Imperial College London

Creating a deep-learning automated audio detection system for Geoffroy's spider monkey, *Ateles geoffroyi*

Duncan Butler

August 2018

A thesis submitted for the partial fulfillment of the requirements for the degree of Master of Science at Imperial College London

Formatted in the journal style of Methods in Ecology and Evolution
Submitted for the MSc in Computational Methods in Ecology and Evolution (CMEE)

Word count: 5,896

Declaration

I declare all this work is my own. Portion of data used collected by me during fieldwork, majority collected and given use of by Jenna Griffiths, 11 *A. geoffroyi* calls taken from the Macaulay Library at the Cornell Lab of Ornithology (https://www.macaulaylibrary.org/)
A large amount of custom Python code written by me, used in this project and created for use in further training of the neural network available at https://github.com/dgabutler/spider-monkey-detector

Acknowledgements

I am massively grateful for the excellent guidance I have recieved from my supervisors, Dr James Rosindell and Jenna Griffiths, over the course of this thesis. I also owe a huge debt of thanks to my family and friends - I love you all, and me finishing this was as much your doing as it was mine

1 Abstract

1. Combined with recent innovations in their applicability to small datasets, convolutional neural networks (a powerful machine learning algorithm) hold great promise for use in ecological monitoring. This is particularly the case in passive acoustic monitoring - a method capable of collecting a large amount of data very efficiently - and in highly biodiverse regions, where factors such as high noise levels had previously significantly limited automated data processing. However, it is a challenge to create effective automated systems, with many factors that can be varied that influence performance. Nonetheless, state-of-the-art performances are possible if a suitable combination of these factors can be achieved.

2. In this project, as a case study and an opportunity to investigate some of these factors, I developed an automated basic detection system for Geoffroys spider monkey, *Ateles geoffroyi*. The factors investigated were several methods of data augmentation (artificially generating new samples) and data preprocessing ('denoising' and standardising) previously shown to increase performance of CNN classifiers for similar problems

3. The main findings were that the factors tested did not enable a CNN to sufficiently learn to recognise the signal of interest. I also discovered that a common measure of machine learning success can cause artefacts in results when applied to small datasets.

4. I highlight key elements likely to be limiting the effectiveness of the system at present and identify possible methodologies to increase performance in further work (providing a large amount of custom-written code to enable further training).

2 Introduction

Threats to biodiversity in highly biodiverse regions such as rainforests are increasing (Alroy 2017). Understanding the full effects of these requires frequent monitoring on large enough landscape scales (Underwood et al., 2005; Porter et al., 2009) and over a sufficiently long period of time (Porter et al., 2005); however, at present there is a lack of sufficient effective monitoring systems (Proença et al., 2017). It is imperative to develop cost-effective monitoring techniques with the potential to be implemented at large landscape scales and over long time periods in crucially important highly biodiverse regions such as rainforests. Due to technological and theoretical advancements, one emerging approach is the combination of innovations in machine learning with passive acoustic monitoring.

Passive acoustic monitoring (hereafter PAM) is the process of collecting acoustic data in the field using sensors such as microphones, to then analyse at a later point. The acoustic data collected can be used to answer a number of questions relating to the ecology and distribution of species (Browning et al., 2017), which can for example be used in the design (location and habitat type) of protected areas (Rayment et al., 2009).

It holds promise as an efficient surveying tool in hyper-biodiverse regions for a number of reasons. Acoustic monitoring approaches can reduce or eliminate biases inherent in other survey methods, including detection bias (as initial data collection is independent of observer skill level (Klingbeil and Willig, 2015)), temporal bias (which has shown in point count studies to result in missed behaviours and underestimated population sizes (Bridges and Dorcas, 2000), and biases caused by human disturbance (Alldredge et al., 2007). Meeting the requirements of more efficient surveying techniques, the area under surveyance can be increased for a comparatively lessened increase in cost, which can allow ecological questions to be tested on large scales (Wrege et al., 2017). Recent significant reductions in cost of surveying devices (from hundreds of pounds to as little as £40 per unit for the recently developed AudioMoth devices (Hill et al., 2018)), as well as improvements to their memory capacity and factors such as weatherproofing (Fanioudakis and Potamitis, 2017), have massively increased the potential of PAM analyses to generate a huge quantity of data. This can be contributed to global repositories of biodiversity information, increasing the potential for wide-scale monitoring and modelling (Honrado et al., 2016). Furthermore, a key benefit is that the data collected forms a permanent record: survey analyses are able to repeatable, different ecological questions can be investigated using the same data (Newson et al., 2017), and factors such as changes of community composition can be looked at (Rogers et al., 2013). PAM techniques will significantly increase the ability to monitor otherwise unobservable cryptic species and behaviours (Wrege et al., 2017). Where suitable, another important application could be in more viably evaluating the effectiveness of conservation actions (Wrege et al., 2017), a critical stage which is too often overlooked in conservation science (Ferraro and Pattanayak, 2006; Legg and Nagy, 2006).

Further reasons why PAM could be particularly beneficial in regions such as rainforests include that acoustic monitoring approaches are much less seasonally restricted (Shonfield and Bayne, 2017) (important in tropical biomes which often have prohibitive seasonal weather), and that it enables surveying of areas where direct observation of species may not be feasible. Additionally, the general advantage of associated reduction in observer effort when using PAM approaches (Digby et al., 2013) are accentuated as survey sites in hyper-biodiverse areas are often remote and potentially difficult to access, allowing for data to be collected over longer time frames more easily. However, while these strengths all enable data to be collected very efficiently, a current key limiting factor is simply being able to process the 'big-data' created.

Although there is broad potential in applying PAM approaches in highly biodiverse regions, managing and analysying the terrabytes of data that these investigations can collect has been a significant problem (Villanueva-Rivera and Pijanowski, 2012; Shonfield and Bayne, 2017). Extraction of the sounds of interest requires an expert to spend a large amount of time listening to the recordings, and rarely quantified sources of bias can be introduced at this stage (Digby et al., 2013). As a result of the processing time required, it is common that only a fraction of data collected is able to be used (Kobayasi and Riquimaroux, 2012). There has therefore been a strong incentive to incorporate techniques from the field of machine learning (hereafter ML), in which algorithms can be designed that are capable of automating the processing element of the task.

Despite there being a documented lack of communication between the two fields of research (Thessen, 2016), traditional ML techniques such as support vector machines, random forests, and naive bayes have been applied to bioacoustic datasets for wide a variety of taxa (see comprehensive recent review by Knight et al. (2017)). These algorithms compare key hand-designed features of input audio data - temporal (e.g. duration) or spectral (e.g. peak frequency) - with those learned from a dataset of labelled training examples. However, while excellent results have been reported using these traditional methods (although classification algorithms coming even close to expert observer accuracy rates are not common (Ovaskainen et al., 2018)) few attempts have been made to combine automated detection systems with PAM data in hyperbiodiverse regions such as rainforests (Browning et al., 2017). This is a bias that also extends to availability of suitable training datasets, and these combined have been described as being a major gap in the field at present (Browning et al., 2017).

This lack of research effort is due to key difficulties of automated detection and classification in these regions. A primary challenge is the generally increased levels of noise, obscuring signals of interest (Browning et al., 2017). Variability in background environmental noise level has also previously limited the effectiveness of fully automated systems (Heinicke et al., 2015), and rainforests are known to have both high variation and high general baseline levels of noise (Waser and Waser, 1977), Traditional machine learning techniques can be very affected by noisy weather conditions, with recordings containing wind and rain often having to be discarded (Stowell et al., 2018). The task of detecting signals of interest in highly biodiverse regions is made more difficult by the increased levels of similarity in sounds produced by different species in these more complex soundscapes (Zamora-Gutierrez et al., 2016).

However, the emergence in recent years of ML methods that are much more resilient to noisy input data, and that are capable of learning optimal discriminative features automatically, has significantly

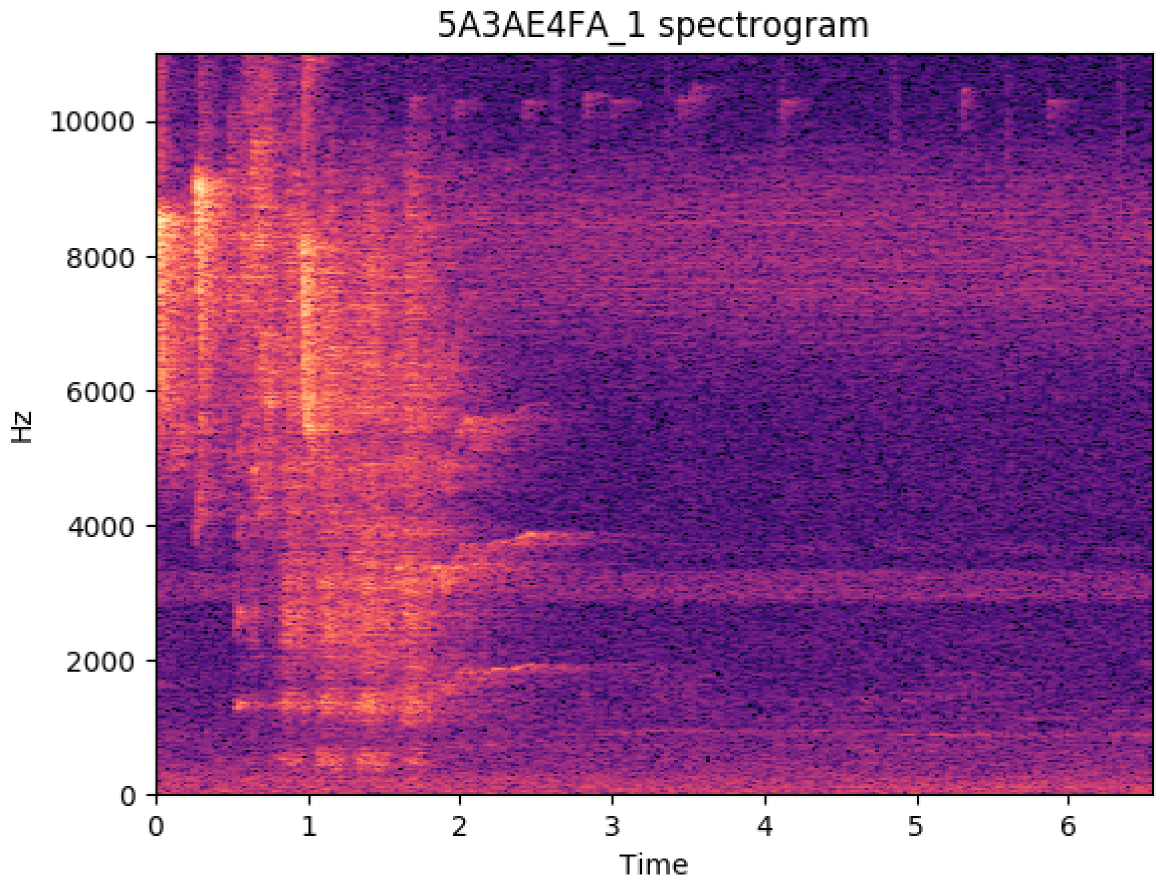


Figure 1: Example of a spectrogram, a visual representation of an audio signal. This is a Geoffroy's spider monkey call known as a 'whinny', which was the signal of interest in the case study I present in this paper

increased the potential of effective monitoring in these regions (Browning et al., 2017). An example of this is the field of deep learning. There is a clear trend in the literature demonstrating its capability to produce state-of-the-art results in general audio detection problems ((Joly et al., 2016; Knight et al., 2017; Kahl et al., 2017), and it has been proposed to be a promising method in particularly noisy environments (Browning et al., 2017). Although some of the most recent innovations are to feed raw audio waveforms into deep learning detectors (Dai et al., 2017), visual representations of audio samples (known as spectrograms, see Figure 1) are most commonly used as inputs to these pattern-recognising algorithms; following a period of training using positive (signal-containing) and negative clips, deep learning algorithms such as convolutional neural networks (hereafter CNNs) are then able to learn to recognise complex patterns, even if these contain variation or are partly masked by noise.

Further innovations have enabled these powerful techniques to be successfully applied to small datasets, circumventing the previous limitation of deep learning approaches of requiring very large amounts of data in order to achieve good results (Kiskin et al., 2017; Salamon and Bello, 2017). One such innovation is data augmentation, the process of applying transformations to data to artificially generate new samples with which to supplement the training of deep learning detectors. This process has the added advantage of increasing the generalisability of automated systems, as augmentations can be applied that mimic a wide variety of conditions possible in real-world conditions (such as overlapping of the signal with a prominent noise) that may well be absent from the training data.

These techniques can be coupled with data preprocessing steps to further increase the possible performance of automated detection systems in highly biodiverse regions, an example of which is the 'denoising' of audio samples. While it is a complex task to separate signal from noise (Ovaskainen et al., 2018), various methods have been developed to to increase signal-to-noise ratios of samples, and this has been shown to boost the performance of automated systems (Stowell et al., 2016). Another often-applied step is standardisation of all input samples, such as bounding input values between 0 and 1, which enables more efficient training of the networks and may help to correct for the problem of different recording sites within a study area having general amplitude levels of varying intensity. Overall, explorations of these emerging techniques are fundamental in leveraging machine learning techniques in highly biodiverse regions.

As a case study, in this project I will develop a basic automated detection system for the endangered neotropical primate Geoffroys spider monkey, *Ateles geoffroyi*, in which a trained deep learning model (a CNN) will be applied to continuous rainforest audio recordings. This species is well-suited for this analysis as they are heavily reliant on acoustic communication, as a result of being almost entirely arboreal, frugivorous (with patchily-distributed food), and living in complex fission-fusion societies splitting into subgroups to forage (Ramos-Fernández, 2008). I will train this detector using preliminary data from a wider project for which this system is intended to be used as a surveying tool, to assess the current distribution and habitat preferences of *A. geoffroyi* over the large scale (2500 km^2) Osa peninsula of Costa Rica in order to build wildlife corridors to connect currently isolated populations.

This problem offers the opportunity to investigate how varying elements of CNN design - as they can consist of a number of different components, with many alterable parameters - and training - the process by which they learn patterns in data - affect their ability to act as generalisable monitoring tools in high-noise environments. I will experiment with two audio preprocessing techniques - denoising and standardising. Due to the very small current size of the training dataset, I will also apply data augmentation, using several methods previously used on similarly small datasets. Finally I will experiment with a number of combinations of tunable elements of the CNNs - known as hyperparameters - to optimise the current system based on the findings of the most effective combination of preprocessing techniques. As more data will be collected as part of the wider project, I will test the effect of increasing data on the performance of the system (termed a 'learning curve') to further assess whether current performance is limited by data availability or architecture design.

3 Methods

3.1 Data: collection and labelling

The original data was continuous recordings of rainforest sounds on the Osa Peninsula, a portion of which was collected in December 2017, which was then supplemented by data I collected during one-month of fieldwork in May 2018. We used AudioMoth recording devices (Hill et al., 2018), which create minute-long '.wav' files named with a hexadecimal code representing the time and date of recording, recorded at 48 kHz. We made the recordings by fastening the devices to trees for periods of approximately three days, orienting the omnidirectional microphone upwards and angled into unsheltered areas of the forest so as to give the best chance of recording clear spider monkey calls. The possibility of water damage influenced how they were placed (for example slightly sheltered by vegetation); however, some water damage did cause some data loss. Nonetheless, over the two recording periods (plus a further one since) we collected a total of approximately 2000 hours of data.

To create the dataset of 'positive training examples (clips containing a spider monkey whinny), a primatologist with four years of experience listening to spider monkey calls listened to 191 hours of the recordings, separating out minute-long clips containing the signal of interest. She then created label files in the software Praat (Boersma and Weenink, n.d.) containing the start and end times of periods with and without the call. Using custom functions written in Python, I clipped the audio files into three second 'positive' sections containing a call. As calls were approximately 1 second long on average, with standard deviation of 0.3 seconds and the longest recorded being 2.1 seconds, I decided that a three second window was suitable. Crump and Houlahan (2017) reported that it was most beneficial to train their CNN detector using positives from as many different locations within the study region as possible, suggesting this was due to increasing the number of unique individuals recorded. Due to a lot of the data labelling being done before the start of the project, most of the positives (67/124, 54%) used to train the network were from only one location. However, a portion of positives added in the later stages were from different locations (31% and 6% from two sites recorded during the data collection period, and a further 11 calls, 9%, recorded in the same region but taken from The Macaulay Library at the Cornell Lab of Ornithology). As this detector is intended to only be used in one region, I only used training from individuals in that region as recommended by Knight et al. (2017) (forgoing the opportunity to add additional positive clips from *A. geoffroyi* available).

I created the 'negative' training examples (three second clips known to not contain the signal of interest) in a three ways: (1) random sampling from call-containing minute-long clips in regions of the clips known to not contain calls; (2) carrying out a process of 'hard-negative mining' (as done by Mac Aodha et al. (2018)) in which an early-stage trained version of the detector was ran on minute-long clips that have labelled call and non-call regions, separating any three-second sections classified as being positive (but known to be negative), and; (3) running early-stage versions of the detector on entire folders (a folder of files was all data recorded from one site in one three-day recording period), and the same expert that originally labelled the calls listened to a large number of the positively-classified clips, separating out any false positives (clips not contain the signal of interest). I applied the hard-negative mining technique as Mac Aodha et al. (2018) reported significant improvements as a result of this training on more challenging examples.

In total, the original dataset with which to train the network consisted of only 124 positive clips and an equal number of negative examples (classes balanced as done by Mac Aodha et al. (2018) and Kiskin et al. (2017) for similar neural network binary detection problems). Where possible, for a given location I balanced the number of negatives with the number of positives so as to not introduce any biases by over-representing certain locations (which may have had different levels/combinations of background noise). This was not possible for the 11 calls that were not recorded by us, and so I balanced these few with further negatives from one of our recording sites. For locations with both hard-negative mined negatives and randomly sampled negatives, I added an equal ratio of both.

3.2 Data: preprocessing, augmentation

I converted all raw audio clips to spectrograms using a fast Fourier transform - a mathematical process which decomposes the audio signal into the separate frequencies that combined to form the signal. I used the Python package Librosa (McFee et al., 2018) for this. Specifically, I created mel-frequency spectrograms, in which the frequency bins are scaled logarithmically, thereby placing lesser importance on distinguishing between higher frequencies. This stage, mimicking how human ears process sounds of differing frequencies, has been shown to be a successful transformation for data reductionality (reducing training time of neural nets), and is commonly used in state-of-the-art deep learning audio detection systems (Stowell et al., 2018). Following this stage, the input was a 128 x 282 matrix (128 frequency bins, over 282 time steps).

To denoise the spectrograms, I chose the denoising function of Aide et al. (2013) - also used in the competition-winning binary detection system of Kahl et al. (2017). This function works by subtracting the mean amplitude of each frequency bin from all values in that bin, keeping only particularly loud signals present in the spectrogram. To standardise the inputs (bounding them between 0 and 1), I divided all values (amplitudes) in each input spectrogram by the largest value in the spectrogram.

I implemented several data augmentation methods used by a number of teams working on similar problems, that aimed to increase the robustness of a detector against levels of background noise as well as boosting the system generalisability e.g. Sprengel et al. (2016); Kahl et al. (2017). These augmentations were: (1) adding a varying amount slight distortion (Gaussian noise) to the mel-spectrograms once generated, and; (2) blending signal-containing files with and non-signal-containing files containing a prominent sound, such as a calling howler monkey or a loud bird. To do the latter, I selected a number of three-second 'noise' clips, augmentation function would randomly select from these, add together mel-frequency spectrograms of the 'signal' and 'noise' clips, and renormalise to ensure the background noise levels had not been artificially doubled.

I also developed a random crop augmentation function, in which the positive signal is repositioned within the three-second sample with a high probability that it is at least part-way cut off (retaining a minimum of 20% of the call within the window). This was to ensure the that network was trained on calls that had been interrupted part-way through, an important stage as the full system splits minute-long files into non-overlapping three-second clips for testing, increasing the possibility that any calls present will span separate input clips.

3.3 Detector: design, optimisation and training

As it is very challenging to entirely build an effective CNN, I followed the general practise of using the general architecture of state-of-the-art network designed for use in a similar detection problem (sound classification using small datasets with data augmentation) (Salamon and Bello, 2017) (see Table 1). There are certain network hyperparameters that commonly vary between papers tackling similar detection problems, which can influence the overall model performance and level of generalisability. I varied the dropout percentage, and number of neurons in the fully-connected layers at the end of the network. To test optimum combinations of the hyperparameters under investigation I implemented an approach called 'grid-searching', which trains CNNs on all possible combinations of the choices for the hyperparameters to determine the optimal values on a problem-specific basis. I chose to run this optimisation stage using the fully augmented dataset, to minimise the likelihood that the optimal hyperparameters were only representing significant attempts to combat overfitting (a common issue with small datasets in which the network begins to learn the exact patterns of the data rather than the general trends, reducing its ability to generalise to unseen data).

I trained the neural net for 40 epochs (a measure of machine learning training time, where one epoch is the number of training steps required for the network to have trained on every sample in the training set). This was because preliminary investigations showed that this was enough time to record the optimal performance of the models before overfitting occurred.

Table 1: General CNN architecture used, taken from Salamon and Bello (2017), with the results of optimum hyperparameters for the network indicated in bold. CNN trained using fully augmented samples.

Conv1, 24 5x5 kernels, Stride (1,1)
$\Downarrow$
Max Pooling, Size (4x2)
$\Downarrow$
Activation: relu layer
$\Downarrow$
Conv2, 48 5x5 kernels, Stride (1,1), Valid Padding
$\Downarrow$
Max Pooling, Size (4x2)
$\Downarrow$
Activation: relu layer
$\Downarrow$
Flatten
$\Downarrow$
Dropout, p=**0.836**
$\Downarrow$
Dense, 64/**128** neurons
$\Downarrow$
Activation: relu layer
$\Downarrow$
Dropout, p=**0.953**
$\Downarrow$
Sigmoid Output

3.4 Detector: evaluation

Machine learning classification algorithms can be evaluated using different metrics, which assess performance taking into account different combinations of the four classification possibilities: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These are obtained by training the algorithm on a percentage of the overall dataset, with general practice being to use 80%-90% depending on total dataset size, and then using the model to make predictions for each of the samples in the remainder of the dataset (termed the validation set). In choosing which metrics to report when comparing the models trained under different conditions, I followed the best-practise recommendations of Knight et al. (2017). These are to report 'recall' (number of calls detected, TP, as a proportion of total calls present in the validation dataset, TP + FN), 'precision' (ratio of calls detected, TP, to total clips classified as being positives, TP + FP), and 'F1 score' (the harmonic mean of recall and precision). A high recall value would represent a situation in which a lot of calls in the validation dataset were correctly detected; however, as a detector that learns to classify every sample as being positive would give a perfect recall score, combining recall score with precision score (which penalises for number of false positives) means that the F1 score is a useful way to summarise overall performance of the detector.

For each preprocessing manipulation under investigation, I used the stratified 10-fold cross-validation function of the machine learning Python package scikit-learn, as this method is regarded to be the most comprehensive method of evaluating the performance of a neural net. The complete dataset is split into 10 'folds' (maintaining the proportion of positives and negatives in each fold). Ten CNNs are then created, with each being trained on a different withheld portion of the overall dataset, allowing average metric values with an indication of measure of spread to be obtained. As the metric value for the validation dataset could possibly increase but then decrease over training time if overfitting occured, the metric value I took to summarise maximum obtainable performance per model training run was the maximum recorded over the training period (a common ML practice e.g. as used by (Norouzzadeh et al., 2018)).

To evaluate the effects of the data augmentation methods, rather than applying the the 10-fold cross-validation function of scikit-learn, I created functions that randomly portioned the dataset into seperate training and validation sets with a 90%/10% split, prior to then augmenting each individually within the separate sets. This was to ensure the complete independence of the training and validation data, a crucial step which can otherwise inflate evaluator performance.

3.5 Overall system design and functionality

The overall detection system iterates over folder of one-minute long files (as produced by the AudioMoth recording devices), processing each for the presence of the signal of interest. For each file, it splits it into 20 three-second clips, which are sequentially inputted into the trained CNN. Each resulting activation value of the last (output) layer of the network (a value between 0 and 1) is checked, and if this value is above a threshold activation value (e.g. 0.5) - specified by the user at runtime - the clip is considered to contain the signal. The metrics reported for the CNN performance were tested with a threshold of 0.5, but I have included the option to alter the threshold when running the system to allow for the altering of the sensitivity of the detector - increasing the threshold would decrease the number of false positives, but also increase the number of false negatives (calls that are more difficult to detect). When running the system, the user can select which of the preprocessing techniques (denoising, standardising, or the combination of both) they would like to apply to the input data.

I programmed the system to give two outputs. The first is a folder containing all detected-positive three second clips (informatively labelled with the original file name of the 60-second clip, the time location within the clip they came from i.e. the start and end of three-second interval, a number representing which of the total number of detected clips from their file they were, and the activation value multiplied by 100 acting as a proxy of confidence). The second output is a summary CSV file of all detected clips, containing file name, approximate position of detected clip in file (secs), the time and date of recording of the original file, and the confidence of the CNN's classification (again, calculated using activation value of output layer for each clip).

4 Results

The learning curve investigating detector performance at progressively increasing training dataset sizes (see Figure 2) showed that, for the very limited sample sizes at present, an increase of number of positives (with a corresponding increase in negatives) showed a barely discernible, and likely non-significant, increase in the performance of the detector, as measured by F1 score.

The steady increase in performance on the training dataset (e.g. see Figure 5a) demonstrates that, over the period trained, the network was able to learn, but the performance on the test/validation dataset staying at around 50% (with a potential slight downward trend) implies that the CNN was performing no better than random, overfitting to the training dataset rather than learning on the discriminative aspects of the signal of interest.

The pattern depicted by Figure 5a is that, when comparing maximum obtainable performance in any metric of datasets that had been augmented, there was a highly significant drop in their performance on the respective validation sets (tested on a withheld 10% portion of each dataset). However, on discovering that the method I used to evaluate seemed to be introducing artefacts when applied to small dataset sizes (see Discussion for expansion on this point), I decided to separately visualise effects of preprocessing techniques for the original dataset only ($n = 248$), and effect of preprocessing and augmentation on larger datasets ($n = 744$ and $n = 2232$), considering those to be potentially more valid comparisons. When applied to the small original dataset, there was no significant performance difference between preprocessing techniques (Figure 4).

The results of the grid-search investigation for optimum configurations of the hyperparameters tested (see Table 1) showed that the values of dropout were $p = 0.836$ in the first dropout layer and $p = 0.953$ in the second dropout layer, with 128 neurons leading to highest performance results (compared to $p = 0.5$ dropout and 64 neurons in the original network architecture).

The time for the system as a whole to process one minute-long file was approximately 1.553 seconds,

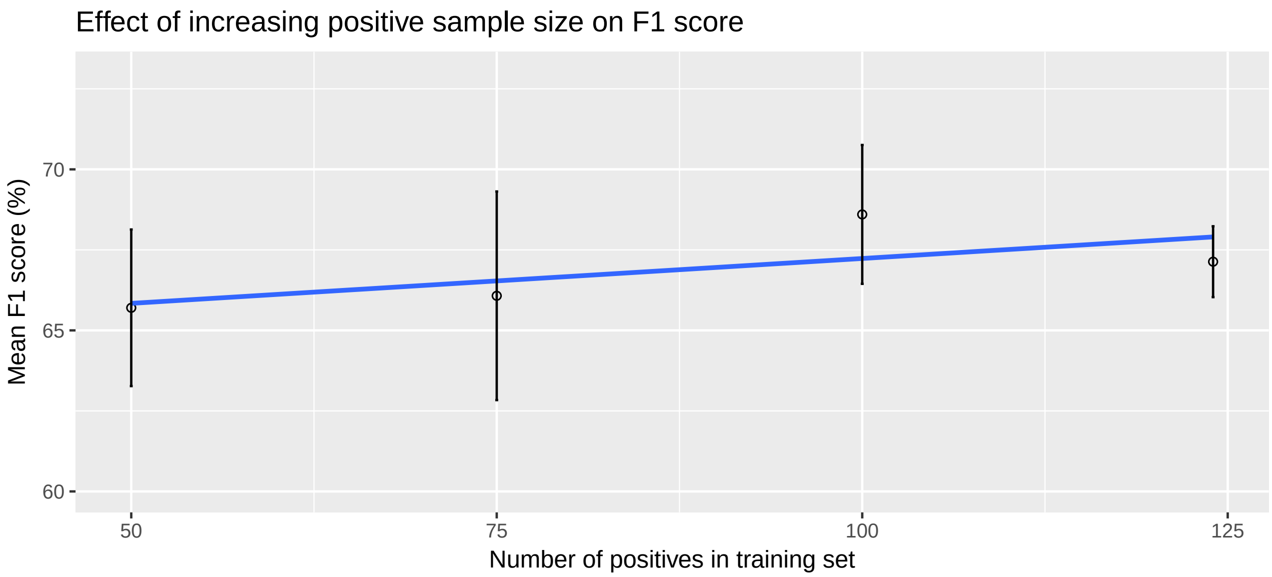


Figure 2: Learning curve, assessing whether potential performance of detector is likely to improve with further data. Metric shown is F1 score, the harmonic mean of precision and recall and therefore a good single measure of detector performance. Four increments of randomly sampled positives (n= 50, 75, 100, and all 124) with a balanced number of randomly sampled negatives were used to train CNNs. The values plotted were the means of ten repetitions, taking maximum metric performance, of training CNNs for 40 epochs. Bars show standard error

and so to process a folder of all files collected over the approximate recording period from one site (72 hours) took approximately 1.9 hours.

5 Discussion

The main findings of this study were that the existing methods for tackling the problems of deep learning with small datasets, and in noisy environments, that I combined were not as effective as I had hoped, with more work needed to develop a sufficiently accurate system. A key discovery was that care should be taken when applying standard ML measures of success (taking maximum performance on validation set during training period) on small sample sizes, as closer inspection of diagnostic plots charting the performance of CNN over training time show that the metric values can fluctuate quite significantly. A reason that this might be is that some samples may be easier to classify than others (possibly due to different signal-to-noise ratios as a result of factors such as differing levels of background noise or proximity to recording device), and, as network weights are are updated when the network is trained on the number of values in the batch size, it is possible that - for a smaller dataset - there is a higher likelihood that a batch may be made up of these more easily classifiable samples. I observed that these fluctuations narrow with more data (compare (a) with (b) in 5) which lends support to this hypothesis.

Figure 4 shows that, for the original dataset of 248 samples, there was no significant difference in

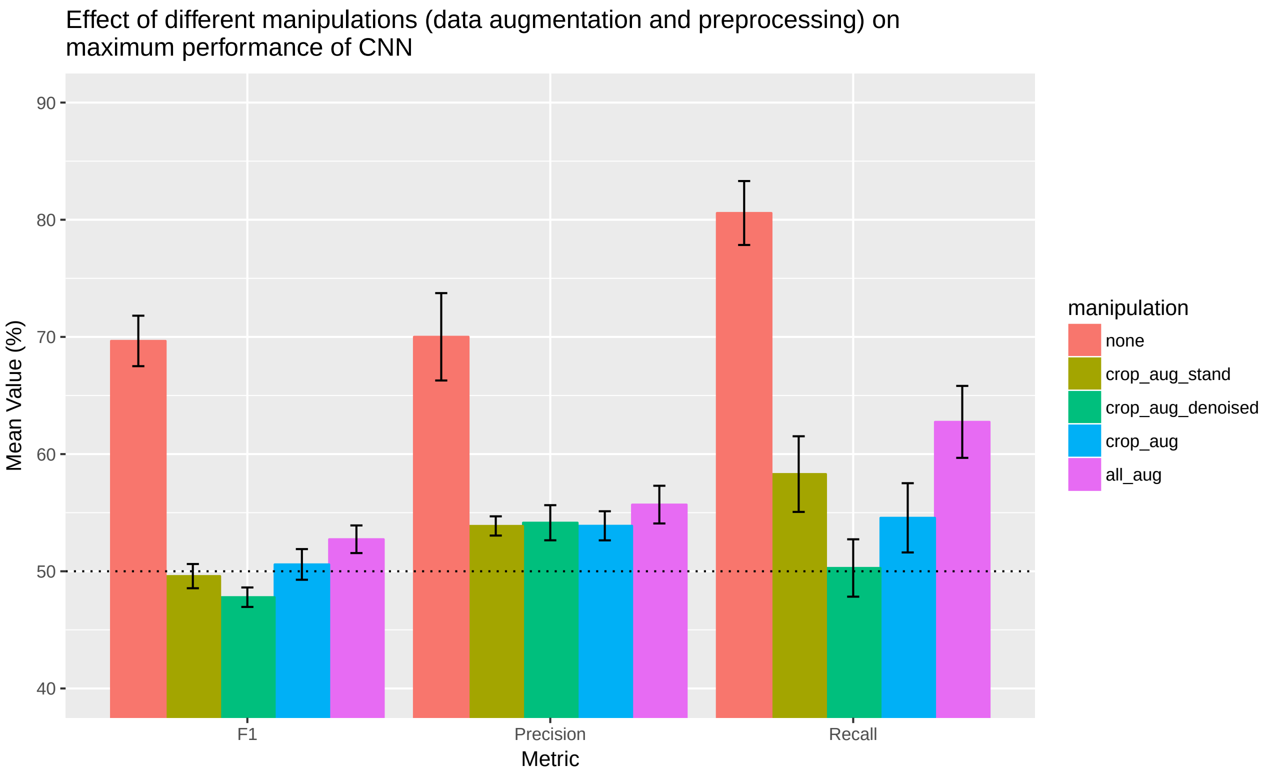


Figure 3: Change in classification performance - measured using three different metrics - of convolutional neural networks (CNNs) trained on augmented data to which I applied different data preprocessing techniques and augmentation methods. Manipulations were no preprocessing/augmenting, standardised crop-augmented data, denoised crop-augmented data, crop-augmented data only, and applying all three augmentation methods (cropping, Gaussian-noise, and file-blending). Metric values are precision (proportion of true positives to total positives), recall (proportion of true positives to true positives plus false negatives) and F1 score (harmonic mean of precision and recall). Bar height represents mean metric value of ten repetitions of training for a given condition with training and validation set reallocated randomly, error bars show standard error. Dashed line at y = 50% representing expected mean if CNN was classifying at random, as positives and negatives balanced in all datasets. Dataset sizes: dataset with no manipulations = 248, crop-augmented datasets n = 744, all-augmented dataset n = 2232

the preprocessing techniques applied. The lack of effect of the denoising function tested is possibly due to a number of the samples being used to train the network having a particularly low signal-to-noise ratio, and so a general increase in distinction of the prominent sounds in the clip will likely have been insufficient to highlight them. The possible lack of effect seen in standardising is also likely due to insufficiency of samples in which the pattern could be seen for the noise of the dataset, leading to an overall inability for the networks to generalise to unseen data.

Contrary to the expected effect, the augmentation methods I applied to the data significantly decreased the performance of the detector as compared to training the detector on the original data alone. While this is very likely again due to a lack of samples containing a clear enough signal (as augmenting may need to be complemented by more sophisticated denoising approaches), the gap in performance was almost certainly exacerbated by the ramifications of the measure of success used

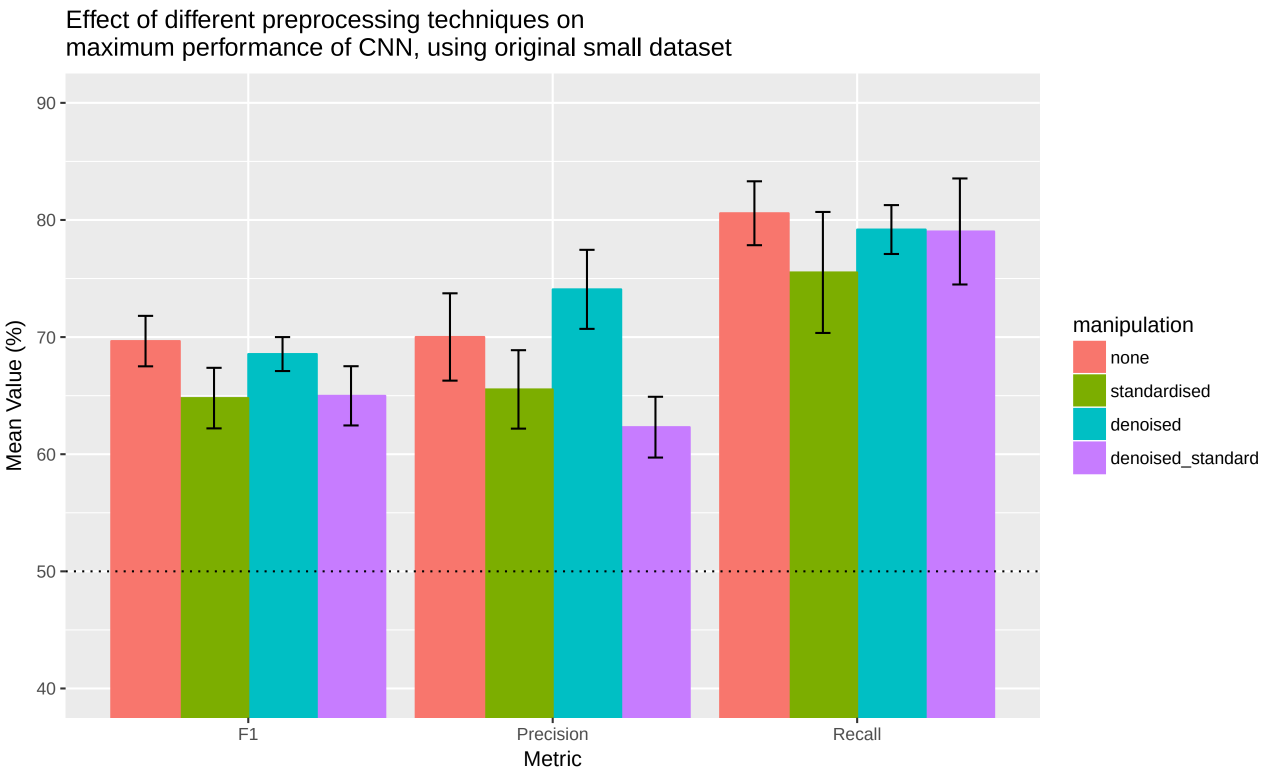
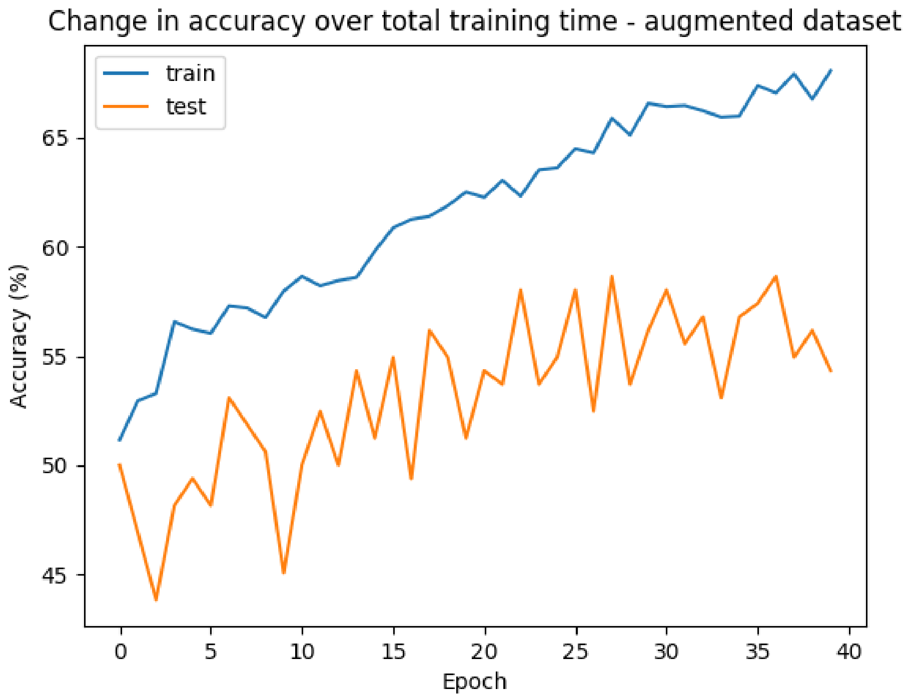


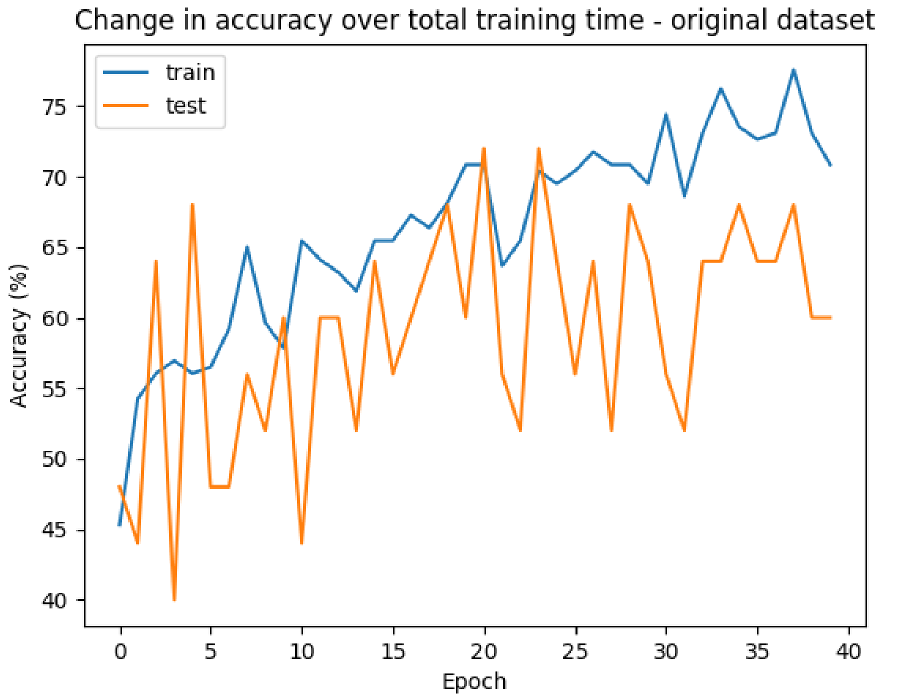
Figure 4: Change in classification performance - measured using three different metrics - of convolutional neural networks (CNNs) trained on unaugmented data to which I applied different data preprocessing techniques (no preprocessing, standardising inputs, 'denoising' inputs, and the combination of standardising and denoising). Metric values are precision (proportion of true positives to total positives), recall (proportion of true positives to true positives plus false negatives) and F1 score (harmonic mean of precision and recall). Bar height represents mean metric value of ten repetitions of training for a given condition, error bars show standard error. Dashed line at y = 50% representing expected mean if CNN was classifying at random, as positives and negatives balanced in all datasets. All preprocessing techniques applied to

being inappropriate on datasets of this size. Investigating the effects of preprocessing and augmenting on larger datasets (in which I observed that the problem of fluctuating metric values was lessened), I found that performing several augmentation methods to the dataset (as done by Kahl et al. (2017) and Salamon and Bello (2017)) resulted in better recall performance than when only one augmentation method was applied. Future work may wish to investigate this, taking a different approach to measure classification accuracy (such as training the network for a given period and evaluating the performance only at the end of that duration, on a withheld validation dataset). Although the CNN was able to improve its performance i.e. learn, on the training sets, this translated to poor detection performance on the validation sets (see Figure 5), implying that the common small dataset problem of overfitting was happening, and the results of the optimisation of the dropout layer proportion support that, for the augmented dataset tested, overfitting certainly occurred, being so high (0.8 and 0.9, with 0.5 being a typically suggested value for dropout proportion).

There were several limitations within my investigation. The typical proportion of overall data



(a) Fully augmented dataset: n = 2232.



(b) Original dataset: n = 248.

Figure 5: Change in binary accuracy (percentage of positives and negatives correctly classified) over training time; comparing trends in two datasets of differing sizes. CNNs were trained on training set - and upward trends of performance values show that they learned as a result. The sample size of (a) was increased using three methods of data augmentation (random-cropping, Gaussian-noise, and noise-file-blending). Both contain a balanced number of positive and negative samples. Dataset sizes (a) n = 2232, and (b) n = 248. Note the significantly larger oscillations in the classification performance of the small dataset.

used as training data varies between ML investigations, typically within the region of 80-90%. In keeping with the process of 10-fold cross-validation commonly used to assess ML algorithms, I tested all manipulations with a training dataset of 90%. However, this will have likely exacerbated the artefacts added by the method I used to measure maximum obtainable performance over the training period. Due to having very little data, I was unable to withhold an entirely separate dataset as a fixed task upon which all modifications under investigation could be tested, which would have allowed for a more direct comparison of the effectiveness of each manipulation. As more data is collected in the future of the wider project, this may become possible to better understand the different approaches.

I have identified a number of possibilities for further work to increase the potential of the system. More sophisticated denoising approaches should be investigated, such as that used by Versteegh et al. (2016), and as more data is collected, to select samples with a greater chance of informatively training the network (i.e. with a sufficient signal-to-noise ratio), functions could be written that automatically assess signal-to-noise, such as those used by Kahl et al. (2017). I applied three augmentation methods in my analysis; however, there are further functions, such as randomly incorporating a small degree of pitch-shifting of the sample, while still maintaining biological validity, that have been shown to have beneficial effects on detector performance with small datasets (Kahl et al., 2017). Although unexplored within the scope of this project, I predict that the system would be significantly improved via the incorporation of a process known as transfer learning. Transfer learning is also designed as a solution to data-limited problems, the first few layers of deep learning models trained on huge datasets tackling a similar problem (in this case detecting patterns in spectrograms of audio) can be taken as they are - leveraging learned overall general signal detecting abilities - but the last few layers (in which the abstract patterns of one specific sound versus another) can be retrained on limited samples containing the signal of interest. This was applied by Strout et al. (2017) for a similar detection problem and was shown to increase performance. It may be beneficial to follow the method of Mac Aodha et al. (2018) in bounding the input data to only the known frequency range of the call of interest, which would act to reduce the input size (and therefore training time of the CNNs) as well as potentially increasing the ability of the detector to learn true pattern in the data.

5.1 Conclusion

In this work, I developed the overall framework of an automated detection system for the calls of *A. geoffroyi*. I selected a CNN as the ML algorithm for use in the system as, based on the literature, it is the most powerful method for larger sample sizes and most robust against higher levels of background noise; both were suitable as the intended use for the system is as part of a wider monitoring project in a challenging soundscape for which significantly more data will be collected. However, as it was currently a problem of limited data, it presented an opportunity to investigate the recent innovation of data augmentation, shown by other works to allow for deep learning to be effective on small datasets. While within the timeframe of this project these methods did not lead to a system that was able to detect the calls with any significant accuracy, the system as a whole can be continued to be trained as more data will be collected (with an aim of labelling increased portions of data from different locations), using the large amount of custom written code for processing and retraining. Once a threshold value of 500 calls is reached, architecture effectiveness will be reassessed, exploring possibilities proposed as a result of this investigation. Generally, further research into the combination of deep learning and PAM in challenging soundscapes such as rainforests is paramount for the development of essential biodiversity monitoring tools.

References

Aide, T. M., Corrada-Bravo, C., Campos-Cerqueira, M., Milan, C., Vega, G. and Alvarez, R. (2013), 'Real-time bioacoustics monitoring and automated species identification', *PeerJ* **1**, e103.

Alldredge, M. W., Pollock, K. H., Simons, T. R., Collazo, J. A. and Shriner, S. A. (2007), 'Time-of-detection method for estimating abundance from point-count surveys', *The Auk* **124**(2), 653–664.

Boersma, P. and Weenink, D. (n.d.), 'Praat: doing phonetics by computer'.
URL: *http://www.praat.org/*

Bridges, A. S. and Dorcas, M. E. (2000), 'Temporal variation in anuran calling behavior: implications for surveys and monitoring programs', *Copeia* **2000**(2), 587–592.

Browning, E., Gibb, R., Glover-Kapfer, P. and Jones, K. E. (2017), 'Passive acoustic monitoring in ecology and conservation'.

Brownlee, J. (n.d.), 'Statistical significance tests for comparing machine learning algorithms', https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/. Accessed: 2018-08-25.

Crump, P. S. and Houlahan, J. (2017), 'Designing better frog call recognition models', *Ecology and evolution* **7**(9), 3087–3099.

Dai, W., Dai, C., Qu, S., Li, J. and Das, S. (2017), Very deep convolutional neural networks for raw waveforms, *in* ‘Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on’, IEEE, pp. 421–425.

Digby, A., Towsey, M., Bell, B. D. and Teal, P. D. (2013), ‘A practical comparison of manual and autonomous methods for acoustic monitoring’, *Methods in Ecology and Evolution* **4**(7), 675–683.

Fanioudakis, L. and Potamitis, I. (2017), ‘Deep networks tag the location of bird vocalisations on audio spectrograms’, *arXiv preprint arXiv:1711.04347* .

Ferraro, P. J. and Pattanayak, S. K. (2006), ‘Money for nothing? a call for empirical evaluation of biodiversity conservation investments’, *PLoS biology* **4**(4), e105.

Heinicke, S., Kalan, A. K., Wagner, O. J., Mundry, R., Lukashevich, H. and Kühl, H. S. (2015), ‘Assessing the performance of a semi-automated acoustic monitoring system for primates’, *Methods in Ecology and Evolution* **6**(7), 753–763.

Hill, A. P., Prince, P., Piña Covarrubias, E., Doncaster, C. P., Snaddon, J. L. and Rogers, A. (2018), ‘Audiomoth: Evaluation of a smart open acoustic device for monitoring biodiversity and the environment’, *Methods in Ecology and Evolution* **9**(5), 1199–1211.

Honrado, J. P., Pereira, H. M. and Guisan, A. (2016), ‘Fostering integration between biodiversity monitoring and modelling’, *Journal of Applied Ecology* **53**(5), 1299–1304.

Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.-P., Champ, J., Planqué, R., Palazzo, S. and Müller, H. (2016), Lifeclef 2016: multimedia life species identification challenges, *in* ‘International Conference of the Cross-Language Evaluation Forum for European Languages’, Springer, pp. 286–310.

Kahl, S., Wilhelm-Stein, T., Hussein, H., Klinck, H., Kowerko, D., Ritter, M. and Eibl, M. (2017), ‘Large-scale bird sound classification using convolutional neural networks’, *Working notes of CLEF* .

Kiskin, I., Orozco, B. P., Windebank, T., Zilli, D., Sinka, M., Willis, K. and Roberts, S. (2017), ‘Mosquito detection with neural networks: the buzz of deep learning’, *arXiv preprint arXiv:1705.05180* .

Klingbeil, B. T. and Willig, M. R. (2015), ‘Bird biodiversity assessments in temperate forest: the value of point count versus acoustic monitoring protocols’, *PeerJ* **3**, e973.

Knight, E., Hannah, K., Foley, G., Scott, C., Brigham, R. and Bayne, E. (2017), 'Recommendations for acoustic recognizer performance assessment with application to five common automated signal recognition programs', *Avian Conservation and Ecology* **12**(2).

Kobayasi, K. I. and Riquimaroux, H. (2012), 'Classification of vocalizations in the mongolian gerbil, meriones unguiculatus', *The Journal of the Acoustical Society of America* **131**(2), 1622–1631.

Legg, C. J. and Nagy, L. (2006), 'Why most conservation monitoring is, but need not be, a waste of time', *Journal of environmental management* **78**(2), 194–199.

Mac Aodha, O., Gibb, R., Barlow, K. E., Browning, E., Firman, M., Freeman, R., Harder, B., Kinsey, L., Mead, G. R., Newson, S. E. et al. (2018), 'Bat detectivedeep learning tools for bat acoustic signal detection', *PLoS computational biology* **14**(3), e1005995.

McFee, B., McVicar, M., Balke, S., Thom, C., Lostanlen, V., Raffel, C., Lee, D., Nieto, O., Battenberg, E., Ellis, D., Yamamoto, R., Moore, J., WZY, Bittner, R., Choi, K., Friesch, P., Stter, F.-R., Vollrath, M., Kumar, S., nehz, Waloschek, S., Seth, Naktinis, R., Repetto, D., Hawthorne, C. F., Carr, C., Santos, J. F., JackieWu, Erik and Holovaty, A. (2018), 'librosa/librosa: 0.6.2'.
URL: *https://doi.org/10.5281/zenodo.1342708*

Newson, S. E., Bas, Y., Murray, A. and Gillings, S. (2017), 'Potential for coupling the monitoring of bush-crickets with established large-scale acoustic monitoring of bats', *Methods in Ecology and Evolution* **8**(9), 1051–1062.

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C. and Clune, J. (2018), 'Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning', *Proceedings of the National Academy of Sciences* p. 201719367.

Ovaskainen, O., Moliterno de Camargo, U. and Somervuo, P. (2018), 'Animal sound identifier (asi): software for automated identification of vocal animals', *Ecology letters* **21**(8), 1244–1254.

Porter, J., Arzberger, P., Braun, H.-W., Bryant, P., Gage, S., Hansen, T., Hanson, P., Lin, C.-C., Lin, F.-P., Kratz, T. et al. (2005), 'Wireless sensor networks for ecology', *AIBS Bulletin* **55**(7), 561–572.

Porter, J. H., Nagy, E., Kratz, T. K., Hanson, P., Collins, S. L. and Arzberger, P. (2009), 'New eyes on the world: advanced sensors for ecology', *BioScience* **59**(5), 385–397.

Proença, V., Martin, L. J., Pereira, H. M., Fernandez, M., McRae, L., Belnap, J., Böhm, M., Brummitt, N., García-Moreno, J., Gregory, R. D. et al. (2017), 'Global biodiversity monitoring: from data sources to essential biodiversity variables', *Biological Conservation* **213**, 256–263.

Ramos-Fernández, G. (2008), 'Communication in spider monkeys: the function and mechanisms underlying the use of the whinny', *Spider monkeys: behavior, ecology, and evolution of the genus Ateles. Cambridge University Press, New York* pp. 220–235.

Rayment, W., Dawson, S. and Slooten, L. (2009), 'Use of t-pods for acoustic monitoring of cephalorhynchus dolphins: a case study with hectors dolphins in a marine protected area', *Endangered Species Research* **10**, 333–339.

Rogers, T. L., Ciaglia, M. B., Klinck, H. and Southwell, C. (2013), 'Density can be misleading for low-density species: benefits of passive acoustic monitoring', *PLoS One* **8**(1), e52542.

Salamon, J. and Bello, J. P. (2017), 'Deep convolutional neural networks and data augmentation for environmental sound classification', *IEEE Signal Processing Letters* **24**(3), 279–283.

Shonfield, J. and Bayne, E. (2017), 'Autonomous recording units in avian ecological research: current use and future applications', *Avian Conservation and Ecology* **12**(1).

Sprengel, E., Jaggi, M., Kilcher, Y. and Hofmann, T. (2016), Audio based bird species identification using deep learning techniques, *in* 'LifeCLEF 2016', number EPFL-CONF-229232, pp. 547–559.

Stowell, D., Stylianou, Y., Wood, M., Pamuła, H. and Glotin, H. (2018), 'Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge', *arXiv preprint arXiv:1807.05812* .

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

Strout, J., Rogan, B., Seyednezhad, S. M., Smart, K., Bush, M. and Ribeiro, E. (2017), Anuran call classification with deep learning, *in* 'Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on', IEEE, pp. 2662–2665.

Thessen, A. (2016), 'Adoption of machine learning techniques in ecology and earth science', *One Ecosystem* **1**, e8621.

Underwood, N., Hambäck, P. and Inouye, B. (2005), 'Large-scale questions and small-scale data: empirical and theoretical methods for scaling up in ecology', *Oecologia* **145**(2), 176–177.

Versteegh, M., Kuhn, J., Synnaeve, G., Ravaux, L., Chemla, E., Cäsar, C., Fuller, J., Murphy, D., Schel, A. and Dunbar, E. (2016), 'Classification and automatic transcription of primate calls', *The Journal of the Acoustical Society of America* **140**(1), EL26–EL30.

Villanueva-Rivera, L. J. and Pijanowski, B. C. (2012), 'Pumilio: a web-based management system for ecological recordings', *The Bulletin of the Ecological Society of America* **93**(1), 71–81.

Waser, P. M. and Waser, M. S. (1977), 'Experimental studies of primate vocalization: Specializations for long-distance propagation', *Zeitschrift für Tierpsychologie* **43**(3), 239–263.

Wrege, P. H., Rowland, E. D., Keen, S. and Shiu, Y. (2017), 'Acoustic monitoring for conservation in tropical forests: examples from forest elephants', *Methods in Ecology and Evolution* **8**(10), 1292–1301.

Zamora-Gutierrez, V., Lopez-Gonzalez, C., Gonzalez, M. C. M., Fenton, B., Jones, G., Kalko, E. K., Puechmaille, S. J., Stathopoulos, V. and Jones, K. E. (2016), 'Acoustic identification of mexican bats based on taxonomic and ecological constraints on call design', *Methods in Ecology and Evolution* **7**(9), 1082–1091.