Mosquito Detection with Neural Networks: The Buzz of Deep Learning

Ivan Kiskin[1,2], Bernardo Pérez Orozco[1,2], Theo Windebank[1,3], Davide Zilli[1,2], Marianne Sinka[4,5], Kathy Willis[4,5,6], and Stephen Roberts[1,2]

[1] University of Oxford, Department of Engineering, Oxford OX1 3PJ, UK,
[2] {ikiskin, ber, dzilli, sjrob}@robots.ox.ac.uk,
[3] theo.windebank@stcatz.ox.ac.uk,
[4] University of Oxford, Department of Zoology, Oxford OX2 6GG, UK,
[5] {marianne.sinka, kathy.willis}@zoo.ox.ac.uk
[6] Royal Botanic Gardens, Kew, Richmond, Surrey, TW9 3AE, UK.

Abstract. Many real-world time-series analysis problems are characterised by scarce data. Solutions typically rely on hand-crafted features extracted from the time or frequency domain allied with classification or regression engines which condition on this (often low-dimensional) feature vector. The huge advances enjoyed by many application domains in recent years have been fuelled by the use of deep learning architectures trained on large data sets. This paper presents an application of deep learning for acoustic event detection in a challenging, data-scarce, real-world problem. Our candidate challenge is to accurately detect the presence of a mosquito from its acoustic signature. We develop convolutional neural networks (CNNs) operating on wavelet transformations of audio recordings. Furthermore, we interrogate the network's predictive power by visualising statistics of network-excitatory samples. These visualisations offer a deep insight into the relative informativeness of components in the detection problem. We include comparisons with conventional classifiers, conditioned on both hand-tuned and generic features, to stress the strength of automatic deep feature learning. Detection is achieved with performance metrics significantly surpassing those of existing algorithmic methods, as well as marginally exceeding those attained by individual human experts. The data and software related to this paper are available at `http://humbug.ac.uk/kiskin2017/`.

Keywords: Convolutional neural networks, Spectrograms, Short-time Fourier transform, Wavelets, Acoustic Signal Processing

1 Introduction

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 [35,34]. 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 [19,4] and sterile insect techniques [3]. 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. Our work uses the acoustic signature of mosquito flight as the trigger for 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 recognising this sound is an effective method to locate the presence of mosquitoes and even offers the potential to categorise 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 the likely fundamental frequency and harmonics, and extensive hand-crafting of features – often similar to traditional speech representation. With impressive performance gains achieved by a paradigm shift to deep learning in many application fields, including bioacoustics [16], an opportunity emerges to leverage these advances to tackle this problem.

Deep learning approaches, however, tend to be effective only once a critical number of training samples has been reached [6]. 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 [23] and hardly ever labelled.

This paper presents a novel approach for classifying mosquito presence using scarce training data. Our approach is based on a convolutional neural network classifier conditioned on wavelet representations of the raw data. The network architecture and associated hyperparameters are strongly influenced by constraints in dataset size. To assess our performance, we compare our methods with well-established classifiers, as well as with simple artificial neural networks, trained on both hand-crafted features and the short-time Fourier transform. We show that our classifications are made more accurately and confidently, resulting in

a precision-recall curve area of 0.909, compared to 0.831 and 0.875 for the highest scoring traditional classifier and dense-layer neural network respectively. This performance is achieved on a classification task where only 70 % of labels are in full agreement amongst four domain experts. We achieve results matching, and even surpassing, human expert level accuracy. The performance of our approach allows realistic field deployments to be made as a smartphone app or on bespoke embedded systems.

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. Section 4 describes the experimental setup, in particular emphasising data-driven architectural design decisions. Section 5 highlights the value of the method. We visualise and interpret the predictions made by our algorithm on unseen data in Section 5.1 to help reveal informative features learned from the representations and verify the method. Finally, we suggest further work and conclude in Section 6.

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 [25]. Both manual and algorithmic techniques have subsequently been used to identify insects [7,36], elephants [8], delphinids [24], and other animals. The benefits of leveraging the sound animals produce – both actively as communication mechanisms and passively as a results 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 review the use of machine learning approaches in bioacoustics, in particular with respect to insect recognition. 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 Insect Detection

Real-time mosquito detection provides a method to combat the transmission of lethal diseases, mainly malaria, yellow fever and dengue fever. Unlike *Orthoptera* (crickets and grasshoppers) and *Hempitera* (e.g. cicadas), which produce strong locating and mating calls, mosquitoes (*Diptera, Culicidae*) are much quieter. The noise they emit is produced by their wingbeat, and is affected by a range of different variables, mainly species, gender, age, temperature and humidity. In the wild, wingbeat sounds are often overwhelmed by ambient noise. For these reasons, laboratory recordings of mosquitoes are regularly taken on tethered mosquitoes in quiet or even soundproof chambers, and therefore do not represent realistic conditions.

Even in this data-scarce scenario, the employment of artificial neural networks has been proven successful for a number of years. In [7] 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 [29,26] and hidden Markov models [20,36], applied to a variety of different features extracted from recordings of singing insects.

Chen et al. [6] attribute the stagnation of automated insect detection accuracy to the mere use of acoustic devices, which are allegedly not capable of producing a signal sufficiently clean to be classified correctly. As a consequence, they replace microphones with pseudo-acoustic optical sensors, recording mosquito wingbeat through a laser beam hitting a phototransistor array – a practice already proposed by Moore et al. [22]. This technique however relies on the ability to lure a mosquito through the laser beam.

Independently of the technique used to record a mosquito wingbeat frequency, the need arises to be able to identify the insect's flight in a noisy recording. The following section reviews recent achievements in the wider context of 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 denoising 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 [15], leading to hand-crafted 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, Ch. 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 non-linear 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 [27], but more frequently complex features and feature combinations are used, applying dimensionality reduction to combat the curse of dimensionality [18]. 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 [21].

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 features 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 [17] and speech recognition [18]. 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 time-series signal processing.

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 [11], which was improved to 0.69 the following year when deep learning was introduced, outperforming the closest scoring hand-crafted method that scored 0.58 [16]. The impressive performance gain came from the utilisation 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. Alongside this example, the most widely used base method to transform the input signals is the STFT [30,14,28].

However, to the best of our knowledge, the more flexible wavelet transform is hardly ever used as the representation domain for a convolutional neural network. As a result, in the following section we present our methodology, 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 Method

We present a novel wavelet-transform-based convolutional neural network architecture for the detection of mosquitoes' flying tone in a noisy audio recording. 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

As an initial step, we extract the training data into a format suitable for the classifier. We choose to use the continuous wavelet transform (CWT) due to its successful application in time-frequency analysis [9] (Step 2 of Algorithm 1). Given the direct relationship between the wavelet scale and centre frequency, we use the bump wavelet [33], expressed in the Fourier domain as:

$$\Psi(s\omega) = e^{1-\sigma^2/(1-(s\omega-\mu)^2)}\mathbb{I}[(\mu-\sigma)/s,(\mu+\sigma)/s], \quad (1)$$

where $\mathbb{I}[\cdot]$ is the indicator function and s is the wavelet scale. High values of μ, as well as small values of σ, result in a wavelet with superior frequency localisation but poorer time localisation.

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 $\mathbf{y}_{\text{train}}$:

$$\mathbf{X}_{\text{train}} \in \mathbb{R}^{N_S \times h_1 \times w_1}, \mathbf{y}_{\text{train}} \in \mathbb{R}^{N_S \times 2},$$

where N_s is the number of training samples formed by splitting the transformed recordings into 2D 'images' with dimensions $h_1 \times w_1$.
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 $y_{i,\text{pred}}$ for each class C_i: $\{C_0 = \text{non-mosquito}, C_1 = \text{mosquito}\}$, where

$$0 \leq y_{i,\text{pred}}(\mathbf{x}) \leq 1, \quad \text{such that} \quad \sum_{i=1}^{n} y_{i,\text{pred}}(\mathbf{x}) = 1.$$

3.2 Neural Network Configurations

A convolutional layer $H_{\text{conv}} : \mathbb{R}^{h_1 \times w_1 \times c} \to \mathbb{R}^{h_2 \times w_2 \times N_k}$ 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 [12, Ch. 9]:

$$\mathbf{Y}_k(i,j) = \mathbf{X} * \mathbf{W}_p = \sum_{i'} \sum_{j'} \mathbf{X}(i-i', j-j')\mathbf{W}_p(i', j'). \quad (2)$$

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).

A fully connected layer $H_{\text{FC}} : \mathbb{R}^m \to \mathbb{R}^n$ with input $\mathbf{x} \in \mathbb{R}^m$ and output $\mathbf{y} \in \mathbb{R}^n$ is given by $\mathbf{y} = H_{\text{FC}}(\mathbf{x}) = \phi(\mathbf{W}\mathbf{x} + \mathbf{b})$, where $\{\mathbf{W}, \mathbf{b}\}$ are the learnable parameters of the network and ϕ is the activation function of layer, often chosen to be non-linear.

The data size constraint results in an architecture choice (Figure 1) of few layers and free parameters. To prevent overfitting, our network comprises an input layer connected sequentially to a single convolutional layer and a fully connected layer, which is connected to the two output classes with

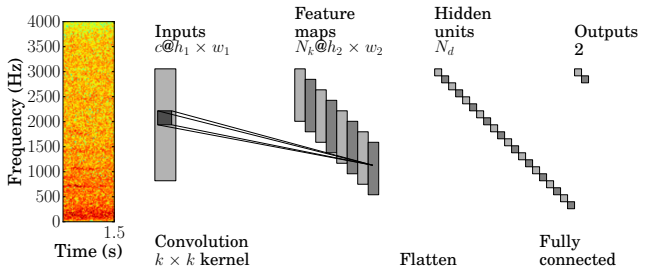


Fig. 1: The CNN pipeline. 1.5 s wavelet spectrogram of mosquito recording is partitioned into images with $c = 1$ channels, of dimensions $h_1 \times w_1$. This serves as input to a convolutional network with N_k filters with kernel $\mathbf{W}_p \in \mathbb{R}^{k \times k}$. Feature maps are formed 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.

dropout [32] with $p = 0.5$. Rectified Linear Units (ReLU) activations are employed based on their desirable training convergence properties [17]. Finally, potential candidate hyperparameters are cross-validated to determine an appropriate model, as detailed in Section 4.2.

Using conventional multi-layer perceptrons (MLPs) one may simply collapse the matrix $\mathbf{X}$ into a single column vector $\mathbf{x}$. Unlike their convolutional counterparts, MLPs are not explicitly asked to seek relationships among adjacent neurons. Whereas this may provide the model with more flexibility to find relationships between seemingly distant nodes, convolutional layers formally make the model acknowledge that units are correlated in space. Without this assumption, MLPs will look for sets of weights in a space in which this constraint has not been made explicit. Our MLP architecture, chosen for comparison with the CNN, is illustrated in Figure 2. The network omits the convolutional layer, taking the form of an input layer followed by two fully connected layers, with dropout with $p = 0.5$ on the connections to the output nodes.

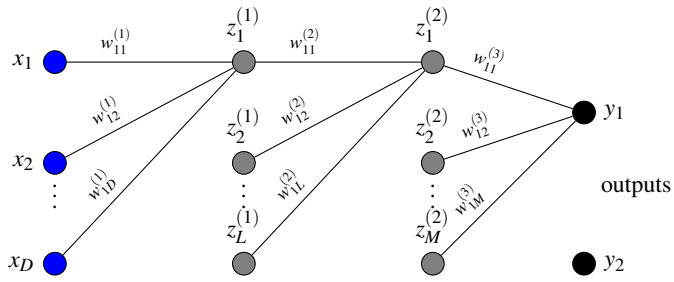


Fig. 2: MLP architecture. For clarity the diagram displays connections for a few units. Each layer is fully connected with ReLU activations. Input dimensions $D = h_1 \times w_1$. Number of hidden units in the first and second layers labelled L and M respectively.

3.3 Traditional Classifier Baseline

As a baseline, we compare the neural network models with more traditional classifiers that require explicit feature design. We choose three candidate classifiers widely used in machine learning with audio: random forests (RFs), naive 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 [31], especially in data-scarce problems.

We have selected ten features: mel-frequency cepstrum slices, STFT spectrogram slices, mel-frequency cepstrum coefficients, entropy, energy entropy, spectral entropy, flux, rolloff, 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 [10]). To select features optimally, we have applied both recursive feature elimination (RFE) and principal component analysis (PCA), and also cross-validated each feature individually. By reducing redundant descriptors we can improve classification performance in terms of both speed and predictive ability, confirmed by the cross-validation results in Section 4.2.

4 Experimental Details

4.1 Data Annotation

The data used here were recorded in January 2016 within culture cages containing both male and female *Culex quinquefasciatus* [5]. The females were not blood-fed and both sexes were maintained on a diet of 10 % w/v sucrose solution. 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.

The signal is sampled at $F_s = 8\,\text{kHz}$, which limits the highest theoretically resolvable frequency to 4 kHz due to the Nyquist limit. Figure 3 (lower) shows the classifications within $y_i = \{0, 1\}$: absence, presence of mosquito, as labelled by four individual human experts. Of these, one particularly accurate label set is taken as a gold-standard reference to both train the algorithms and benchmark with the remaining experts. The resulting label rate is given as $F_l = 10\,\text{Hz}$. The labels are up-sampled to match the spectral feature frequency, F_{spec}, which is calculated as $F_{\text{spec}} = F_s/h_1$, provided the overlap between windowed Fourier transforms in samples is half the number of Fourier coefficients.

4.2 Parameter Cross-Validation

In this section we report the design and parameter considerations that we used cross-validation to estimate. The available 57 recordings were split into 37 training and 20 test signals, creating approximately 6,000 to 60,000 training samples, for

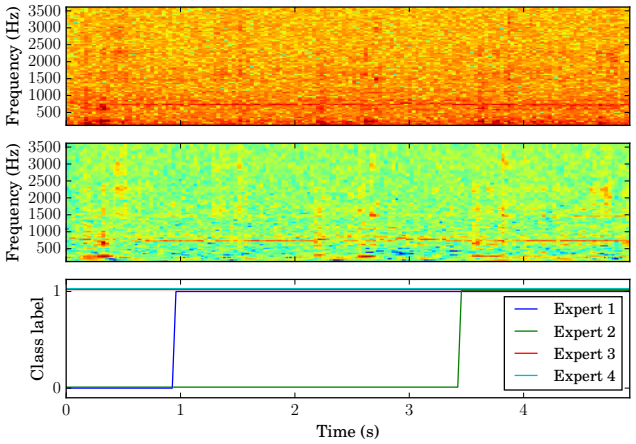


Fig. 3: STFT (top) and wavelet (middle) representations of signal with $h_1 = 256$ frequency bins and wavelet scales respectively. Corresponding varying class labels (bottom) as supplied by human experts. The wavelet representation shows greater contrast in horizontal constant frequency bands that correspond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

window widths $w_1 = 10$ and $w_1 = 1$ samples, respectively. Both neural networks were trained with a batch size of 256 for 20 epochs, according to validation accuracy in conjunction with early stopping criteria.

We start with the CNN and note that the characteristic length scale of the signal determines the choice of slice width. For musical extracts, or bird songs, it is crucial to capture temporal structure. This favours taking longer sections, allowing an appropriate convolutional receptive field in the time domain (along the x-axis). A mosquito tone is relatively consistent in frequency over time, so shorter slices are likely to provide a larger training set without loss of information per section. We thus restrict ourselves to dividing the training data into 320 ms fixed width samples ($w_1 = 10$). When choosing the filter widths to trial, we note that spectrogram samples are correlated in local regions and will contain harmonics that are non-local. The locality is confined to narrow frequency bands, as well as through time (along the y and x-axes respectively). Taking this into account, we arrive at the cross-validation grid and results of Table 1.

For the MLP, we choose to cross-validate the narrowest training sample width $w_1 = 1$, and the CNN architecture sample width $w_1 = 10$ forming a column vector $\mathbf{x}_{\text{train}} \in \mathbb{R}^{h_1 w_1 \times 1}$ for each training sample. We then estimate the optimal number of hidden units as given in Table 1.

The traditional classifiers are cross-validated with PCA and RFE dimension reduction as given by n, m in Table 1. The best performing feature set for all traditional classifiers is the set extracted by cross-validated recursive feature elimination as in [13], outperforming all PCA reductions for every classifier-feature pair. The result is a feature set that we denote as RFE_{88} which retains 88 dimensions from the ten original features which spanned 304 dimensions ($F_{10} \in \mathbb{R}^{304}$).

Table 1: Cross-validation results. Optimal hyperparameters given in bold.

Classifier	Features	Cross-validation grid
CNN	STFT	$k \in \{2, \mathbf{3}, 4, 5\}$, $N_k \in \{8, 16, \mathbf{32}\}$, $N_d \in \{16, 64, \mathbf{128}, 256\}$.
CNN	Wavelet	$k \in \{2, 3, 4, \mathbf{5}\}$, $N_k \in \{8, 16, \mathbf{32}\}$, $N_d \in \{16, 64, \mathbf{128}, 256\}$.
MLP	STFT	$w_1 \in \{1, \mathbf{10}\}$, $L \in \{8, 256, 1028, \mathbf{2056}\}$, $M \in \{\mathbf{64}, 512, 1024\}$.
MLP	Wavelet	$w_1 \in \{1, \mathbf{10}\}$, $L \in \{8, \mathbf{256}, 1028, 2056\}$, $M \in \{64, 512, \mathbf{1024}\}$.
NB, RF, SVM	$F_{10} \in \mathbb{R}^{304}$	$\text{PCA} \in \mathbb{R}^N, N \in 0.8^n \times 304$, $n \in \{0, 1, \ldots, 12\}$, $\text{RFE} \in \mathbb{R}^M, M \in 304 - 8m$, $m \in \{0, 1, \ldots, \mathbf{27}, \ldots 35\}$.

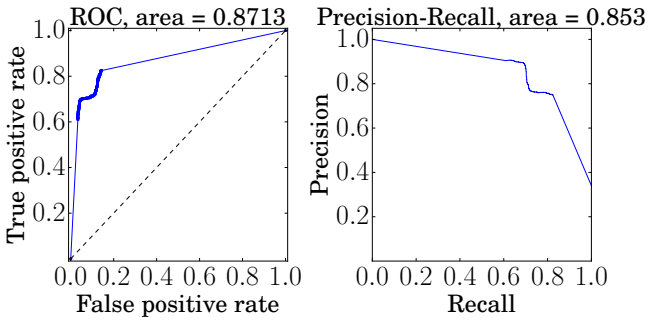
5 Classification Performance

The performance metrics are defined at the resolution of the extracted features and presented in Table 2. We emphasise that the ultimate goal is deployment in fieldwork on smartphones or embedded devices. The device will be in constant *listening* mode, and mainly consume power during the data *write* mode that is initiated by signal detections. A high true negative rate (TNR) is very desirable for this application, as preventing false positive detections leads to critical conservation of battery power. Taking this into account, we highlight four key results.

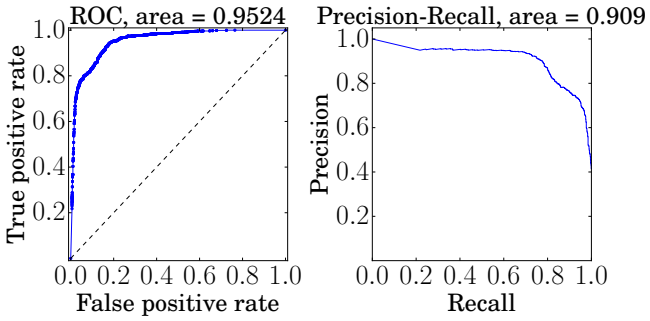
Firstly, training the neural networks on wavelet features shows a consistent relative improvement compared to training on STFT features. We attribute the improved receiver operating characteristic curve (ROC) area to the network producing better estimates of the uncertainty of each prediction. As a result, a greater range of the detector output $0 \leq y_i \leq 1$ is utilised. This is best represented by the contrast in smoothness of the ROC curves, as well as the spread of predictions visible for the classifier test output in Figure 5.

Secondly, the addition of the convolutional layer provides a significant increase in every performance metric compared to the MLPs. Therefore, omitting the specific locality constraint of the CNN degrades performance.

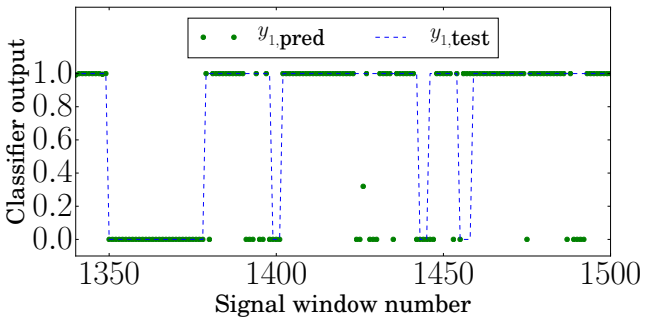
Thirdly, the CNN trained on wavelet features is able to perform classifications with F_1 score, precision-recall (PR) and ROC areas, far exceeding the results obtained with traditional classifiers. This is despite using an elaborate hand-tuned feature selection scheme that cross-validates both PCA and RFE to extract salient features. By also comparing the lower achieving CNN conditioned on STFT features, we note that both the feature representation and architecture add critical value to the detection process.



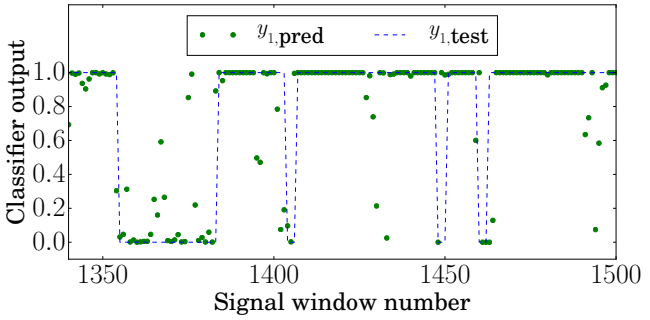
(a) Spectrogram



(b) Wavelet



(a) Spectrogram



(b) Wavelet

Fig. 5: ROC, precision-recall, and classifier outputs over test data for 5a: STFT with 256 Fourier coefficients and 5b: wavelet with 256 scales. Target prediction for a range of signal windows is given by the blue dotted line, with actual predictions denoted by green dots. Each prediction is generated over $w_1 = 10$ samples – a window of 320 ms.

Table 2: Summary classification metrics. The metrics are evaluated from a single run on test data, following 10-fold cross-validation of features and hyperparameters on training dataset.

Classifier	Features	F_1 score	TPR	TNR	ROC area	PR area
MLP	STFT	0.751	0.65	0.96	0.858	0.830
MLP	Wavelet	0.745	0.63	0.97	0.921	0.875
CNN	STFT	0.779	0.69	0.96	0.871	0.853
CNN	Wavelet	**0.817**	0.73	**0.97**	**0.952**	**0.909**
Naive Bayes	STFT	0.521	0.65	0.74	0.743	0.600
Naive Bayes	RFE_{88}	0.484	0.51	0.83	0.732	0.414
Random Forest	STFT	0.674	0.69	0.89	0.896	0.733
Random Forest	RFE_{88}	0.710	0.68	0.93	0.920	0.800
SVM	STFT	0.685	**0.83**	0.81	0.902	0.775
SVM	RFE_{88}	0.745	0.73	0.93	0.928	0.831
CNN, median filter	Wavelet	0.854	0.78	0.98	0.970	0.939
Expert 1	N/A	0.819	0.89	0.85	0.873	0.843
Expert 2	N/A	0.856	0.92	0.88	0.901	0.873
Expert 3	N/A	0.852	0.77	0.98	0.874	0.901

Finally, median filtering the CNN's predictions conditioned on the wavelet features considerably boosts performance metrics, allowing our algorithm to outperform human experts. By using a median filter kernel (of 1 second) that represents the smoothness over which human labelling approximately occurred, we are able to compare performance with human expert labelling. Since human labels were supplied as absolute (either $y_i = 1, y_i = 0$), an incorrect label incurs a large penalty on the ROC and precision-recall curve areas. This results in a far exceeding ROC area of 0.970 for the CNN-wavelet network, compared to 0.873, 0.901 and 0.874 of the three human experts respectively. However, even raw accuracies are comparable, as indicated by the near identical F_1 score of the best hand label attempt and our filtered algorithm. Further algorithmic improvements are readily attainable (e.g. classifier aggregation and temporal pooling), but fall beyond the scope of this paper.

5.1 Visualising Discriminative Power

In the absence of data labels, visualisations 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 $\mathbf{X}_i(f)$ of samples that maximally activate the network's units. We collect the highest N predictions for the mosquito class, $\hat{y}_1$, and non-mosquito class, $\hat{y}_0$, respectively. The high-scoring test data forms a tensor $\mathbf{X}_{i,\text{test}} \in \mathbb{R}^{N\times 256\times 10}, i = \{0,1\}$, which is the concatenation of N spectrogram patches. The frequency spectra are then computed by taking the ensemble average across the patches and individual columns as follows:

$$\mathbf{x}_{i,\text{test}}(f) = \frac{1}{10}\frac{1}{N}\sum_{j=1}^{10}\sum_{k=1}^{N} X_{ijk}, \text{ where } X_{ijk} \in \mathbb{R}^{256}. \quad (3)$$

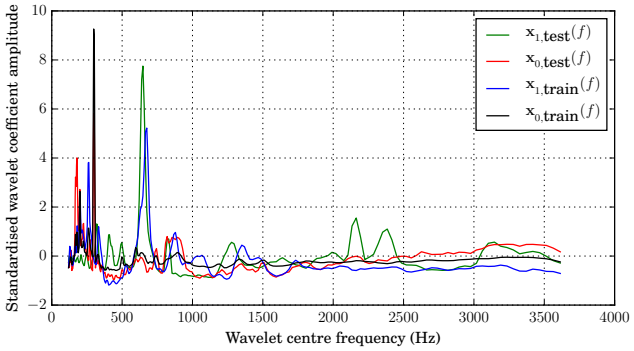


Fig. 6: Plot of standardised wavelet coefficient amplitude against centre frequency of each wavelet scale for the top 10 % predicted outputs over a test dataset. The learned spectra $\mathbf{x}_{i,\text{test}}(f)$ for the highest N scores closely match the frequency characteristics of the labelled class samples $\mathbf{x}_{i,\text{train}}(f)$.

Similarly, we compute spectra $\mathbf{x}_{i,\text{train}}(f)$ for the two classes from the N_s labelled training samples. We make our spectra zero-mean and unit-variance in order to make direct comparisons between the high-scoring test spectra for each class $\mathbf{x}_{i,\text{test}}(f)$, and their reference from the training set $\mathbf{x}_{i,\text{train}}(f)$. The resulting test spectrum for the mosquito class ($\mathbf{x}_1(f)$, Figure 6) shows a distinct frequency peak around 650 Hz. This peak clearly matches the audible frequency of the mosquito, confirming that the network is making predictions based on learnt features of the true signal. The same holds true for the noise spectra ($\mathbf{x}_0(f)$), which is dominated by a component around 300 Hz. A mismatch between learnt and labelled spectra would raise warning flags to the user, suggesting the network may for example be learning to detect the noise profile of the microphones used for data collection rather than the mosquito flight tones.

6 Conclusions

This paper presents a novel approach for acoustic classification of free-flying mosquitoes in a real-world, data-scarce scenario. We show that a convolutional neural network outperforms generic classifiers such as random forests and support vector machines commonly used in the field. The neural network, trained on a raw wavelet spectrogram, also outperforms traditional, hand-crafted feature extraction techniques, surpassing any combination of alternative feature-algorithm pairs. Moreover, we conclude that the addition of a convolutional layer results in performance gains over non-convolutional neural networks with both Fourier and wavelet representations. With the further addition of rolling median filtering, the approach is able to improve on human expert labelling.

Furthermore, our generic feature transform allows us to visualise 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 mosquito, rather than a peculiarity of the recording such as the microphone noise profile. Future work will generalise our model to multiple classes, such as individual mosquito species, and deploy our algorithm in a physical device to allow for large-scale collection of data.

Acknowledgements. This work is part-funded by a Google Impact Challenge award. Ivan Kiskin is sponsored by the 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.

References

1. Ai, O.C., Hariharan, M., Yaacob, S., Chee, L.S.: Classification of speech dysfluencies with MFCC and LPCC features. Expert Systems with Applications 39(2), 2157–2165 (2012)
2. Akay, M.: Time Frequency and Wavelets in Biomedical Signal Processing. IEEE press series in Biomedical Engineering (1998)
3. Alphey, L., Benedict, M., Bellini, R., Clark, G.G., Dame, D.A., Service, M.W., Dobson, S.L.: Sterile-insect methods for control of mosquito-borne diseases: an analysis. Vector-Borne and Zoonotic Diseases 10(3), 295–311 (2010)
4. Bhatt, S., Weiss, D.J., Cameron, E., Bisanzio, D., Mappin, B., Dalrymple, U., Battle, K.E., Moyes, C.L., Henry, A., Eckhoff, P.A., Wenger, E.A., Briet, O., Penny, M.A., Smith, T.A., Bennett, A., Yukich, J., Eisele, T.P., Griffin, J.T., Fergus, C.A., Lynch, M., Lindgren, F., Cohen, J.M., Murray, C.L.J., Smith, D.L., Hay, S.I., Cibulskis, R.E., Gething, P.W.: The effect of malaria control on Plasmodium falciparum in Africa between 2000 and 2015. Nature 526(7572), 207–211 (10 2015)
5. Bhattacharya, S., Basu, P.: The southern house mosquito, Culex quinquefasciatus: profile of a smart vector. J Entomol Zoo Stud 4, 73–81 (2016)
6. Chen, Y., Why, A., Batista, G., Mafra-Neto, A., Keogh, E.: Flying insect classification with inexpensive sensors. Journal of insect behavior 27(5), 657–677 (2014)
7. Chesmore, E., Ohya, E.: Automated identification of field-recorded songs of four British grasshoppers using bioacoustic signal recognition. Bulletin of Entomological Research 94(04), 319–330 (2004)
8. Clemins, P.J., Johnson, M.T.: Automatic type classification and speaker identification of African elpehant vocalizations (2002)
9. Daubechies, I., Lu, J., Wu, H.T.: Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. Applied and computational harmonic analysis 30(2), 243–261 (2011)
10. Giannakopoulos, T.: pyAudioAnalysis: An open-source Python library for audio signal analysis. PloS one 10(12), e0144610 (2015)
11. Goëau, H., Glotin, H., Vellinga, W.P., Planqué, R., Rauber, A., Joly, A.: LifeCLEF bird identification task 2015. In: CLEF2015 (2015)
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), `http://www.deeplearningbook.org`
13. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine learning 46(1), 389–422 (2002)
14. Gwardys, G., Grzywczak, D.: Deep image features in music information retrieval. International Journal of Electronics and Telecommunications 60(4), 321–326 (2014)

15. Humphrey, E.J., Bello, J.P., LeCun, Y.: Feature learning and deep architectures: new directions for music informatics. Journal of Intelligent Information Systems 41(3), 461–481 (2013)
16. Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.P., Champ, J., Planqué, R., Palazzo, S., Müller, H.: LifeCLEF 2016: multimedia life species identification challenges. In: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 286–310. Springer (2016)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
18. Lee, H., Pham, P., Largman, Y., Ng, A.Y.: Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Advances in neural information processing systems. pp. 1096–1104 (2009)
19. Lengeler, C.: Insecticide-treated nets for malaria control: real gains. Bulletin of the World Health Organization 82(2), 84–84 (2004)
20. Leqing, Z., Zhen, Z.: Insect sound recognition based on sbc and hmm. In: Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on. vol. 2, pp. 544–548. IEEE (2010)
21. Li, D., Sethi, I.K., Dimitrova, N., McGee, T.: Classification of general audio data for content-based retrieval. Pattern recognition letters 22(5), 533–544 (2001)
22. Moore, A., Miller, J.R., Tabashnik, B.E., Gage, S.H.: Automated identification of flying insects by analysis of wingbeat frequencies. Journal of Economic Entomology 79(6), 1703–1706 (1986)
23. Mukundarajan, H., Hol, F.J.H., Castillo, E.A., Newby, C., Prakash, M.: Using mobile phones as acoustic sensors for high-throughput surveillance of mosquito ecology. bioRxiv (2017), http://www.biorxiv.org/content/early/2017/03/25/120519
24. Oswald, J.N., Barlow, J., Norris, T.F.: Acoustic identification of nine delphinid species in the eastern tropical pacific ocean. Marine Mammal Science 19(1), 20–037 (2003), http://dx.doi.org/10.1111/j.1748-7692.2003.tb01090.x
25. Parsons, S., Jones, G.: Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. Journal of Experimental Biology 203(17), 2641–2656 (2000)
26. Pinhas, J., Soroker, V., Hetzoni, A., Mizrach, A., Teicher, M., Goldberger, J.: Automatic acoustic detection of the red palm weevil. Computer and Electronics in Agriculture 63, 131–139 (2008), http://www.sciencedirect.com/science/article/pii/S0168169908000628
27. Potamitis, I.: Classifying insects on the fly. Ecological Informatics 21, 40–49 (2014)
28. Potamitis, I.: Deep learning for detection of bird vocalisations. arXiv preprint arXiv:1609.08408 (2016)
29. Potamitis, I., Ganchev, T., Fakotakis, N.: Automatic acoustic identification of crickets and cicadas. In: Nineth International Symposium on Signal Processing and Its Applications, 2007. ISSPA 2007. pp. 1–4 (2007), http://ieeexplore.ieee.org/document/4555462/
30. Sainath, T.N., Kingsbury, B., Saon, G., Soltau, H., Mohamed, A.r., Dahl, G., Ramabhadran, B.: Deep convolutional neural networks for large-scale speech tasks. Neural Networks 64, 39–48 (2015)
31. Silva, D.F., De Souza, V.M., Batista, G.E., Keogh, E., Ellis, D.P.: Applying machine learning and audio analysis techniques to insect recognition in intelligent traps. In: Machine Learning and Applications (ICMLA), 2013 12th International Conference on. vol. 1, pp. 99–104. IEEE (2013)
32. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1), 1929–1958 (2014)
33. Vialatte, F.B., Solé-Casals, J., Dauwels, J., Maurice, M., Cichocki, A.: Bump time-frequency toolbox: a toolbox for time-frequency oscillatory bursts extraction in electrophysiological signals. BMC neuroscience 10(1), 46 (2009)
34. World Health Organization, et al.: World Health Organization fact sheet 387, vector-borne diseases (2014), http://www.who.int/kobe_centre/mediacentre/vbdfactsheet.pdf, accessed: 2017-04-21
35. World Health Organization, et al.: World malaria report 2016. Geneva: WHO. Embargoed until 13 (2016)
36. Zilli, D., Parson, O., Merrett, G.V., Rogers, A.: A hidden Markov model-based acoustic cicada detector for crowdsourced smartphone biodiversity monitoring. Journal of Artificial Intelligence Research 51, 805–827 (2014)