.856 ± 0.019	0.92 ± 0.03	0.88 ± 0.02	0.873 ± 0.008
Labeller 3	Audacity	0.852 ± 0.018	0.77 ± 0.02	0.98± 0.02	0.901 ± 0.007

Table 4 **BirdCLEF subset**: summary classification metrics reported as means ± the standard deviation from $n = 30$ random dataset splits with 50% training data, and 50% test data

Classifier	Features	F_1 score	PR area
CNN	MFCC	0.860 ± 0.028	0.915 ± 0.031
SVM	MFCC	0.891 ± 0.029	0.931 ± 0.029
CNN	RFE_{88}	0.853 ± 0.036	0.853 ± 0.036
SVM	RFE_{88}	0.857 ± 0.024	0.905 ± 0.030
CNN	STFT	0.909 ± 0.031	0.927 ± 0.031
SVM	STFT	0.757 ± 0.021	0.821 ± 0.027
CNN	**Wavelet**	**0.925 ± 0.021**	**0.947 ± 0.023**
SVM	Wavelet	0.896 ± 0.023	0.939 ± 0.022

precision-recall curve areas, explaining the large PR area deficit attributed to human labelling.

Secondly, the CNN provides a consistent performance boost with every feature combination, even for the features specifically handcrafted for the use with SVMs (RFE_{88}).

Finally, we note that the wavelet pipeline strongly outperforms the STFT, with both the CNN and SVM.

5.2 Bird classification

We now make three observations from Fig. 4 and Table 4, representing a scenario that is novel to the classifier pipelines.

Firstly, the wavelet features provide the best performance with both the CNN and the SVM, with the top result achieved by the CNN wavelet pipeline. As with the prior application, the wavelet significantly outperforms the STFT with all classifiers, with the difference magnified in this application.

Secondly, the downside to the elaborate hand-tuned feature selection scheme (RFE_{88}) quickly becomes evident when comparing performance conditioned on these features (with F_1 scores of approximately 0.85) to the results of either general deep learning configuration (with F_1 scores of 0.91 and 0.93 for the STFT and wavelet, respectively). We find results are consistent with claims made about the unsustainable nature of handcrafted feature and classifier design [28].

Finally, the CNN performs significantly better with high-dimensional, generalizable, features (STFT and wavelet) in this more difficult problem.

6 Visualizing discriminative power

In the absence of data labels, visualizations can be key to understanding how neural networks obtain their discriminative power. To ensure that the characteristics of the signal have been learnt successfully, we compute the frequency spectra $\mathbf{x}_{i,\text{test}}(f)$ of samples that maximally activate the network's units. We compare this to the training spectra $\mathbf{x}_{i,\text{train}}(f)$ using Algorithm 2.

Figures 5 and 6 show that the test samples closely resembling the training set cause the highest activations—a property we expect from our algorithms to verify they have successfully been trained. Furthermore, Fig. 5 shows that our prior expectation for the mosquito class matches the spectral content that triggers the most confident predictions. This is in the form of a distinct frequency peak around 660 Hz and its harmonic at 1325 Hz, which differs significantly from the noise class. Similarly, Fig. 6 shows unique spectral regions dedicated to each species, also with significant deviation from the noise class.

As we chose a wavelet basis with a scale directly proportional to a centre frequency, we can directly compare spectral representations with the STFT. The wavelet representation results in the more easily distinguishable peaks in the mosquito class (Fig. 5), and overall smoother spectral representations of the bird calls (Fig. 6). We note that a mismatch between high-scoring test and labelled spectra (or matches in non-information bearing regions of the spectrum) may suggest the network could be learning to detect the noise profile of the microphones used for data collection rather than the sound emitted by the object of interest.

Algorithm 2 Spectra calculation

```
for feature in {STFT, wavelet} do
    for class in $C_i$ do
        Collect highest $N$ predictions, $\hat{y}_i$
        Collect corresponding inputs, $\mathbf{X}_{i,\text{test}} \in \mathbb{R}^{N \times h_1 \times w_1}$,
            ▷ forming a concatenation of 2D images with dimensions $h_1 \times w_1$.
        Collect $N_s$ training samples to form $\mathbf{X}_{i,\text{train}}$
        Take ensemble average across patches and individual columns:
            $\mathbf{x}_{i,\text{test}}(f) = \frac{1}{w_1}\frac{1}{N}\sum_{j=1}^{w_1}\sum_{k=1}^{N} X_{ijk,\text{test}},\ \text{where } X_{ijk} \in \mathbb{R}^{h_1},$   (5)
            $\mathbf{x}_{i,\text{train}}(f) = \frac{1}{w_1}\frac{1}{N_s}\sum_{j=1}^{w_1}\sum_{k=1}^{N_s} X_{ijk,\text{train}}.$   (6)
        Normalise by mean and standard deviation
    end for
end for
```

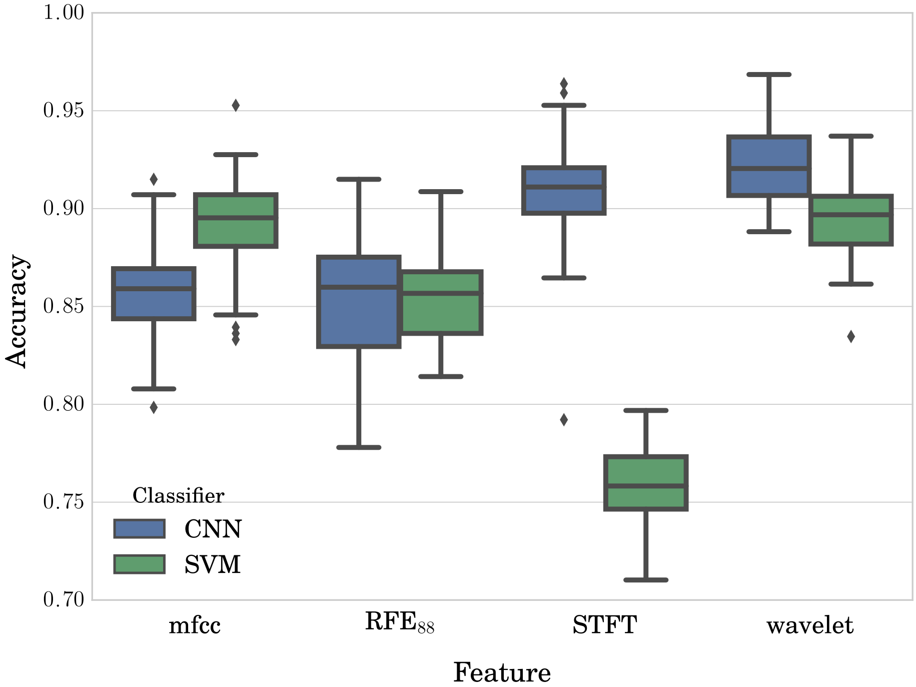


Fig. 4 **BirdCLEF** subset: boxplots of mean accuracy per class (F_1 score) for $n = 30$ trials of the CNN and SVM methods, grouped by feature combination

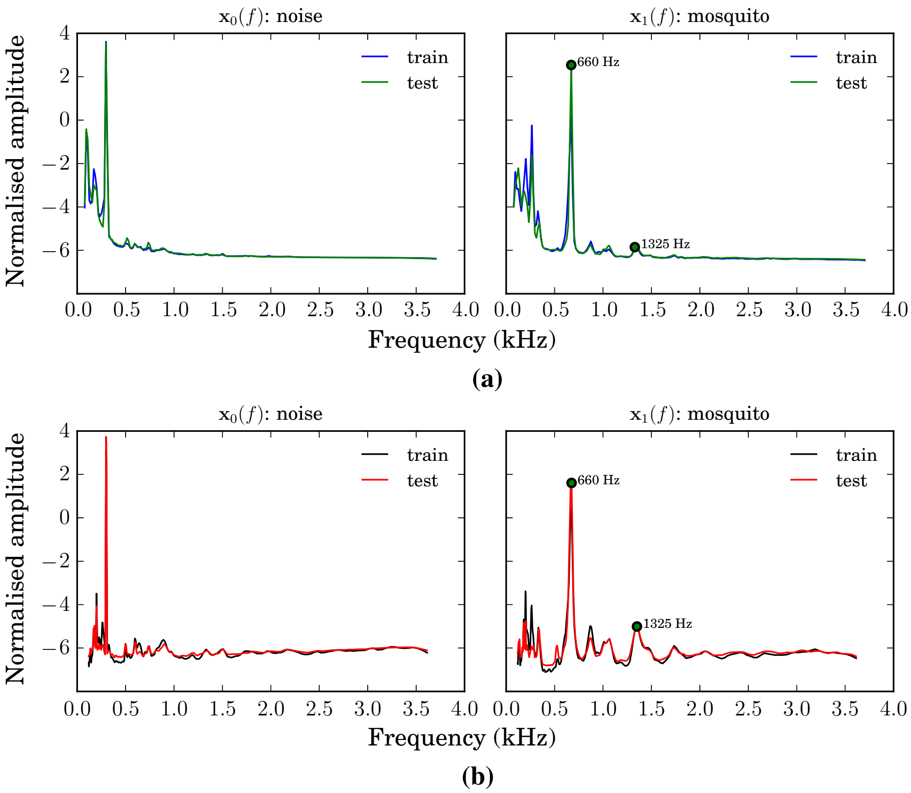
classifiers such as support vector machines commonly used in the field.

Moreover, we highlight the importance of the generality of deep learning approaches by evaluating classification performance over a 10 class subset of bird species recordings, where the wavelet-trained CNN outperforms traditional classification algorithms with no hyperparameter re-tuning of either approach. The consistent improvement observed with wavelet features over the short-time Fourier transform serves to warrant further research on whether the STFT is the correct choice to use as a base transform, as is overwhelmingly used in the literature.

7 Conclusions

This paper presents a novel approach for acoustic classification in a real-world, data-scarce scenario. We are able to more accurately and reliably differentiate between the presence and absence of a mosquito than human labellers. Furthermore, we show that a CNN outperforms generic

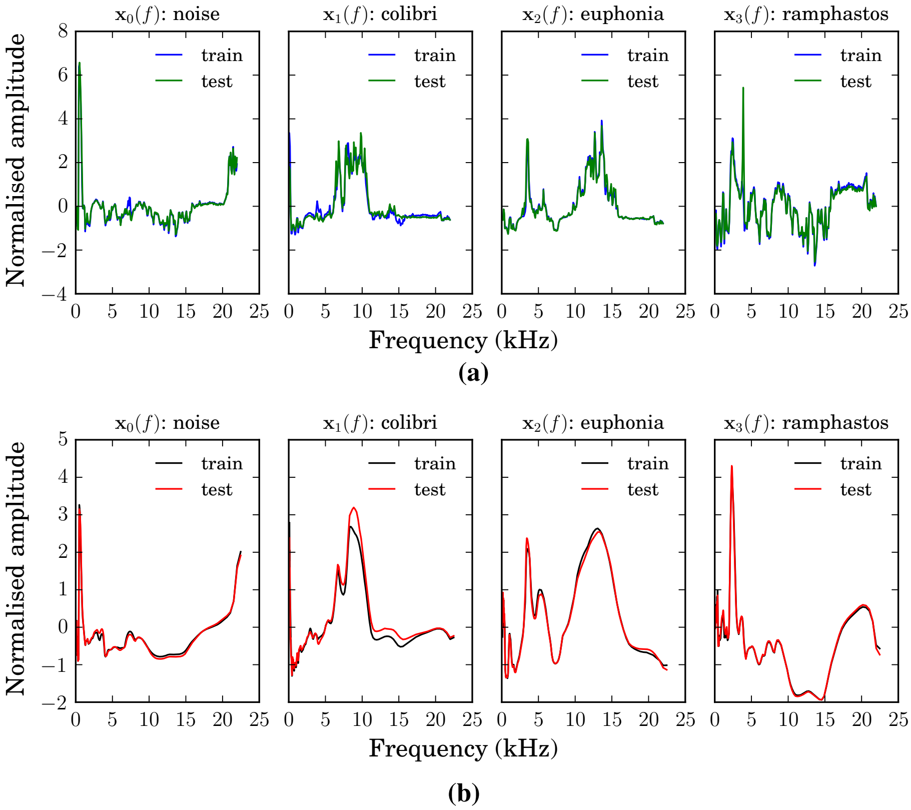
Fig. 5 **Culex mosquito** dataset: plot of normalized feature coefficient against STFT frequency bin (**a**), and wavelet centre frequency (**b**), for the 10% most confident predicted outputs over a test dataset. The learned spectra $\mathbf{x}_{i,\text{test}}(f)$ for the highest N scores closely match the labelled class spectra $\mathbf{x}_{i,\text{train}}(f)$



(a)

(b)

Fig. 6 **BirdCLEF** subset: plot of normalized feature coefficient against STFT frequency bin (**a**) and wavelet centre frequency (**b**), for the 10% most confident predicted outputs over a test dataset. The learned spectra $\mathbf{x}_{i,\text{test}}(f)$ closely match the labelled class spectra $\mathbf{x}_{i,\text{train}}(f)$



(a)

(b)

Finally, our generic feature transform allows us to visualize the learned class representation by back-propagating predictions made by the network. We thus verify that the network correctly infers the frequency characteristics of the signal, rather than a peculiarity of the recording such as the microphone noise profile. As more data becomes available, future work will aim to deploy our algorithm in a physical device to allow for large-scale bioacoustic classification.

Acknowledgements This work is part-funded by a Google (Grant No. UK Impact Challenge 2014). Ivan Kiskin is sponsored by the AIMS CDT (aims.robots.ox.ac.uk). This work is part of the HumBug project (humbug.ac.uk), a collaborative project between the University of Oxford and Kew Gardens. SR would like to thank the Royal Academy of Engineering for its support and NVIDIA for computing support. IK would like to thank Adam Cobb (acobb@robots.ox.ac.uk) and Richard Everett (richard@robots.ox.ac.uk) of the Oxford Machine Learning Research Group for their helpful suggestions, criticisms and support throughout. Funding was provided by Engineering and Physical Sciences Research Council (Grant No. EP/L015897/1).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ai OC, Hariharan M, Yaacob S, Chee LS (2012) Classification of speech dysfluencies with MFCC and LPCC features. Expert Syst Appl 39(2):2157–2165
2. Akay M (1998) Time frequency and wavelets in biomedical signal processing. IEEE press series in Biomedical Engineering
3. Alexandridis AK, Zapranis AD (2013) Wavelet neural networks: a practical guide. Neural Netw 42:1–27
4. Alphey L, Benedict M, Bellini R, Clark GG, Dame DA, Service MW, Dobson SL (2010) Sterile-insect methods for control of mosquito-borne diseases: an analysis. Vector Borne Zoonotic Dis 10(3):295–311
5. Audacity (2018) Audacity(R): free audio editor and recorder (computer application), version 2.2.2. https://audacityteam.org/. Accessed 03 May 2018
6. Bansal A, Kumar A (2015) Heisenberg uncertainty inequality for Gabor transform. arXiv preprint arXiv:150700446
7. Bhatt S, Weiss DJ, Cameron E, Bisanzio D, Mappin B, Dalrymple U, Battle KE, Moyes CL, Henry A, Eckhoff PA, Wenger EA, Briet O, Penny MA, Smith TA, Bennett A, Yukich J, Eisele TP, Griffin JT, Fergus CA, Lynch M, Lindgren F, Cohen JM, Murray CLJ, Smith DL, Hay SI, Cibulskis RE, Gething PW (2015) The effect of malaria control on *Plasmodium falciparum* in Africa between 2000 and 2015. Nature 526(7572):207–211
8. Chen Y, Yang B, Dong J (2006) Time-series prediction using a local linear wavelet neural network. Neurocomputing 69(4–6):449–465
9. Chen Y, Why A, Batista G, Mafra-Neto A, Keogh E (2014) Flying insect classification with inexpensive sensors. J Insect Behav 27(5):657–677
10. Chesmore E, Ohya E (2004) Automated identification of field-recorded songs of four British grasshoppers using bioacoustic signal recognition. Bull Entomol Res 94(04):319–330
11. Chollet F, et al (2015) Keras. https://keras.io. Accessed 07 June 2018
12. Christopoulos C, Skodras A, Ebrahimi T (2000) The JPEG2000 still image coding system: an overview. IEEE Trans Consum Electron 46(4):1103–1127
13. Claesen M, De Smet F, Suykens JA, De Moor B (2014) Fast prediction with SVM models containing RBF kernels. arXiv preprint arXiv:14030736
14. Clements AN (1999) The biology of mosquitoes, vol 2. CABI Publishing, Wallingford
15. Clemins PJ, Johnson MT, Leong KM, Savage A (2005) Automatic classification and speaker identification of African elephant (*Loxodonta africana*) vocalizations. J Acoust Soc Am 117(2):956–963
16. Cody MA (1992) The fast wavelet transform: beyond fourier transforms. Dr Dobb's J 17(4):16–28
17. Daubechies I (1990) The wavelet transform, time-frequency localization and signal analysis. IEEE Trans Inf Theory 36(5):961–1005
18. Daubechies I, Lu J, Wu HT (2011) Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool. Appl Comput Harmon Anal 30(2):243–261
19. Dieleman S, Schrauwen B (2014) End-to-end learning for music audio. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6964–6968
20. Donoho DL (1995) De-noising by soft-thresholding. IEEE Trans Inf Theory 41(3):613–627
21. Du P, Kibbe WA, Lin SM (2006) Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. Bioinformatics 22(17):2059–2065. https://doi.org/10.1093/bioinformatics/btl355
22. Giannakopoulos T (2015) pyAudioAnalysis: an open-source Python library for audio signal analysis. PloS One 10(12):e0144610
23. Goëau H, Glotin H, Vellinga WP, Planqué R, Rauber A, Joly A (2015) LifeCLEF bird identification task 2015. In: CLEF2015
24. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press. http://www.deeplearningbook.org
25. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1):389–422
26. Gwardys G, Grzywczak D (2014) Deep image features in music information retrieval. Int J Electron Telecommun 60(4):321–326
27. Hsieh TJ, Hsiao HF, Yeh WC (2011) Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm. Appl Soft Comput 11(2):2510–2525
28. Humphrey EJ, Bello JP, LeCun Y (2013) Feature learning and deep architectures: new directions for music informatics. J Intell Inf Syst 41(3):461–481
29. Joly A, Goëau H, Glotin H, Spampinato C, Bonnet P, Vellinga WP, Champ J, Planqué R, Palazzo S, Müller H (2016) LifeCLEF 2016: multimedia life species identification challenges. In: International conference of the cross-language evaluation forum for European languages. Springer, pp 286–310
30. Kadambe S, Srinivasan P (2006) Adaptive wavelets for signal classification and compression. AEU Int J Electron Commun 60(1):45–55

31. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
32. Lee H, Pham P, Largman Y, Ng AY (2009) Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Advances in neural information processing systems, pp 1096–1104
33. Lengeler C (2004) Insecticide-treated nets for malaria control: real gains. Bull World Health Organ 82(2):84–84
34. Leqing Z, Zhen Z (2010) Insect sound recognition based on SBC and HMM. In: IEEE international conference on intelligent computation technology and automation (ICICTA), vol 2, pp 544–548
35. Li D, Sethi IK, Dimitrova N, McGee T (2001) Classification of general audio data for content-based retrieval. Pattern Recognit Lett 22(5):533–544
36. Moore A, Miller JR, Tabashnik BE, Gage SH (1986) Automated identification of flying insects by analysis of wingbeat frequencies. J Econ Entomol 79(6):1703–1706
37. Mukundarajan H, Hol FJH, Castillo EA, Newby C, Prakash M (2017) Using mobile phones as acoustic sensors for high-throughput surveillance of mosquito ecology. bioRxiv https://doi.org/10.1101/120519
38. OShaughnessy D (2008) Automatic speech recognition: history, methods and challenges. Pattern Recognit 41(10):2965–2979
39. Oswald JN, Barlow J, Norris TF (2003) Acoustic identification of nine delphinid species in the eastern tropical Pacific Ocean. Mar Mamm Sci 19(1):20–037. https://doi.org/10.1111/j.1748-7692.2003.tb01090.x
40. Parsons S, Jones G (2000) Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. J Exp Biol 203(17):2641–2656
41. Pinhas J, Soroker V, Hetzoni A, Mizrach A, Teicher M, Goldberger J (2008) Automatic acoustic detection of the red palm weevil. Comput Electron Agric 63:131–139
42. Potamitis I (2014) Classifying insects on the fly. Ecol Inform 21:40–49
43. Potamitis I (2016) Deep learning for detection of bird vocalisations. arXiv preprint arXiv:160908408
44. Potamitis I, Ganchev T, Fakotakis N (2007) Automatic acoustic identification of crickets and cicadas. In: Nineth international symposium on signal processing and its applications. ISSPA 2007, pp 1–4. http://ieeexplore.ieee.org/document/4555462/
45. Sainath TN, Kingsbury B, Saon G, Soltau H, Ar Mohamed, Dahl G, Ramabhadran B (2015) Deep convolutional neural networks for large-scale speech tasks. Neural Netw 64:39–48
46. Sevilla A, Bessonne L, Glotin H (2017) Audio bird classification with inception-v4 extended with time and time-frequency attention mechanisms. Working notes of CLEF 2017
47. Silva DF, De Souza VM, Batista GE, Keogh E, Ellis DP (2013) Applying machine learning and audio analysis techniques to insect recognition in intelligent traps. In: IEEE 12th international conference on machine learning and applications (ICMLA), vol 1, pp 99–104
48. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
49. Valens C (1999) A really friendy guide to wavelets. Technical Report. Department of Computer Science, The University of New Mexico
50. Vialatte FB, Solé-Casals J, Dauwels J, Maurice M, Cichocki A (2009) Bump time-frequency toolbox: a toolbox for time-frequency oscillatory bursts extraction in electrophysiological signals. BMC Neurosci 10(1):46
51. World Health Organization, et al (2014) World Health Organization fact sheet 387, vector-borne diseases. http://www.who.int/kobe_centre/mediacentre/vbdfactsheet.pdf. Accessed 21 Apr 2017
52. World Health Organization et al (2016) World malaria report 2016. Geneva: WHO Embargoed until 13 December 2016
53. Zilli D, Parson O, Merrett GV, Rogers A (2014) A hidden Markov model-based acoustic cicada detector for crowdsourced smartphone biodiversity monitoring. J Artif Intell Res 51:805–827