Received: 22 December 2021 | Accepted: 30 March 2022

DOI: 10.1111/2041-210X.13873

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

RESEARCH ARTICLE

Automated call detection for acoustic surveys with structured calls of varying length

Yuheng Wang[1] | Juan Ye[2] | David L. Borchers[1,3]

[1]Centre for Research into Ecological and Environmental Modelling, School of Mathematics and Statistics, University of St Andrews, The Observatory, St Andrews, UK

[2]School of Computer Science, University of St Andrews, St Andrews, UK

[3]Centre for Statistics in Ecology, The Environment, and Conservation, University of Cape Town, Cape Town, South Africa

Correspondence
Yuheng Wang
Email: yw99@st-andrews.ac.uk

Funding information
China Scholarship Council, Grant/Award Number: 202008060348

Handling Editor: Aaron Ellison

Abstract

1. When recorders are used to survey acoustically conspicuous species, identification calls of the target species in recordings is essential for estimating density and abundance. We investigate how well deep neural networks identify vocalisations consisting of *phrases* of varying lengths, each containing a variable number of *syllables*. We use recordings of Hainan gibbon *Nomascus hainanus* vocalisations to develop and test the methods.
2. We propose two methods for exploiting the two-level structure of such data. The first combines convolutional neural network (CNN) models with a hidden Markov model (HMM) and the second uses a convolutional recurrent neural network (CRNN). Both models learn acoustic features of syllables via a CNN and temporal correlations of syllables into phrases either via an HMM or recurrent network. We compare their performance to commonly used CNNs LeNet and VGGNet, and support vector machine (SVM). We also propose a dynamic programming method to evaluate how well phrases are predicted. This is useful for evaluating performance when vocalisations are labelled by phrases, not syllables.
3. Our methods perform substantially better than the commonly used methods when applied to the gibbon acoustic recordings. The CRNN has an *F*-score of 90% on phrase prediction, which is 18% higher than the best of the SVM or LeNet and VGGNet methods. HMM post-processing raised the *F*-score of these last three methods to as much as 87%. The number of phrases is overestimated by CNNs and SVM, leading to error rates between 49% and 54%. With HMM, these error rates can be reduced to 0.4% at the lowest. Similarly, the error rate of CRNN's prediction is no more than 0.5%.
4. CRNNs are better at identifying phrases of varying lengths composed of a varying number of syllables than simpler CNN or SVM models. We find a CRNN model to be best at this task, with a CNN combined with an HMM performing almost as well. We recommend that these kinds of models are used for species whose vocalisations are structured into phrases of varying lengths.

KEYWORDS
acoustic survey, automated call detection, convolutional recurrent neural network, gibbon calls, hidden Markov model, machine learning

1 | INTRODUCTION

Acoustic surveys are being used increasingly for wildlife surveys (Gibb et al., 2019). Acoustic recordings are now commonplace (Priyadarshani et al., 2018; Usman et al., 2020) and it is appropriate to ask how best to identify animal calls in these recordings. Identifying calls manually is time-consuming and labour-intensive, can be subjective, and must be done by trained professionals (Chen & Maher, 2006; Somervuo et al., 2006). A variety of machine learning techniques has been used for automatic call detection or classification. However, these methods are either designed to detect simple calls of short duration (e.g. anuran calls; Alonso et al., 2017; Colonna et al., 2015) or calls of roughly fixed length (Stiffler et al., 2018; Zhang & Li, 2015). In this paper, we develop methods to tackle structured calls of the sort illustrated in Figure 1. Calls can be viewed as *phrases*, each of which contains a number of syllables. Both syllables and phrases can be of varying lengths, and there may be different numbers of syllables in each phrase. In general, the intervals between syllables are shorter than those between phrases, but this difference can be small. All these characteristics make call detection a challenging task.

We aim to use phrase detection to estimate animal abundance and density, which are key quantities required for management and conservation. To do this, one either has to identify which detected vocalisations came from which animals (in order to use the animal as the sampling unit) or to estimate vocalisation density per unit time without identifying which vocalisations come from which animals, and then separately estimate vocalisation rate in order to convert phrase density into animal density (Buckland, 2006; Buckland et al., 2001). When phrases are made up of multiple syllables of varying length, the variance in syllable production rate over any period is greater than that of phrase rate and so it is convenient to work with the phrase as the sampling unit, rather than the syllable. In addition, when vocalisations are identified to animals, this is often done by localising the vocalisation source using (primarily) estimates of direction to source from multiple detectors (Stevenson et al., 2015), and in this case, there is negligible additional information about location in individual syllables beyond that contained in the phrase as a whole, so here too it is convenient to work with the phrase as the sampling unit rather than the syllable.

Furthermore, because we are interested in abundance and density, we are interested in identifying all phrases, not just identifying whether there was at least one phrase. While presence/absence data (e.g. whether or not the species of interest was heard at least once in some time window) are useful for estimating occupancy and species range, it is a very inferior kind of data for estimating abundance and density.

However, detecting phrases in hierarchically structured vocalisations is challenging. Many existing methods perform segment-based prediction; that is, they divide audio files into segments with a small time window (e.g. 1 s) and then predict labels for each segment. The performance of these methods is sensitive to the segment length. If the segment length is set shorter than the intervals between syllables within a phrase, then there might be many segments within phrases that are labelled positive but do not contain any vocalisations. This can result in a high false negative rate for phrase prediction even if each segment is correctly predicted. On the other hand, if the segment length is set too long, then a segment may contain multiple phrases, which leads to the underestimation of phrase prediction. This problem is exacerbated when phrases may be of varying lengths. Therefore, selecting the appropriate segment length is often a difficult design decision to make.

We propose instead two methods for automatic identification of structured phrases of the sort described above. The first combines convolutional neural network (CNN) (LeCun et al., 2015) models with a hidden Markov model (HMM) and the second uses a convolutional recurrent neural network (CRNN). Both models learn acoustic features of syllables via a CNN and temporal correlations of syllables into phrases either via HMM post-processing or a recurrent network. The novelty of our proposed methods is that we adopt a sequence-to-sequence (Sutskever et al., 2014) prediction strategy to address the hierarchical vocalisation problem. That is, we input

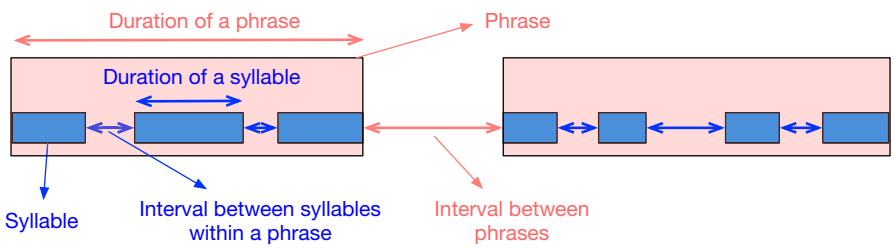


FIGURE 1 An illustrative example of phrases and syllables. A phrase consists of a number of syllables. Both syllables (in blue boxes) and phrases (in pink boxes) can be of varying lengths. In general, there exists a shorter interval between syllables and a longer interval between phrases, but their difference can be small

a sequence of segments (comprising a fixed number of consecutive segments) and output a sequence of predictions, corresponding to each segment in input. This strategy is widely used in machine translation (Wu et al., 2016) and text recognition (Shi et al., 2017). Unlike many existing methods, our methods do not require predefined time intervals for phrases, and are less sensitive to segment or sequence length, as long as the segment length is reasonably small (e.g. 1 s) and the sequence length is larger than the maximum duration of a phrase. In addition, a CRNN is trained in an end-to-end manner to optimise the learning of acoustic and temporal features altogether and HMM post-processing can be added to any pre-trained method. We also propose a new event-based evaluation method to assess the performance of methods at the phrase level.

The rest of the paper is organised as follows. Section 2 introduces the background to our work, followed by a description of the methods, the data, data pre-processing, our evaluation metrics and the computational experiments setting that we use to evaluate methods. Section 3 compares the performance of our methods to a number of the existing machine learning methods. In Section 4 we draw conclusions from the application of these methods to the Hainan gibbon data.

2 | MATERIALS AND METHODS

We propose two methods to identify phrases: a CNN with HMM post-processing, and a CRNN. To the best of our knowledge, we are the first to propose such methods for detecting phrases in recordings of calls with a two-level structure. Our CRNN combines a CNN and a recurrent neural network (RNN) and trains it in an end-to-end, sequence-to-sequence manner. We first describe the background of our work and introduce these methods below, followed by a description of the Hainan gibbon dataset that we use to develop and test the methods, and the acoustic data pre-processing. We then introduce the evaluation metrics used to assess performance, including the new method we propose for evaluating matching between predicted phrases and labelled phrases. Finally, we describe the computational experiments we conduct to test performance and robustness.

2.1 | Background

CNNs have become popular for acoustic signal processing because they enable automatic feature extraction, without manual feature selection. A variety of such methods has been used for animal sound detection and classification, including pre-trained CNNs (ResNet50: Efremova et al., 2019; Zhong et al., 2020), (LeNet: Jiang et al., 2019; Shiu et al., 2020), (VGGNet: Ibrahim et al., 2020; Nanni, Costa, et al., 2020), customised CNN architectures such as context-adaptive neural networks (CA-NNs: Lostanlen et al., 2019), joint detection and classification CNNs (JDC-CNNs: Kong et al., 2017), and 1D multi-view CNNs (Xu et al., 2020). CNNs tend to perform better than traditional machine learning techniques like SVMs, random forests and K-nearest neighbour (Florentin et al., 2020; Kiskin et al., 2020; Lostanlen et al., 2018; Mac Aodha et al., 2018; Salamon et al., 2017; Xu et al., 2020).

However, CNNs still require audio recordings to be split into segments of fixed length and they produce a single prediction for each segment. Longer audio segments contain more information and so tend to result in better classification by CNN. Shorter segments allow CNNs to make predictions at a higher temporal resolution, but these predictions tend to have worse classification performance because each segment contains less information. For example, in the analysis of the Hainan gibbon data that we consider, Dufourq et al. (2020a) have found that a CNN with 10-s segments (which is close to the length of the longest phrases in the data) achieved better performance than one with segments of 1, 1.5 or 2 s (which is shorter than the length of most phrases in the data). In the analysis of whale whistle data, Jiang et al. (2019) have proposed using a double CNN structure (a small window for high-resolution detection and a big window that takes the results from the small window CNN for classification) to mitigate this dilemma. However, this structure increases the complexity of the model and introduces training difficulties, while it does not effectively improve detection accuracy.

CNNs tend to gather information at high resolution but are unable to use broader-scale contextual information, such as whether or not the current point in a recording is preceded by a call signal. RNNs, on the other hand, integrate information from a sequence of time windows but they lack the ability to extract information directly from recordings, or spectrogram representations of recordings. Like RNNs, HMMs can model long-term temporal correlation in animal sound recordings (Putland et al., 2018; Stowell et al., 2017), although they are unable to extract features from the acoustic data and require this to be done beforehand.

CNNs and RNNs are often combined to leverage the strength of both. Applications include sound event detection in daily life (Çakır et al., 2017), vocabulary tasks (Sainath et al., 2015), text recognition (Shi et al., 2017), and in ecology, bird sound, koala sound and whale sound detection (Cakir et al., 2017; Himawan et al., 2018; Madhusudhana et al., 2021). A variety of architectures have been used; for example, (Madhusudhana et al., 2021) proposed using a CNN to extract acoustic features on whale's notes and then employing an RNN to learn temporal features. To do so, they input a fixed-length sequence of segments and predict a single label. However, this requires some prior knowledge to select an appropriate sequence length, taking into account the duration of a vocalisation and intervals between vocalisations. In this paper, we propose two methods to overcome the above limitations. They facilitate learning temporal correlations of small segments that are combined into syllables and then into phrases.

2.2 | Method 1: CNN + HMM

Our first method is to employ CNNs to extract visual features from spectrogram images and predict the presence of gibbon phrases on

each 1-s segment. However, this method treats each segment independently, which can result in many false positives and false negatives. To capture temporal patterns between these small segments, we use an HMM to model temporal correlations between consecutive segments to improve the predictions from CNN.

Normally, the CNN takes spectrogram images as input and outputs a probability (see Figure 2) measuring the confidence of a segment containing a gibbon syllable, or a binary classification (e.g. 1 or 0 indicating detecting a gibbon syllable or not) obtained by thresholding this output. We also consider using the linear output from the last dense layer of CNN, as the values are real and we may assume Gaussian distribution for these values. The latter two forms of output (depending on which method is being used) are then treated as the observations in an HMM to model the temporal correlation in the occurrence of positive segments, as detailed below.

2.2.1 | CNN

We adapt the two commonly used CNN architectures for animal call detection: VGGNet (Simonyan & Zisserman, 2015) and LeNet (Lecun et al., 1998). As shown in Figure 2, both CNNs are configured with the same input dimension for the grey-scale spectrogram images; that is, $32 \times 32 \times 1$. They contain convolutional layers and dense layers. Features are extracted from the input spectrogram images by convolutional layers and sent to dense layers for classification. A sigmoid function following the last dense layer produces a probability output, indicating the confidence of a segment containing a vocalisation or not. A confidence threshold can be set to decide the class membership. If the confidence is greater than the threshold, we infer a positive detection. Here we consider the threshold as a hyperparameter and use grid search to find the optimal values automatically. The search range goes from 0.1 to 0.9 with a step size of 0.1.

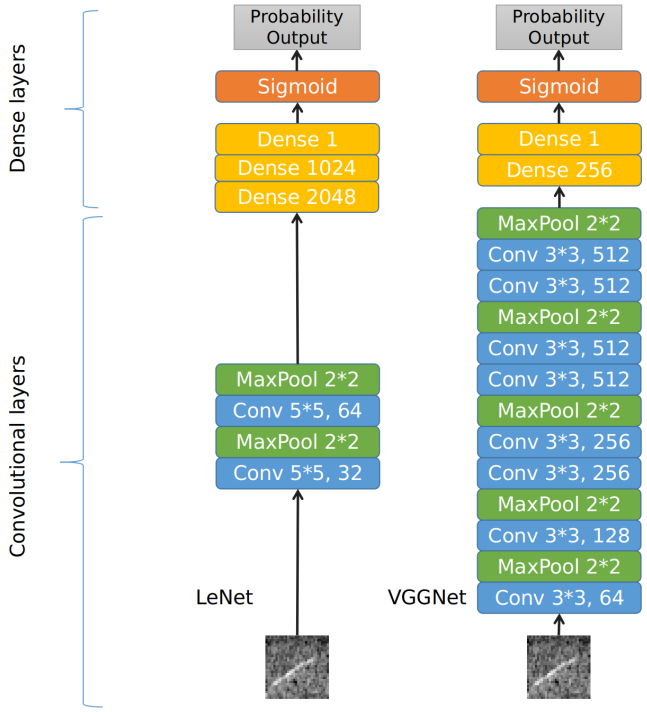


FIGURE 2 LeNet and VGGNet architectures. The structure contains convolutional and max-pooling layers to extract information from images, and dense layers followed a sigmoid function for a probability output

The LeNet has two convolutional layers with 32 and 64 filters respectively and a kernel size of 5×5. To reduce the computational cost for further processing, each convolutional layer is followed by a 2×2 max-pooling layer to halve the size of the convolutional features. After the last max-pooling layer, LeNet has three dense layers. Each layer except the output layer has a rectified linear unit (ReLU) as the activation function. For each dense layer, we set a dropout rate (Srivastava et al., 2014) of 0.5, which means deactivating half of the neurons in a layer randomly at the training stage. This helps promote the generalisation of the model and mitigate against overfitting.

Our VGGNet has a similar architecture to VGG16 (Simonyan & Zisserman, 2015), which consists of five convolutional blocks (each containing two or three convolutional layers and one max-pooling layer) and two dense layers. Each convolutional layer has a number of 3×3 kernels, and the number of filters for each layer is set to 64, 128, 256, 256, 512, 512, 512 and 512. The last convolutional layer in each block is followed by 2×2 max-pooling layers to perform downsampling on the convolutional features. To reduce the computational cost, we remove one convolutional layer from each convolutional block compared to the original VGG16. Figure 2 shows the architecture of LeNet and our customised VGGNet.

2.2.2 | Post-processing with a Hidden Markov Model

Approaches like those described above ignore the temporal correlation in vocalisations that is present in hierarchically structured data as described in the Introduction. To deal with this, we add a post-processing step where a supervised HMM is employed to learn temporal correlations between consecutive segments. This is then used to refine the predictions obtained from the previous techniques. We describe the process in more detail below.

HMMs contain Markov chains to model the evolution of unobserved states. Here, the state is the phrase and we model the time series of phrases as a Markov chain. With a HMM, the probability of an observation sequence $\boldsymbol{o} = (o_1, o_2, \ldots, o_T)$ is given by

$$P(o_1, o_2, \ldots, o_T) = \sum_{l_1, l_2, \ldots, l_T} P(\boldsymbol{l}, \boldsymbol{o}) = \sum_{l_1, l_2, \ldots, l_T} \pi(l_0) \prod_{t=1}^{T} p(l_t | l_{t-1}) q(o_t | l_t), \quad (1)$$

where $P(\boldsymbol{l}, \boldsymbol{o})$ is the joint distribution of the unobserved states $\boldsymbol{l} = (l_1, \ldots, l_T)$ (where $l_t = 1$ if segment t is in a phrase, and $l_t = 0$ otherwise) and the observations $\boldsymbol{o}$, $p(l_t | l_{t-1})$ is the probability of transitioning from state l_{t-1} to l_t (assumed to be time invariant), and $q(o_t | l_t)$ is the emission probability, that is, the probability of observing o_t when the

label (state) is l_t. We set our HMM time step to 1 s to correspond to the CNN segment lengths.

We consider two types of observations o_t: one is binary prediction output and the other is the output extracted from the last dense layer of a CNN before the sigmoid function. We assume the second type will give finer-grained information on the prediction. For binary o_t, we assume the emission probability $q(o_t|l_t)$ to be a Bernoulli distribution.

$$q(o_t|l_t) = Ber(o_t;\theta_{l_t}), \tag{2}$$

where the θ_{l_t} is the probability of observing o_t when the true state is l_t.

When o_t is the output from the last dense layer of the CNN, we assume either that o_t has a Gaussian distribution or a Gaussian mixture distribution with emission probabilities.

$$q(o_t|l_t) = N(o_t;\mu_{1l_t},\sigma_{1l_t}) \text{ or } q(o_t|l_t) = \sum_{k=1}^{K} w_{kl_t} N(o_t;\mu_{kl_t},\sigma_{kl_t}), \tag{3}$$

where μ_{kl_t} and σ_{kl_t} are the mean and variance of the kth distribution when in state l_t, and w_{kl_t} is the kth mixture weight when in state l_t. We have considered various numbers of components in the mixture Gaussian distributions and the searching range is $K = \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1,024\}$. We have used a Bayesian information criterion (BIC) (Neath & Cavanaugh, 2012) to find the best K automatically during the training stage.

The parameters of the Markov chain for $\boldsymbol{l}$ and the parameters of the emission distributions for $\boldsymbol{o}$, given $\boldsymbol{l}$, are estimated from a training dataset in which both $\boldsymbol{l}$ and $\boldsymbol{o}$ are observed, by maximising $P(\boldsymbol{l}, \boldsymbol{o})$ with respect to the Markov chain and emission distribution parameters. The Viterbi algorithm is then used with these estimated parameters to predict the unobserved states in survey data in which $\boldsymbol{l}$ is unknown and $\boldsymbol{o}$ is obtained by applying the CNN to the survey data.

In order to capture the overall temporal patterns, we set the HMM sequence length to be the same as a whole audio length. We set the initial state probabilities $\pi(l_0)$ based on the first label of each labelled recording in the training data.

2.3 | Method 2: Convolutional recurrent neural network

Here we present a CRNN that can learn temporal features in spectrogram images. We adopt a sequence-to-sequence CRNN; that is, we input a sequence of T seconds of spectrogram images and output the corresponding T seconds of predictions of gibbon phrase presence in each segment. The CRNN architecture is presented in Figure 3. It consists of convolutional layers learning features, recurrent layers learning temporal correlations between features, and a fully connected layer for phrase prediction.

For the convolutional layers, we adopt the above customised VGGNet architecture, as VGGNet has shown promising performance when integrated with an RNN (Shi et al., 2017). To speed up the convergence of the CRNN, we add a batch normalisation layer after each of the last four convolutional layers (Ioffe & Szegedy, 2015). The output from the last convolutional layer has the form $\boldsymbol{F} \in \mathbb{R}^{T\times W\times H}$, where T is the number of frames in time axis, and W and H are the dimensions of feature outputs for each second of spectrogram image from the convolutional layers; that is, 1×512.

For the recurrent layers, we first stack the CNN output, building T feature vectors with the length of 512, corresponding to T seconds in the spectrograms. The feature vectors are then fed as input to recurrent layers for further processing. The RNN has a strong ability to capture contextual information within a sequence (Shi et al., 2017), which greatly mitigates the problem that a segment may only cover part of a gibbon phrase. We use two stacked gated recurrent units (GRUs) (Chung et al., 2014) with 256 hidden units. The GRU is a simplified version of long–short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997), which is specially designed to address the vanishing gradient problem (Shi et al., 2017), allowing the RNN to store a long-range of context information. The sequence length of the GRU is also T, which will eventually produce T feature vectors corresponding to T seconds in the spectrogram. In order to mitigate against overfitting and improve the generalisation capability of the network, we apply the dropout to each recurrent layer with a rate of 0.5.

The fully connected layer is built on the top of the GRU. It takes features from the GRU and outputs a sequence of predictions that are real numbers corresponding to T seconds of the input spectrogram. A sigmoid function is then used to map these numbers into the interval $[0, 1]$. If the output is greater than a threshold, then a positive label is inferred.

For Methods 1 and 2, phrase prediction is conducted on segment-based predictions; that is, consecutive segments that share the same label are combined into phrases.

2.3.1 | Configuration and hyperparameter tuning

We adopt the architecture of Shi et al. (2017) for our CRNN in terms of the number of neurons within recurrent layer, and configure CRNN with the following hyperparameters and settings: sequence length, the choice of uni- or bi-directional GRU, different optimisers, dropout rate, learning rate, batch size and the number of epochs. We run a grid search on combinations of different values of each hyperparameter and choose the ones with the best F-score. We consider pairs of sequence length T, batch size and epochs since a long sequence length will take up more memory and take a longer time to converge, which results in smaller batch size and larger epochs. For example, the pairs being experimented are (T, batch_size, epochs) = {(2, 512, 20), (4, 512, 20), (6, 256, 20), (8, 256, 20), (10, 128, 20), (40, 48, 30), (100, 24, 50), (400, 6, 50), (1200, 2, 60)}. We also tried different learning rates ({5e-3, 1e-3, 5e-4, 1e-4}) and dropout rates ({0.3, 0.5}). The training time of each combination varies from 2 to 13 hr, and in total, the 72 combinations take about 400 hr to run.

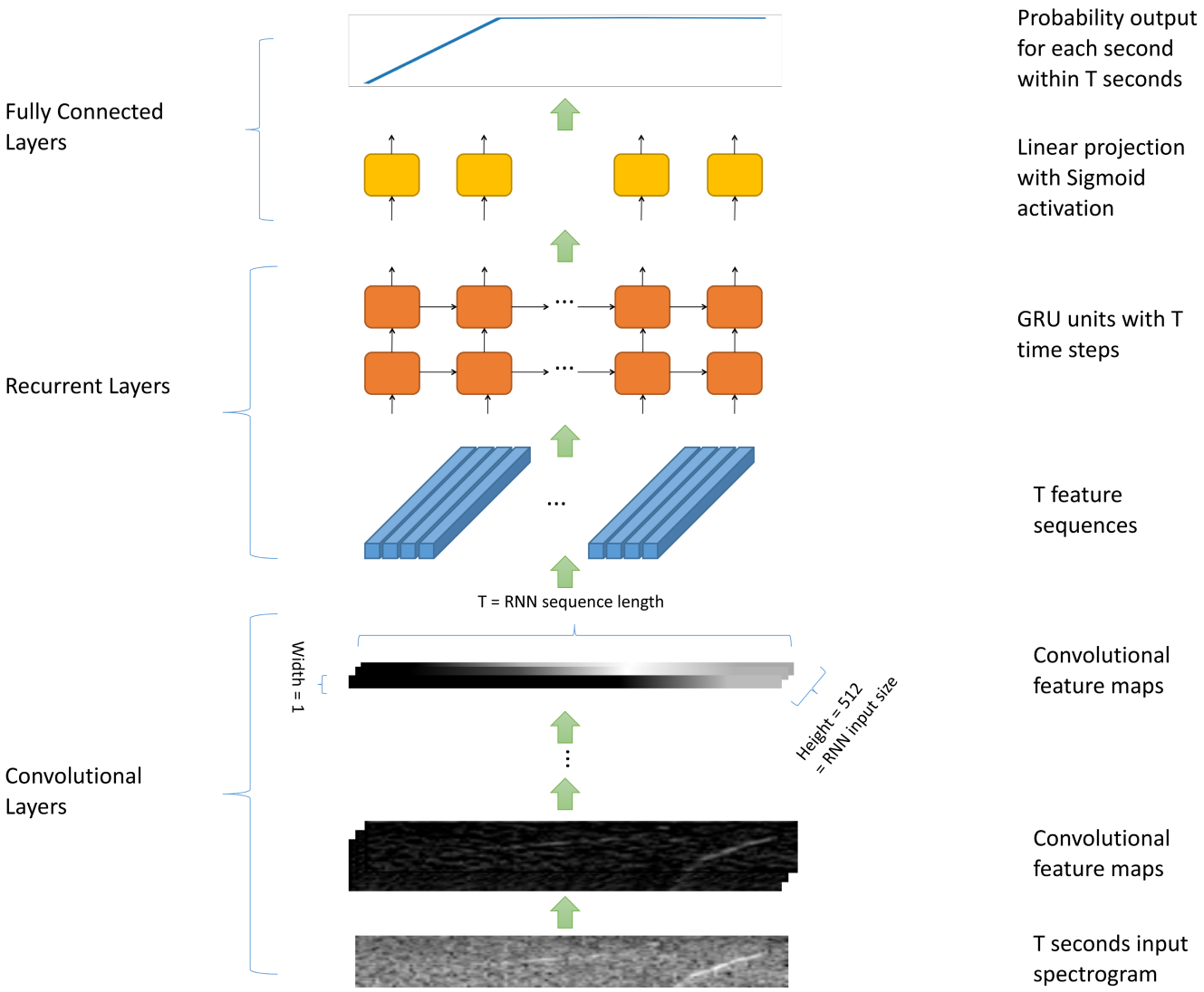


FIGURE 3 Overview of the CRNN network structure with components (1) convolutional layers, which extract features from a *T*-second spectrogram image sequence, (2) recurrent layers, which take convolutional layers' output feature vectors stacked over a channel axis and (3) a fully connected layer as output layer with a sigmoid function to map raw output into a probability

TABLE 1 Hyperparameter configuration

Hyperparameter	Setting
Sequence Length	400
Batch size	6
Epochs	50
GRU	Uni-directional GRU
Optimiser	ADAM
Learning rate	$1e-3$
Dropout rate	0.5

A bi-directional GRU in a CRNN takes much more time to train and does not improve the performance in our experiments. Thus we only apply a uni-directional GRU in the CRNN. We adopt a robust, commonly used optimiser ADAM (Kingma & Ba, 2017). Table 1 lists the optimised configurations.

2.4 | Sound data description

We evaluate our methods using recordings of Hainan gibbons *Nomascus hainanus*, whose vocalisations are made up of phrases of varying lengths containing a variable number of syllables. Like most gibbon species, they live in a forest that is too thick for visual surveys to be an effective means of surveying them. They are therefore surveyed acoustically (Deng et al., 2014), typically by people listening for their calls (Kidney et al., 2016), although surveying is increasingly being done by digital devices. We used the open-access dataset of Dufourq et al. (2020b), which contains 28 8-hr recordings of Hainan gibbon calls collected in Bawangling National Nature Reserve, Hainan, China with eight Song Meter SM3 recorders. Recordings start from 5 a.m. or 6 a.m. and last 8 hr each day, with an acoustic sampling rate at 9.6 KHz and a bit depth of 16. A short vocal syllable (called a by Dufourq et al., 2020a) that lasts between 0.2 and 2.75 s is the smallest acoustic unit we consider. A phrase, as demonstrated

in Figure 4, consists of one to six consecutive syllables separated by short intervals, and lasting between 1 and 11 s (see Figure 5a). There are typically longer pauses between phrases than between syllables within a phrase (Dufourq et al., 2020a). Without taking account of the structured nature of phrases, it is difficult to detect and separate adjacent phrases successfully in a fully automatic manner, as the duration of a phrase and intervals between phrases may vary substantially.

According to Dufourq et al. (2020a), there are ambient noises such as bird calls and rain events, which could affect the classifier performance. Following (Lin et al., 2013), we calculated approximate gibbon phrase signal-to-noise ratio (SNR) using $\text{SNR} = P_{\text{phrase\_signal}} - P_{\text{noise}}$, where $P_{\text{phrase\_signal}}$ represents the root mean square (RMS) amplitude of each gibbon phrase, and P_{noise} represents the RMS amplitude over 1 s segment prior to the beginning of each gibbon phrase. The SNRs of the gibbon phrases are well distributed from around −20 to 30 dB with a mean of around 2.6 dB, which makes the detection task tricky as only a few phrases were observed with high SNR. (see Figure 5b). Phrases would typically be the unit of interest for monitoring and therefore, in these data, it is the phrases, not syllables that are labelled. Our task here is to predict phrases. There is a total of 9,199 s of gibbon sound, consisting of 1,858 labelled phrases in the 28 acoustic files, and 797,201 s without gibbon sounds.

2.5 | Data pre-processing

We divide each 8-hr recording into 1-s segments without overlap. The audio datasets are labelled per phrase so that the start and stop times are recorded for every phrase, from which every 1-s segment is allocated a class (positive/negative), whether it contains a gibbon vocalisation or not. Some segments do not because there are pauses between syllables within phrases. Each segment is then automatically converted to a log-scale Mel-spectrogram representation with 32 frequency bins and a Hanning window with 512 frames window size and 256 frames overlapping. This setting resulted in a good balance between time and frequency resolution in our preliminary investigation. Following Dufourq et al. (2020a), we restrict attention to the frequency interval between 1 and 2 kHz. Segments in the form of images (log Mel-spectrogram) are resized into 32 × 32 pixels. Min–max normalisation is applied on each image for the convenience of the CNN application. After pre-processing, each sound recording generates a sequence of images, stored in chronological order with each second labelled as positive (noted as 1) if it is in a gibbon phrase; otherwise, negative (noted as 0).

2.6 | Evaluation metrics

The performance of the gibbon vocalisation detection algorithm can be evaluated in a variety of ways. The most appropriate measure of performance will depend on the intended use. We focus here on the use of acoustic detectors to monitor populations and estimate their distribution and abundance. At the simplest level, monitoring involves counting the number of phrases per unit time, the *encounter rate*. If, as is common, the unit of detection is a phrase, then we want our method to accurately predict the number of phrases in a recording in order to monitor the phrase encounter rate.

Most (but not all) methods of estimating absolute wildlife abundance are designed to cope with false negatives (e.g. missed calls) but are sensitive to false positives (e.g. using sounds that are not calls of the target type). This means that these methods are generally not biased by low recall, but they may be biased by low precision. When evaluating our methods we need to take into account both recall and precision.

We evaluate the methods in three ways. For the first two, we adopt commonly used metrics: precision, recall and *F*-score:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad R = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

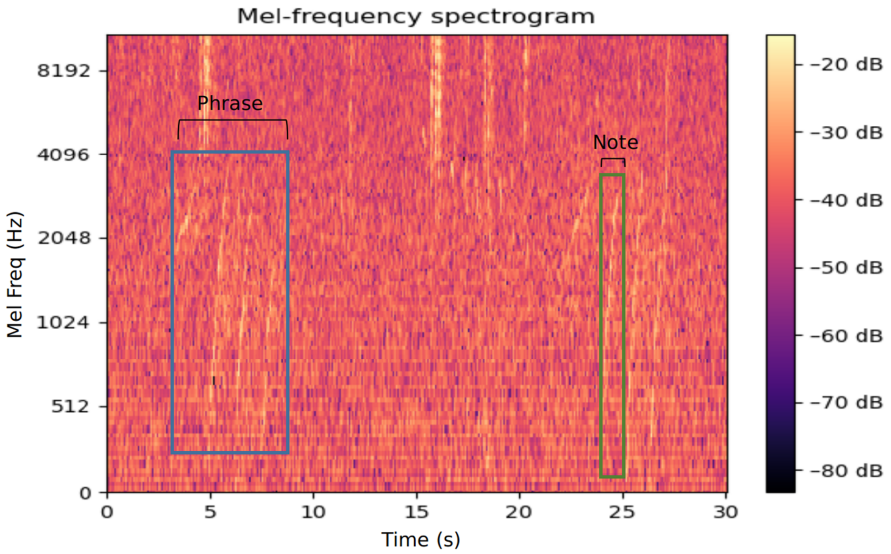


FIGURE 4 An example of two individual phrases are shown as a log scale Mel-spectrogram. Each phrase consists of several (typically 1–6) syllables. The interval between syllables is usually small compared to the interval between phrases

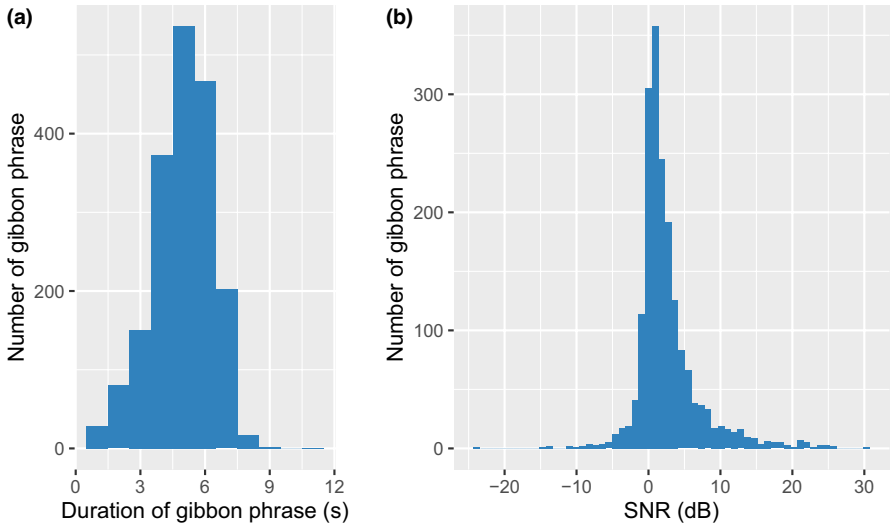


FIGURE 5 (a) Distribution of durations for all gibbon phrases. (b) Distribution of SNRs for all gibbon phrases

where TP represents the number of true positives, FP represents the number of false positives, and FN represents the number of false negatives. *F*-score (*F*) balances precision and recall, which is defined as:

$$F = \frac{2 \times P \times R}{P + R}, \quad \text{or} \quad F = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}. \tag{5}$$

We use these measures when predicting at the segment level, and when predicting at the event (phrase) level. One of the widely used sound event evaluation methods is *collar*, which aims to match the start and end of a predicted event to a true event (Mesaros et al., 2016). Here, an event, or a phrase, refers to a sequence of consecutive segments that share the same predicted label. This method is suitable for tasks requiring precise start (and end) time prediction for long blocks of audio (Lafay et al., 2017; Mesaros et al., 2019). However, for the purposes of monitoring a wildlife population, or obtaining a relative index of abundance, the collar-based method can have several drawbacks. First, if the length of phrases is highly variable, defining an appropriate threshold for matching onset and offset time is difficult to do and requires subjective choices. Second, the signal strength is quite weak at the beginning and the end of phrases (as it is with gibbon vocalisations), which may lead to annotation uncertainty (Kwon et al., 2019) and introduce bias in the collar-based evaluation metrics.

Therefore, we propose a new way to evaluate predicted against observed phrases with dynamic programming used for sequence alignment (Eddy, 2004; Sellers, 1974). The method measures the overlap between predicted phrases and labelled phrases, and finds the best match with dynamic programming, which takes into account both the number of overlaps and the degree of overlap of predicted and true phrases. Given two sequences of predicted and true phrases, the number of true positives (TP), is defined to be the matching number of predicted and true phrases obtained from dynamic programming. The number of false positives (FP) and the false negatives (FN) can be derived from TP: FP = (number of predicted phrases) − TP, and FN = (number of labelled phrases) − TP. We describe the algorithm in Appendix A.

Our third measure is simply how well we predict the total number of phrases in an audio file (the encounter rate accuracy). For each method, we calculate the encounter error rate:

$$\text{Encounter error} = \frac{|\ \text{Number of labelled phrases} - \text{Number of predicted phrases}\ |}{\text{Number of labelled phrases}}. \tag{6}$$

2.7 | Computational experiments to test performance

We use fourfold cross-validation that involves splitting our 28 separate audio recordings equally into fourfold. We use onefold for testing and the remaining three for training and validation. We iterate four times so each fold is used for testing once. The results are averaged over these four runs. Our splitting method on audio files prevents data leakage. For the CRNN, when forming segments into sequences, we use 50% overlapping during training. That is, the consecutive sequences have half of their segments in common. This is used to increase the amount of training data. Even though our dataset is imbalanced with the majority being non-gibbon calls, we have not employed any over- and downsampling techniques, as they might either generate noisy data or break the temporal correlation.

We compare the performance of our proposed methods with a classic technique commonly used in animal sound detection; that is, SVM with a radial basis function (RBF) kernel (Salamon et al., 2017) on Mel-frequency cepstral coefficients (MFCCs). For each 1-s segment, we extract MFCC using 40 Mel bands, keeping the first 25 coefficients, then calculating delta (first-order difference) and acceleration (second-order difference) MFCC from these values. As implemented in Salamon et al. (2014), 11 summary statistics including minimum value, maximum value and mean value, are used to summarise MFCC coefficients, producing feature vectors of length 275 per second. After generating MFCC features we train an SVM model

for phrase prediction. We also consider the CNNs with spectrogram images as another baseline. This aims to demonstrate the strength of HMM post-processing in learning temporal relationships, thus leading to improved performance.

Since the main purpose of HMM post-processing is to smooth out some of the variations in predictions, we also compare the HMM against a simple moving average method, which averages the probability outputs on a sequence of segments from the CNN. We applied a grid search with a training/validation set to decide the sequence length, with the search range $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1,024\}$ together with the confidence thresholds from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. However, the experiments show that the moving average post-processing does not improve performance because the optimised sequence length is always learned to be 1 and thus it leads to the same result as the CNN or SVM.

To further test the robustness of the proposed method, we applied time stretching (Salamon et al., 2017), pitch shifting (Lostanlen et al., 2019; Pandeya et al., 2018) and random cropping (Nanni, Maguolo, & Paci, 2020) to the original dataset during the evaluation procedure, which adds different degrees of complexity to the dataset. Following Salamon et al. (2017), we set the time stretch ratios to be $\{0.81, 0.93, 1.07, 1.23\}$. We set the pitch shift using values $\{-1, -0.5, 0.5, 1\}$, since our data are within the narrow frequency band between 1 and 2 kHz. In the random crop simulation, we randomly crop out various proportions $\{0.1, 0.2, 0.3, 0.4\}$ of the data from each segment to simulate missing data from a microphone. In addition, we experimented with training the model with fewer samples. That is, for each iteration in the cross-validation, we only use a proportion of the training set to train our model using the proportion in $\{0.75, 0.5, 0.25\}$.

CRNN and CNN models are implemented in Python 3.6 using Pytorch (Paszke et al., 2019), and audio is processed with librosa (McFee et al., 2015). The HMM models are implemented using the hmmlearn and scikit-learn (Pedregosa et al., 2012) libraries and the SVM is implemented with the cuML (Raschka et al., 2020) library. Experiments are done on a machine running Ubuntu 20.04 LTS with an Intel i7-9700 CPU, 32GB of RAM and an Nvidia RTX 2060 super 8 GB Graphic Processing Unit.

3 | RESULTS

In this section, we compare methods with segment-based evaluation metrics, the proposed phrase-based evaluation metrics, and the encounter error rate based on the original gibbon dataset. We also compare the performance of different methods with simulated datasets, as described in the previous section.

3.1 | Segment-based performance

As shown in Figure 6, a CRNN with a sequence length of 400 achieves the best F-score, precision and recall among all the methods. Its *F*-score is between 10% and 37% higher than VGGNet, LeNet and SVM with MFCC features. This result demonstrates the strength of a CRNN in learning temporal relationships between segments.

In contrast, the HMM, which also has the ability to learn temporal correlation, only improved the performance modestly. More specifically, an HMM with a Bernoulli emission probability increases F-scores on VGGNet, LeNet and SVM only by about 4%. The HMM with a Gaussian mixture model (GMM) emission probability increases the recall for the VGGNet and LeNet models but decreases the precision as well as the F-score. Therefore, we conclude that an HMM with a Bernoulli emission probability is better than an HMM with a GMM distributed emission probability for refining segment predictions.

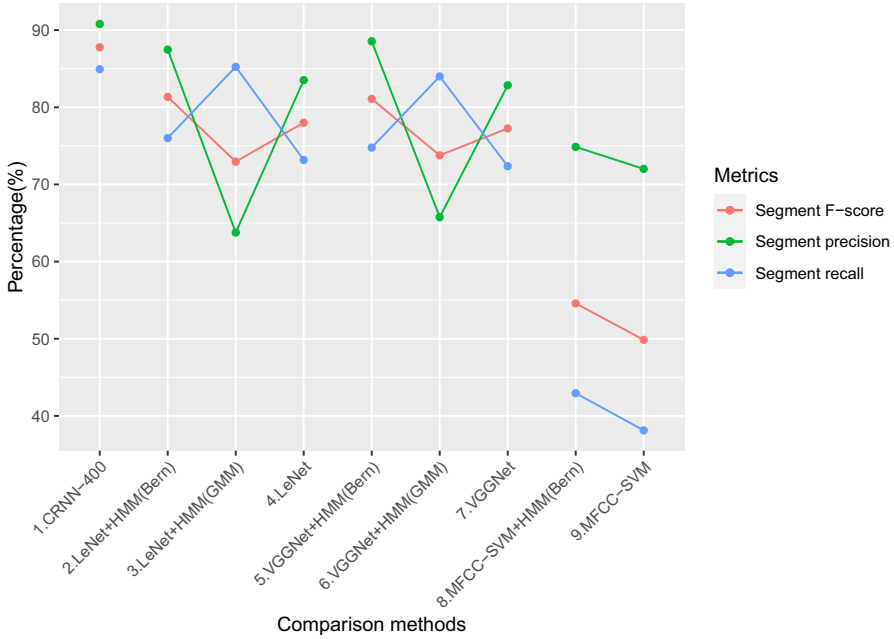


FIGURE 6 The segment-based precision, recall and *F*-score for CRNN structures with sequence length 400, which has the best performance among different sequence lengths in our experiment, 2 CNNs such as VGGNet and LeNet, CNNs with HMM post-processing, a baseline SVM model and SVM with HMM (Bernoulli) post-processing

The effect of sequence length on the performance of the CRNN is shown in Figure 7. The F-score increases gradually with the sequence lengths, up to a length of 400. The CRNN with a sequence length of 400 (CRNN-400) achieves the best *F*-score of 87.77%. There is a trade-off between precision and recall as functions of sequence lengths, while both precision and recall show an upward trend.

3.2 | Phrase-based performance

Figure 8 compares the phrase prediction performance of each method. The CRNN achieves the best *F*-score and recall, and the second-best precision. The HMM post-processing method greatly improves the performance of CNN. Specifically, the HMM with a Bernoulli emission probability boosts LeNet's and SVM's precision from 60% and 45% to 95% and 86% respectively. This makes it attractive for use with common abundance estimation methods that do not account for false positives. As with the segment-based results, all the CNNs perform better than the classic SVM.

3.3 | Encounter rate accuracy

Figure 9 shows the encounter rate accuracy for all the techniques. The CRNN identifies 1,867 phrases, leading to the encounter error rate of 0.5% on 1,858 labelled phrases. The CNN models

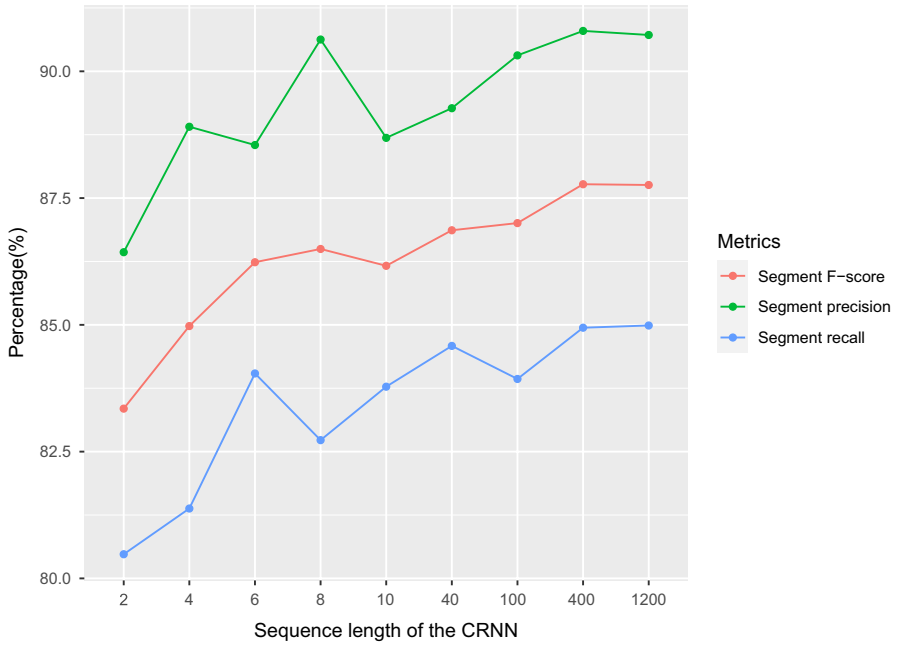


FIGURE 7 The segment-based precision, recall and *F*-score of the CRNN on different sequence lengths

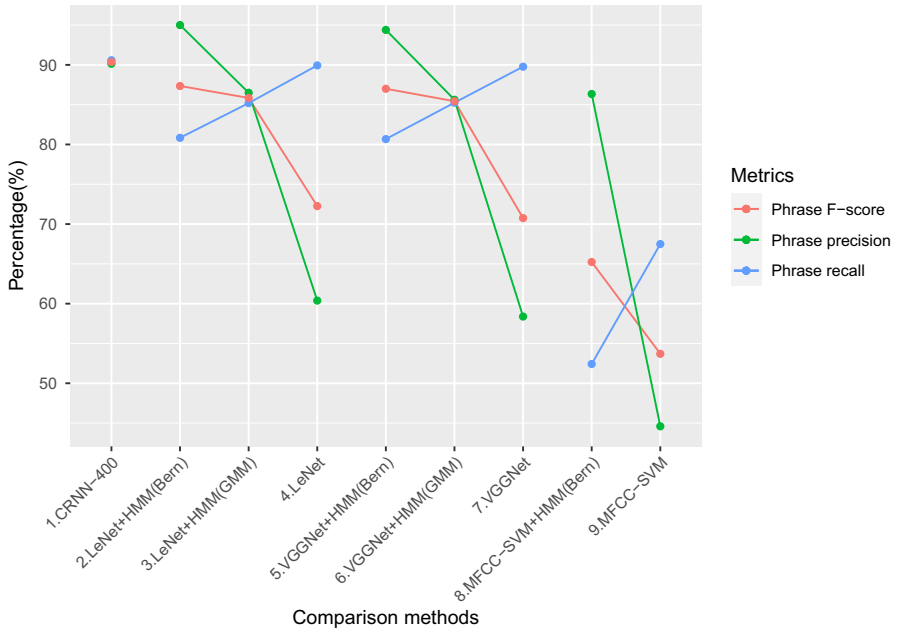
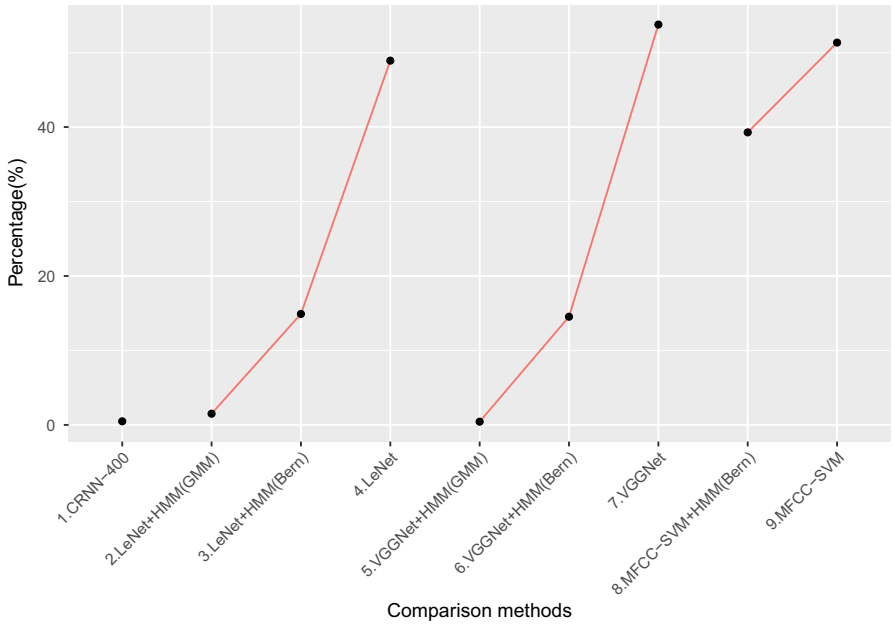


FIGURE 8 The phrase prediction performance for CRNN, two CNNs such as VGGNet and LeNet, CNNs with HMM post-processing, a baseline SVM model and SVM + HMM (Bernoulli) post-processing

FIGURE 9 The percentage error of predicted gibbon phrase number based on true phrase number for CRNN, two CNNs such as VGGNet and LeNet, CNNs with HMM post-processing, SVM and SVM with HMM (Bernoulli) post-processing



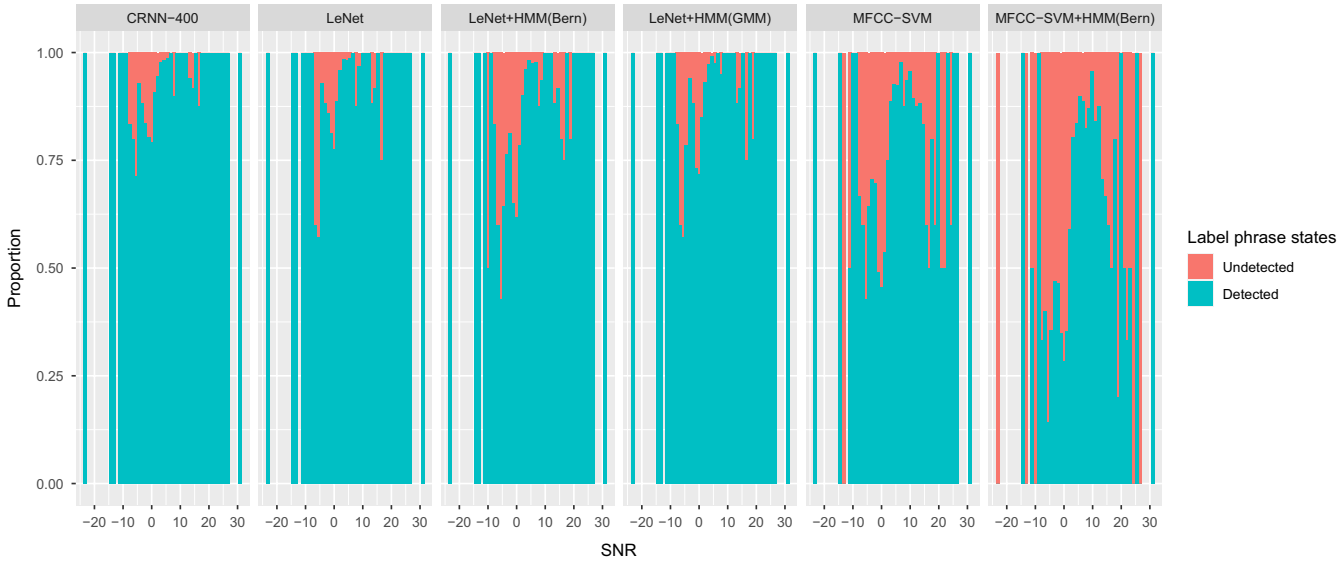


FIGURE 10 The proportion of detected gibbon phrase corresponding to the SNRs for CRNN, LeNet, LeNet with HMM post-processing, SVM and SVM with HMM (Bernoulli) post-processing

alone produced a large number of false positive predictions, overestimating the number of phrases by 54% and 49% respectively. The HMM post-processing method is effective in correcting false positive detection. For example, the HMM with Gaussian mixture achieves the encounter error rate of 1.5% on LeNet and 0.4% on VGGNet, which is the best performance among all results. Similarly, the HMM with a Bernoulli emission probability corrects the predictions on SVM, reducing the error rate from 51% to 39%.

3.4 | Robustness

First, we assess the impact of the SNR on performance. To do this, we collect gibbon phrases together and categorise their SNRs into bins. Then we calculate the *detection* and *non-detection* frequencies for phrases in each bin. Intuitively, a higher SNR value indicates a stronger signal, which will lead to a higher proportion of detections. As we can see in Figure 10, the CRNN outperforms all the other methods at all levels of SNRs. It has lower detection proportions when the SNR is lower but is still much better than the other methods. The SVM is worst; for example, when the SNR is around −20, the detection proportion may drop to zero. The HMM lowers the detection rates on LeNet and SVM. On investigating further, we found that although the HMM is better at dealing with false positive phrase predictions, it also smooths out true positive phrase predictions.

As we do not have access to other suitably hierarchically structured datasets, we created simulated datasets of varying degrees

of complexity to assess the robustness of our methods. To do so, we employed commonly used data transformation methods including time stretching, pitch shifting and random cropping of the test data only to simulate adverse conditions in the real world, such as missing acoustic signals, or malfunctioning microphones. We train our models on non-augmented data and then test them on the augmented data to see to what degree our methods will be affected. We present our results in Figure 11 and observe that our methods are more robust to these perturbations than the LeNet or SVM. For pitch shifting, the LeNet with an HMM performs the best; its *F*-score drops by no more than 8% on both segment- and phrase-based prediction when pitch shifts to 1.0. The CRNN is more affected in this case, while the SVM degrades greatly with an increased pitch shift. For random cropping, when the rate increases to 0.4, the LeNet's *F*-score drops 22% on phrase prediction and 13% on segment prediction. This shows that the performance of phrase prediction greatly depends on how well individual segments are predicted. All methods seem to be insensitive to time stretching because there is little information loss with this procedure.

Figure 12 presents the results for the experiments with less training data. The *F*-scores on all the methods decrease with less training data. In segment prediction, the HMM post-processing lifts the performance of the LeNet and the SVM in a stable way, while the CRNN still performs the best overall. The HMM shows great potential in phrase detection; it outperforms the CRNN when the training data are very small.

4 | DISCUSSION

As more acoustic surveys move from using human detectors to using digital detectors, reliable identification of target species vocalisations in recordings by machine learning methods become increasingly important for monitoring and assessment. Identification is particularly difficult for vocalisations that have a two-level (or multi-level) structure, like the data we consider, in which a varying number of syllables (first level) occur within phrases (second level) of varying duration. Although segment-based methods like CNNs can

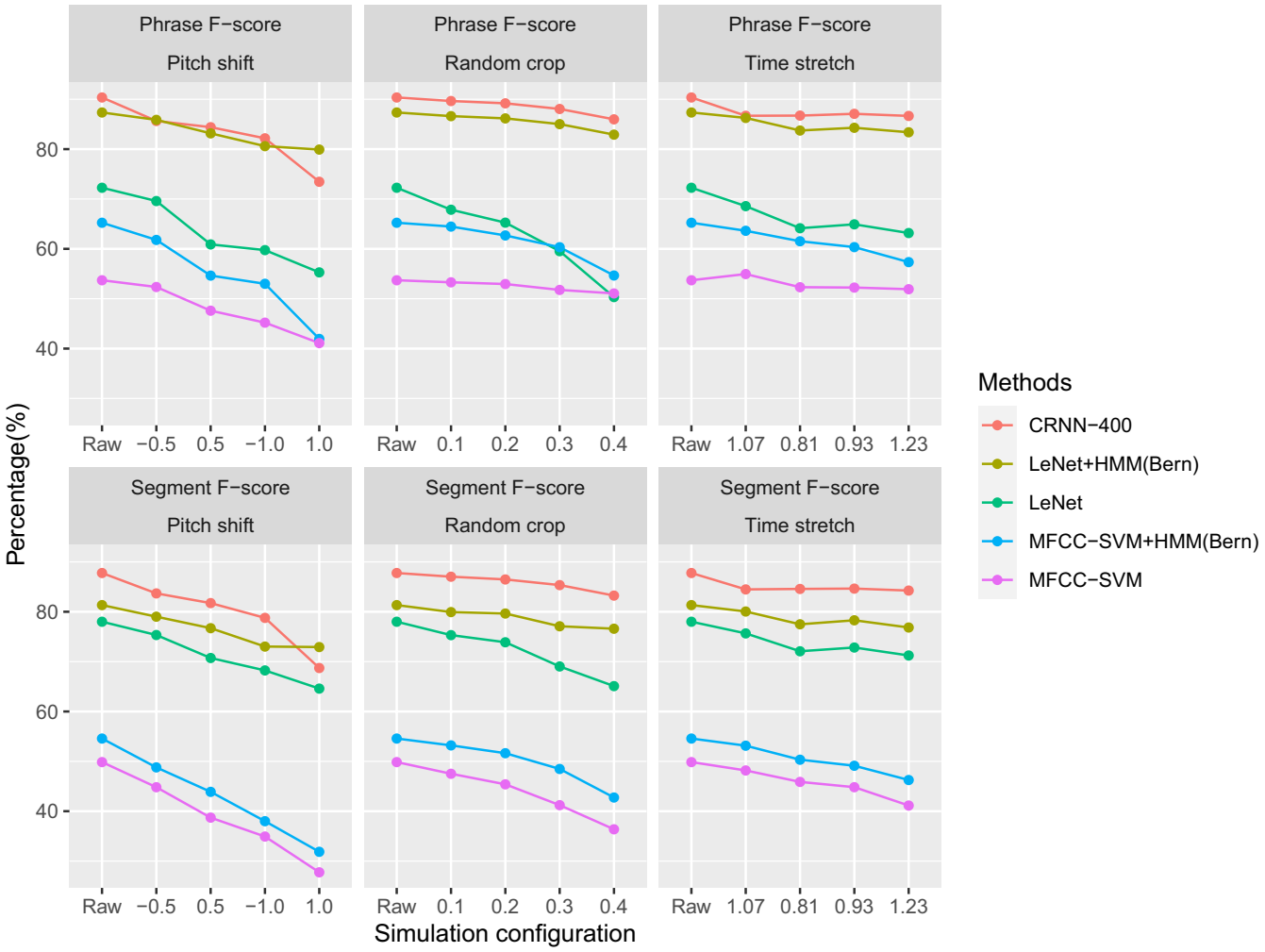


FIGURE 11 The segment and phrase-based performance for SVM, SVM with HMM (Bernoulli) post-processing, LeNet, LeNet with HMM (Bernoulli) post-processing and CRNN, using data augment methods including time stretch by 4 *ratios* = (0.81, 0.93, 1.07, 1.23), pitch shift by 4 *values* = (− 1, − 0.5, 0.5, 1) and random crop by 4 *ratios* = (0.1, 0.2, 0.3, 0.4)

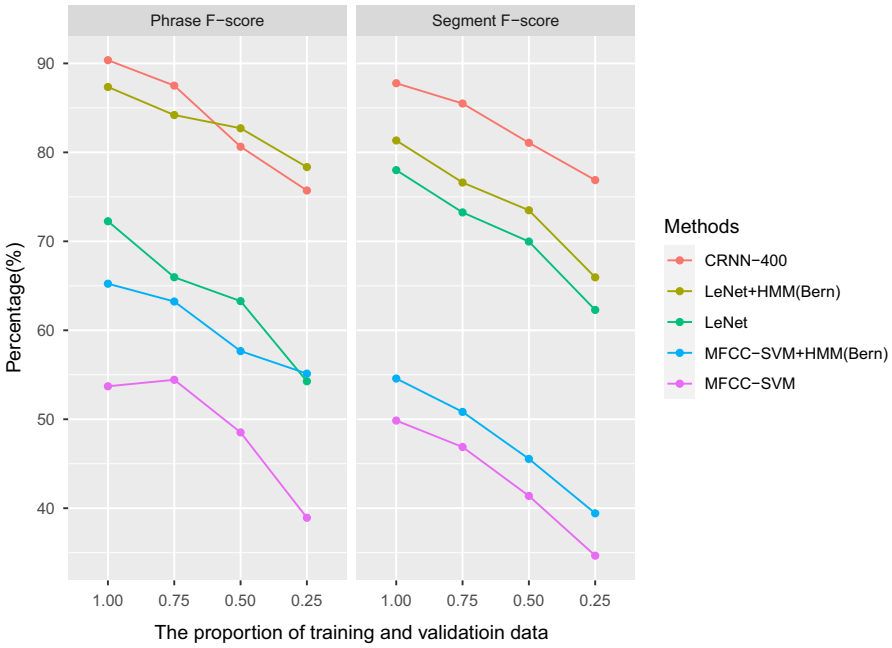


FIGURE 12 The segment and phrase-based performance for SVM, SVM with HMM (Bernoulli) post-processing, LeNet, LeNet with HMM (Bernoulli) post-processing, and CRNN with proportional training and validation data by 3 *ratios* = (0.75, 0.50, 0.25)

be designed to detect a partial phrase, their predictions with short segments tend to perform poorly because the shorter the segment, the less information it contains, and this produces false positives and false negatives at the segment level, which adversely affects the phrase prediction.

The methods we develop above perform substantially better at this task than CNNs and SVMs. A CRNN performs best at predicting both segments and phrases, and a CNN combined with an HMM performs next best. We note that an HMM is much more computationally efficient than a CRNN in that the post-processing HMM with a Bernoulli emission probability takes 3.52 s to train and 0.31 s to predict and the HMM with a GMM takes 1,205 s to train and 0.56 s to predict. The HMM with a GMM takes much longer than the HMM with a Bernoulli because we need to decide the number of Gaussian components to use in the GMM with BIC through a grid search. In comparison, the CRNN-400 takes about 10 hr to train and 62 s to predict. Also, HMMs can be added as a post-processing step to any pre-trained segment-based machine learning or deep learning method with no additional modelling.

CNNs and SVMs perform well if an appropriate segment can be selected (one that is long enough to include phrases but not so long as to include multiple phrases). However, when intervals between phrases are variable and not consistently larger than intervals within phrases, the choice of segment length can be difficult and case specific, and an incorrect choice may lead to either overestimating or underestimating the number of phrases. Our methods perform well in such cases and are shown to be more robust to perturbations of the acoustic recordings.

We also proposed and implemented a way of evaluating prediction performance that measures how well phrases are predicted, rather than how good predictions are at the somewhat arbitrary time unit that acoustic files are segmented into for the application of machine learning (ML) methods. The method is also preferable to the commonly used collar-based method because it does not require phrase starting time to be identified (something that can be error-prone) and it is less sensitive to annotation ambiguity. Users can customise the prediction performance evaluation method by changing the threshold that defines the overlap of predicted and labelled phrases, to be more or less strict in defining the matching of phrases.

Finally, although we have only applied ML methods to phrases of variable length that contain a variable number of syllables, we anticipate that the CRNN and CNN with HMM methods will perform well on phrases comprising continuous vocalisations of variable length too, as there is nothing in these methods that requires phrases to be composed of separate syllables. When phrases are of variable length, methods based on recognising vocalisations in segments or windows of fixed length will tend to break phrases into multiple parts if segments or windows are small, and so over-estimate the number of phrases, or to combine phrases with periods of non-vocalisation if the segments or windows are large, and so under-estimate the number of phrases. The CRNN and CNN with HMM methods do not suffer from this problem. Our method might also be useful for other animal species whose calls share similar acoustic characteristics, including birds (Chen & Maher, 2006; Somervuo et al., 2006) and whales (Bergler et al., 2019; Jiang et al., 2019); however, this will need further validation.

AUTHORS' CONTRIBUTIONS

Y.W., J.Y. and D.L.B. conceived the ideas; Y.W. and J.Y. developed the methods; Y.W. analysed the data; Y.W. and D.L.B. led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

ACKNOWLEDGEMENT

Y.W. is partly funded by the China Scholarship Council (CSC) for Ph.D. study at the University of St Andrews, UK.

CONFLICT OF INTEREST

None of the authors have a conflict of interest.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/2041-210X.13873.

DATA AVAILABILITY STATEMENT

Data are available from the Zenodo https://doi.org/10.5281/zenodo.3991714 (Dufourq et al., 2020b). All code scripts are available from the Zenodo https://doi.org/10.5281/zenodo.6461670 (Wang et al., 2022).

ORCID

Yuheng Wang https://orcid.org/0000-0003-3335-8706
Juan Ye https://orcid.org/0000-0002-2838-6836
David L. Borchers https://orcid.org/0000-0002-3944-0754

REFERENCES

Alonso, J. B., Cabrera, J., Shyamnani, R., Travieso, C. M., Bolanos, F., Garcia, A., Villegas, A., & Wainwright, M. (2017). Automatic anuran identification using noise removal and audio activity detection. *Expert Systems with Applications*, *72*, 83–92. https://doi.org/10.1016/j.eswa.2016.12.019

Bergler, C., Schröter, H., Cheng, R. X., Barth, V., Weber, M., Nöth, E., Hofer, H., & Maier, A. (2019). ORCA-SPOT: An automatic killer whale sound detection toolkit using deep learning. *Scientific Reports*, *9*, 1–17.

Buckland, S. T. (2006). Point transect surveys for songbirds: Robust methodologies. *The Auk*, *123*, 345–357. https://doi.org/10.1093/auk/123.2.345

Buckland, S. T., Anderson, D. R., Burnham, K. P., Laake, J. L., Borchers, D. L., & Thomas, L. J. (2001). *Introduction to distance sampling*. Oxford University Press.

Cakir, E., Adavanne, S., Parascandolo, G., Drossos, K., & Virtanen, T. (2017). Convolutional recurrent neural networks for bird audio detection. *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 1744–1748. https://doi.org/10.23919/EUSIPCO.2017.8081508.

Çakır, E., Parascandolo, G., Heittola, T., Huttunen, H., & Virtanen, T. (2017). Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*, 1291–1303. https://doi.org/10.1109/TASLP.2017.2690575

Chen, Z. X., & Maher, R. C. (2006). Semi-automatic classification of bird vocalizations using spectral peak tracks. *Journal of the Acoustical Society of America*, *120*, 2974–2984. https://doi.org/10.1121/1.2345831

Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Colonna, J. G., Cristo, M., Salvatierra, M., & Nakamura, E. F. (2015). An incremental technique for real-time bioacoustic signal segmentation. *Expert Systems with Applications*, *42*, 7367–7374. https://doi.org/10.1016/j.eswa.2015.05.030

Deng, H., Zhou, J., & Yang, Y. (2014). Sound spectrum characteristics of songs of Hainan Gibbon (*Nomascus hainanus*). *International Journal of Primatology*, *35*, 547–556. https://doi.org/10.1007/s10764-014-9767-3

Dufourq, E., Durbach, I., Hansford, J. P., Hoepfner, A., Ma, H., Bryant, J. V., Stender, C. S., Li, W., Liu, Z., Chen, Q., Zhou, Z., & Turvey, S. T. (2020a). Automated detection of hainan gibbon calls for passive acoustic monitoring. *bioRxiv*. https://doi.org/10.1101/2020.09.07.285502.

Dufourq, E., Durbach, I., Hansford, J. P., Hoepfner, A., Ma, H., Bryant, J. V., Stender, C. S., Li, W., Liu, Z., Chen, Q., Zhou, Z., & Turvey, S. T. (2020b). Automated detection of hainan gibbon calls for passive acoustic monitoring. *Zenodo*, https://doi.org/10.5281/zenodo.3991714.

Eddy, S. (2004). What is dynamic programming? *Nature Biotechnology*, *22*, 909–910. https://doi.org/10.1038/nbt0704-909

Efremova, D. B., Sankupellay, M., & Konovalov, D. A. (2019). Data-efficient classification of birdcall through convolutional neural networks transfer learning. *Digital Image Computing: Techniques and Applications*, *DICTA 2019*, *2019*, 2–4. https://doi.org/10.1109/DICTA47822.2019.8946016

Florentin, J., Dutoit, T., & Verlinden, O. (2020). Detection and identification of European woodpeckers with deep convolutional neural networks. *Ecological Informatics*, *55*, 101023. https://doi.org/10.1016/j.ecoinf.2019.101023

Gibb, R., Browning, E., Glover-Kapfer, P., & Jones, K. E. (2019). Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring. *Methods in Ecology and Evolution*, *10*, 169–185. https://doi.org/10.1111/2041-210x.13101

Himawan, I., Towsey, M., Law, B., & Roe, P. (2018). Deep learning techniques for koala activity detection. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, September 2018, pp. 2107–2111. https://doi.org/10.21437/Interspeech.2018-1143.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Ibrahim, A. K., Zhuang, H., Chérubin, L. M., Schärer-Umpierre, M. T., Nemeth, R. S., Erdol, N., & Ali, A. M. (2020). Transfer learning for efficient classification of grouper sound. *The Journal of the Acoustical Society of America*, *148*, EL260–EL266. https://doi.org/10.1121/10.0001943

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Jiang, J., Bu, L. R., Duan, F. J., Wang, X. Q., Liu, W., Sun, Z. B., & Li, C. (2019). Whistle detection and classification for whales based on convolutional neural networks. *Applied Acoustics*, *150*, 169–178. https://doi.org/10.1016/j.apacoust.2019.02.007

Kidney, D., Rawson, B., Borchers, D., Stevenson, B., Marques, T., & Thomas, L. (2016). An efficient acoustic density estimation method with human detectors applied to gibbons in cambodia. *PLoS ONE*, *11*, e0155066. https://doi.org/10.1371/journal.pone.0155066

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization.

Kiskin, I., Zilli, D., Li, Y., Sinka, M., Willis, K., & Roberts, S. (2020). Bioacoustic detection with wavelet-conditioned convolutional neural networks. *Neural Computing and Applications*, *32*, 915–927. https://doi.org/10.1007/s00521-018-3626-7

Kong, Q., Xu, Y., & Plumbley, M. D. (2017). Joint detection and classification convolutional neural network on weakly labelled bird audio detection. *25th European Signal Processing Conference, EUSIPCO 2017*, January 2017, pp. 1749–1753. https://doi.org/10.23919/EUSIPCO.2017.8081509.

Kwon, H., Abowd, G. D., & Plötz, T. (2019). Handling annotation uncertainty in human activity recognition. *Proceedings of the 23rd International Symposium on Wearable Computers*, ISWC '19, pp.

109–117. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3341163.3347744.

Lafay, G., Benetos, E., & Lagrange, M. (2017). Sound event detection in synthetic audio: Analysis of the dcase 2016 task results. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 2017, pp. 11–15. https://doi.org/10.1109/WASPAA.2017.8169985.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. https://doi.org/10.1038/nature14539

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324. https://doi.org/10.1109/5.726791

Lin, T. H., Chou, L. S., Akamatsu, T., Chan, H. C., & Chen, C. F. (2013). An automatic detection algorithm for extracting the representative frequency of cetacean tonal sounds. *Journal of the Acoustical Society of America*, *134*, 2477–2485. https://doi.org/10.1121/1.4816572

Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S., & Bello, J. P. (2018). Birdvox-full-night: A dataset and benchmark for avian flight call detection. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings*, April 2018, pp. 266–270. https://doi.org/10.1109/ICASSP.2018.8461410.

Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S., & Bello, J. P. (2019). Robust sound event detection in bioacoustic sensor networks. *PLoS ONE*, *14*, 1–32. https://doi.org/10.1371/journal.pone.0214168

Mac Aodha, O., Gibb, R., Barlow, K. E., Browning, E., Firman, M., Freeman, R., Harder, B., Kinsey, L., Mead, G. R., Newson, S. E., Pandourski, I., Parsons, S., Russ, J., Szodoray-Paradi, A., Szodoray-Paradi, F., Tilova, E., Girolami, M., Brostow, G., & Jones, K. E. (2018). Bat detective—Deep learning tools for bat acoustic signal detection. *PLoS Computational Biology*, *14*, 1–19. https://doi.org/10.1371/journal.pcbi.1005995

Madhusudhana, S., Shiu, Y., Klinck, H., Fleishman, E., Liu, X., Nosal, E. M., Helble, T., Cholewiak, D., Gillespie, D., Širovic, A., & Roch, M. A. (2021). Improve automatic detection of animal call sequences with temporal context. *Journal of the Royal Society Interface*, *18*, 20210297. https://doi.org/10.1098/rsif.2021.0297

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in python. *Proceedings of the 14th Python in Science Conference*, vol. 8, pp. 18–25.

Mesaros, A., Diment, A., Elizalde, B., Heittola, T., Vincent, E., Raj, B., & Virtanen, T. (2019). Sound event detection in the DCASE 2017 challenge. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *27*, 992–1006. https://doi.org/10.1109/TASLP.2019.2907016

Mesaros, A., Heittola, T., & Virtanen, T. (2016). Metrics for polyphonic sound event detection. *Applied Sciences*, *6*, 162. https://doi.org/10.3390/app6060162

Nanni, L., Costa, Y. M., Aguiar, R. L., Mangolin, R. B., Brahnam, S., & Silla, C. N. (2020). Ensemble of convolutional neural networks to improve animal audio classification. *EURASIP Journal on Audio, Speech, and Music Processing*, *2020*, 8. https://doi.org/10.1186/s13636-020-00175-3

Nanni, L., Maguolo, G., & Paci, M. (2020). Data augmentation approaches for improving animal audio classification. *Ecological Informatics*, *57*, 101084. https://doi.org/10.1016/j.ecoinf.2020.101084

Neath, A. A., & Cavanaugh, J. E. (2012). The Bayesian information criterion: Background, derivation, and applications. *WIREs Computational Statistics*, *4*, 199–203. https://doi.org/10.1002/wics.199

Pandeya, Y. R., Kim, D., & Lee, J. (2018). Domestic cat sound classification using learned features from deep neural nets. *Applied Sciences (Switzerland)*, *8*, 1–17. https://doi.org/10.3390/app8101949

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*, 8024–8035.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2012) Scikit-learn: Machine learning in python. *CoRR*, abs/1201.0490.

Priyadarshani, N., Marsland, S., & Castro, I. (2018). Automated birdsong recognition in complex acoustic environments: A review. *Journal of Avian Biology*, *49*, jav.01447. https://doi.org/10.1111/jav.01447

Putland, R. L., Ranjard, L., Constantine, R., & Radford, C. A. (2018). A hidden Markov model approach to indicate Bryde's whale acoustics. *Ecological Indicators*, *84*, 479–487. https://doi.org/10.1016/j.ecolind.2017.09.025

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *arXiv preprint arXiv:200204803*.

Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings*, August 2015, pp. 4580–4584. https://doi.org/10.1109/ICASSP.2015.7178838.

Salamon, J., Bellol, J. P., Farnsworth, A., Kelling, S., & IEEE. (2017). Fusing shallow and deep learning for bioacoustic bird species classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 141–145. IEEE, New York. https://doi.org/10.1109/ICASSP.2017.7952134.

Salamon, J., Jacoby, C., & Bello, J. P. (2014). A dataset and taxonomy for urban sound research. *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, pp. 1041–1044. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2647868.2655045.

Sellers, P. H. (1974). On the theory and computation of evolutionary distances. *SIAM Journal on Applied Mathematics*, *26*, 787–793.

Shi, B., Bai, X., & Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*, 2298–2304. https://doi.org/10.1109/TPAMI.2016.2646371

Shiu, Y., Palmer, K. J., Roch, M. A., Fleishman, E., Liu, X., Nosal, E. M., Helble, T., Cholewiak, D., Gillespie, D., & Klinck, H. (2020). Deep neural networks for automated detection of marine mammal species. *Scientific Reports*, *10*, 1–12. https://doi.org/10.1038/s41598-020-57549-y

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*.

Somervuo, P., Harma, A., & Fagerlund, S. (2006). Parametric representations of bird sounds for automatic species recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*, 2252–2263. https://doi.org/10.1109/TASL.2006.872624

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Stevenson, B. C., Borchers, D. L., Altwegg, R., Swift, R. J., Gillespie, D. M., & Measey, G. J. (2015). A general framework for animal density estimation from acoustic detections across a fixed microphone array. *Methods in Ecology and Evolution*, *6*, 38–48. https://doi.org/10.1111/2041-210X.12291

Stiffler, L. L., Schroeder, K. M., Anderson, J. T., McRae, S. B., & Katzner, T. E. (2018). Quantitative acoustic differentiation of cryptic species illustrated with King and Clapper rails. *Ecology and Evolution*, *8*, 12821–12831. https://doi.org/10.1002/ece3.4711

Stowell, D., Benetos, E., & Gill, L. F. (2017). On-bird sound recordings: Automatic acoustic recognition of activities and contexts. *IEEE-ACM Transactions on Audio Speech and Language Processing*, *25*, 1193–1206. https://doi.org/10.1109/taslp.2017.2690565

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215

Usman, A. M., Ogundile, O. O., & Versfeld, D. J. (2020). Review of automatic detection and classification techniques for cetacean vocalization. *IEEE Access*, *8*, 105181–105206. https://doi.org/10.1109/ACCESS.2020.3000477

Wang, Y., Ye, J. & Borchers, D. L. (2022). Automated call detection for acoustic surveys with structured calls of varying length. Zenodo, https://doi.org/10.5281/zenodo.6461670.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., … Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Xu, W., Zhang, X., Yao, L., Xue, W., & Wei, B. (2020). A multi-view CNN-based acoustic classification system for automatic animal species identification. *Ad Hoc Networks*, *102*, 102115. https://doi.org/10.1016/j.adhoc.2020.102115

Zhang, X. X., & Li, Y. (2015). Adaptive energy detection for bird sound detection in complex environments. *Neurocomputing*, *155*, 108–116. https://doi.org/10.1016/j.neucom.2014.12.042

Zhong, M., LeBien, J., Campos-Cerqueira, M., Dodhia, R., Lavista Ferres, J., Velev, J. P., & Aide, T. M. (2020). Multispecies bioacoustic classification using transfer learning of deep convolutional neural networks with pseudo-labeling. *Applied Acoustics*, *166*, 107375. https://doi.org/10.1016/j.apacoust.2020.107375

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Wang, Y., Ye, J. & Borchers, D. L. (2022). Automated call detection for acoustic surveys with structured calls of varying length. *Methods in Ecology and Evolution*, *00*, 1–16. https://doi.org/10.1111/2041-210X.13873