IST-Africa 2023 Conference Proceedings
Miriam Cunningham and Paul Cunningham (Eds)
IST-Africa Institute and IIMC, 2023
ISBN: 978-1-905824-71-7

Screening of Bioacoustics Recordings for Ecosystem Monitoring - an Application to Audio Recordings of Bats

Muhinyia wa NDEGWA[1], Ciira wa MAINA[2], Mark KEITH[3]
[1,2]*Centre for Data Science and Artificial Intelligence, Dedan Kimathi University of Technology. P.O. BOX - PRIVATE BAG – 10143, Dedan Kimathi - Nyeri, Kenya*
[3]*Department of Zoology and Entomology, University of Pretoria,*
Private Bag X20, Hatfield 0028, Pretoria, South Africa
Email: [1]muhinyia.ndegwa@interns.dkut.ac.ke, [2]ciira.maina@dkut.ac.ke
[3]mark.keith@up.ac.za

Abstract: Ecosystem monitoring using bats as bio indicators can be achieved through their echolocation recordings. By analysing the content of bats' recordings, ecologists can infer aspects such as species diversity, population dynamics among others. This information is crucial in assessing ecosystem health. Collecting recordings passively is straightforward by deploying recorders. Drawing inferences from these recordings calls for automatic screening tools to help ecologists detect, localise and characterise bat calls present. We developed an audio processing pipeline to enhance screening of acoustic recordings for bats' calls detection and localization. The recordings were collected within Dedan Kimathi University of Technology using AudioMoth recorders. Our pipeline leverages simple methods such as median clipping and serves as an initial screening stage before further analysis using sophisticated methods such as machine learning techniques. The pipeline obtained good results and proved effective in detecting and localising bat calls from audio recordings.

Keywords: Bioacoustics, AudioMoth, Spectrogram, Echolocation, Denoising

1. Introduction

Human activities have resulted in degradation of ecosystems. This has led to adverse effects on biodiversity including habitat loss, species decline and extinction. The structure and functioning of ecosystems are changing at an unprecedented rate [1]. Ecosystem monitoring is critically important to understand these changes and subsequently inform the best conservation and management strategies. For successful ecosystem monitoring, it is important to consider bio indicators that respond rapidly and represent elements of change under consideration [1]. Bats occupy a wide range of ecological niches; their multisensory nature makes them sensitive to changes in the ecosystem. This qualifies them as important targets while assessing ecosystem health [2].

Majority of bats species emit echolocation calls while navigating, searching for prey or even communication purposes [3]. These calls constitute important acoustic signatures that can be analysed for various purposes such as studying population dynamics and species classification.

Acoustic sensing is one of the methods that has proved instrumental in ecosystem monitoring. This method is preferred as it offers a non-invasive, passive and accurate way of collecting acoustic data. It is straightforward and inexpensive to collect huge quantities of bioacoustics data. The major hurdle lies in the analyses of these audio recordings due to their large quantity and high noise prevalence. Noise may override bat calls making it almost impossible to perceive them. In other scenarios, echolocating bats may be far away from the recording device thus diminishing the calls [3].

There is a need to devise reliable, robust, inexpensive and automatic audio analysis tools to facilitate extraction of useful information from these noisy recordings. Manual inspection of recordings' spectrograms have been used for analysis, however, this method is time intensive, tedious and the results are highly dependent on the users' experience. There are available commercial tools for audio analysis that employ methods such as amplitude thresholding, locating areas of smooth frequency change and comparison of spectrograms to a well-known reference spectrogram [3, 4]. These comparisons require experienced expertise, and are time consuming.

This work presents an initial screening of acoustic recordings using a simple and readily available open source tool that we developed. It uses libraries for Digital Signal Processing such as librosa, image processing algorithms as well as median clipping methods. The proposed processing pipeline will serve as the initial data cleaning stage before using machine-learning methods. This ensures data is of good quality prior to conducting further analysis and saves on storage space.

2. Objectives

2.1 Main Objective

To develop an audio processing pipeline for screening bioacoustics recordings to help detect, localise and characterise bat calls.

2.2 Specific Objectives

a) Develop appropriate signal processing techniques to identify potential bat calls in audio data processing.
b) Exploit bat call properties such as expected duration and inter pulse duration to eliminate spurious calls.
c) Investigate appropriate clustering approaches to identify similar call types to aid in species identification.
d) Test the pipeline on recordings collected at Dedan Kimathi University of Technology.

3. Methodology

3.1 Data Description

The dataset consisted of 285 recordings each 55 seconds long collected within Dedan Kimathi University of Technology for a period of one week in June 2022. Data collection was conducted passively using AudioMoth recorders, which were deployed in areas frequented by bats such as deserted old buildings. The recorders were set up in the evening since bats are nocturnal. Since bats emit ultrasonic calls, recording was done at a high sample rate of 250 kHz to ensure high frequency activities of up to 125 kHz were captured.

3.2 Audio Loading

The input to the processing pipeline are WAV recordings from which we aim to detect and localise bat calls. We therefore need to load and represent them in a convenient format for analysis. We used the librosa module to load all the raw audio files and represented them as numpy arrays.

3.3 Spectrogram Computation

Spectrograms describe a visual representation of the frequency spectrum present in a signal and their variation with time. This makes it possible to analyse the frequency content of a signal with respect to time unlike when we just have amplitudes and time. Such a signal is said to be in the frequency domain. To achieve this transformation we perform Fourier transform decomposition, which breaks a signal into its basic sinusoidal components where each has a specific frequency, amplitude and phase. To prevent information loss, the signal is first split into short overlapping segments (windows); we take the Fourier transform for each segment and then combine the results for all segments. This mechanism is called Short Time Fourier Transform (STFT). The size of these windows determines the time and frequency resolution obtained. A small window translates to high time resolution but low frequency resolution while a larger window size favours frequency resolution over time resolution. Other parameters of interest are the hop size, overlap and sampling rate. The hop size defines the distance to slide the window from frame to frame. The overlap is the overlapping length between subsequent frames while the sampling rate is the number of samples drawn per unit time (normally per second). Various window functions can be used in a short time Fourier transformation [4]. In this pipeline, a hanning window was preferred since it is a smooth function, which helps reduce spectral leakages. Table 1 shows the parameters used to compute the spectrograms for all the audio recordings in this study.

Table 1: Parameters Used in Computing Spectrograms

Parameter	Sampling rate	Window	Window-length	Window-length	Overlap	Hop length
Value	250 kHz	hanning	10 Ms	2500 samples	75%	625

3.4 Denoising

Noise has the effect of blurring the target audio signal, denoising helps to suppress noise making it easy to detect the desired signal. There are various denoising methods such as filtering, spectral subtraction and spectral gating [6]. In this study, denoising followed the median thresholding approach [7]. The intuition here is that bat calls will occur in high frequency bands (higher than the median). We therefore filtered out noise by discarding low frequency bands. To achieve this, we select pixels that are three times larger than the row median and three times larger than the column median. High pass and low pass filters of 120 kHz and 12 kHz respectfully were used to filter out low and high broadband noises. All pixel values that satisfy the criteria are set to 1 while all other pixels are set to 0. The column mask and the row mask are both combined to obtain a binary mask.

The resulting binary mask was subjected to opening using a 3 × 5 kernel. The choice of kernel was through experiments. In the opening process, the mask was first passed through binary erosion to clear small objects. Finally, a binary dilation is done to expand the objects in the mask. This results in an indicator vector similar to the shape of the spectrogram. This indicator provides the basis for further processing.

3.5 Call Activity Detection

To detect calls we considered known characteristics of bat calls. The parameters considered here are minimum duration for a call, the distance between calls and the pulse sequencing. For duration, a call should be at least 10 milliseconds long. Calls within a range of 500 milliseconds are considered to have emanated from the same source and thus are processed together as a group. For consistent detection, calls that do not belong to a group are eliminated. A group would therefore consist of at least two pulses. We developed an algorithm to detect call activities from the resultant indicator vector following the steps outlined below.

3.5.1 Components Extraction

The algorithm processes the mask and obtains regions with a series of ones. These denote interesting parts of the audio with the potential of containing bats calls. The output to this is a grouping of pulse series alongside their lengths.

3.5.2 Elimination of Short Components

Short components originate from noise. We therefore eliminated components that do not meet the threshold of a valid bat call duration i.e. 10 milliseconds.

3.5.3 Group Close Components Together

Components are compared based on the time interval between pulses. Those that fall within the distance range of 500 milliseconds are collected into a group and are processed as one group.

3.5.4 Elimination of Lone Components

We only included two pulses or more (pulse sequence) and single pulses were excluded. A variety of pulse sequence characteristics are required to allow for reliable and consistent identification.

3.6 Parameters Extraction

Once calls were detected, we extracted call parameters to be used in machine learning modelling. The parameters of interest are maximum frequency, minimum frequency, average frequency and the duration of a call (call length). YIN [10] was used in estimating the frequencies. These parameters are illustrated in figure 1.

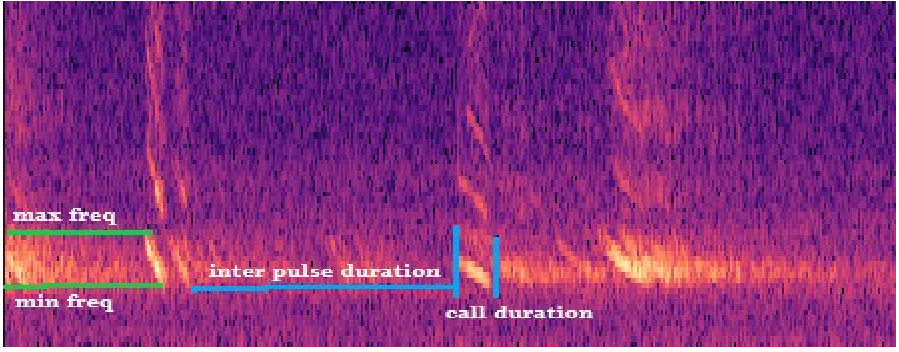


Figure 1: Call Parameters from a Recording Obtained Using AudioMoth Device at Dedan Kimathi University of Technology in June 2022

3.7 Unsupervised Machine Learning Methods (UMLM)

UMLM is described as a machine learning technique used in extracting patterns in unlabelled data. Clustering is the main task around unsupervised machine learning and aims at grouping similar objects together. Data points exhibiting similar characteristics are grouped into one cluster, data points drawn from different clusters would therefore be unique.

By applying clustering methods, we aim to discover recordings containing calls from the same species in which they should be collected in one cluster. Gaussian mixture model (GMM), which is a density based probabilistic algorithm, was fitted. The input feature to this algorithm consists of an array of parameters illustrated in Figure 1.

3.7.1 Model Evaluation

Evaluation of the clustering algorithm was through visual inspection of the spectrograms from a cluster. Spectrograms drawn from the same cluster should look alike.

4. Technology Description

4.1 AudioMoth

AudioMoth [8] is a passive acoustic monitoring tool built on a single credit card sized PCB. Compared to other acoustic tools, AudioMoths are effective due to their low power consumption, small size and ease of use. The device costs roughly 90 USD per unit, which is about ten times cheaper compared to other commercial acoustic tools [8]. The device's small size makes it portable and easy to deploy in various ecosystems while the low power consumption benefits it for long-term deployments. Configuration of AudioMoth devices for deployment is intuitive. AudioMoths have several recording sample rates, which make them suitable for capturing various acoustic activities such as insects' sounds recorded at a sample rate of 8 KHz, perceivable animal's sounds occurring at a sample rate of 48 KHz and ultrasonic calls occurring at very high frequencies.

4.2 Librosa

Librosa [9] is a free and open source python library for music and audio analysis. It enables developers to develop applications for working with sound and music documents utilising python programming language. The library is easy to use and provides various APIs for audio loading, computations, and visualisation. This tool facilitated the loading of WAV files and conversion from time to frequency domain.

4.3 Audacity

It is a free, open source, cross platform audio editor software supporting recording, playing, importing, editing among other functionalities. It supports various audio formats. The software enables features for customizable spectrogram mode, frequency components analysis, amplitude envelope analysis, among other features. We used this tool for audio recordings validation.

4.4 YIN

It is a robust fundamental frequency estimation algorithm based on the autocorrelation method [7], which exploits the fact that Autocorrelation function (ACF) of a periodic signal has peaks at integer multiples of the signal period. The spacing between peaks is equal to the signal period. Yin computes the ACF of a signal, finds the first peak that corresponds to the period of the fundamental frequency. Getting the inverse of this period therefore gives the fundamental frequency that is then compared to a pre-set threshold.

5. Results and Discussion

To outline the efficacy of the processing pipeline, we demonstrate the transformations an audio recording undergoes in various processing stages based on the extraction/processing pipelines established.

Figure 3 shows the waveform for a sample recording and its corresponding spectrogram.

Table 2: Parameters for Calls Extraction

Parameter	Pulse Duration	Inter-Pulse Duration
Value	10 Ms	500 Ms

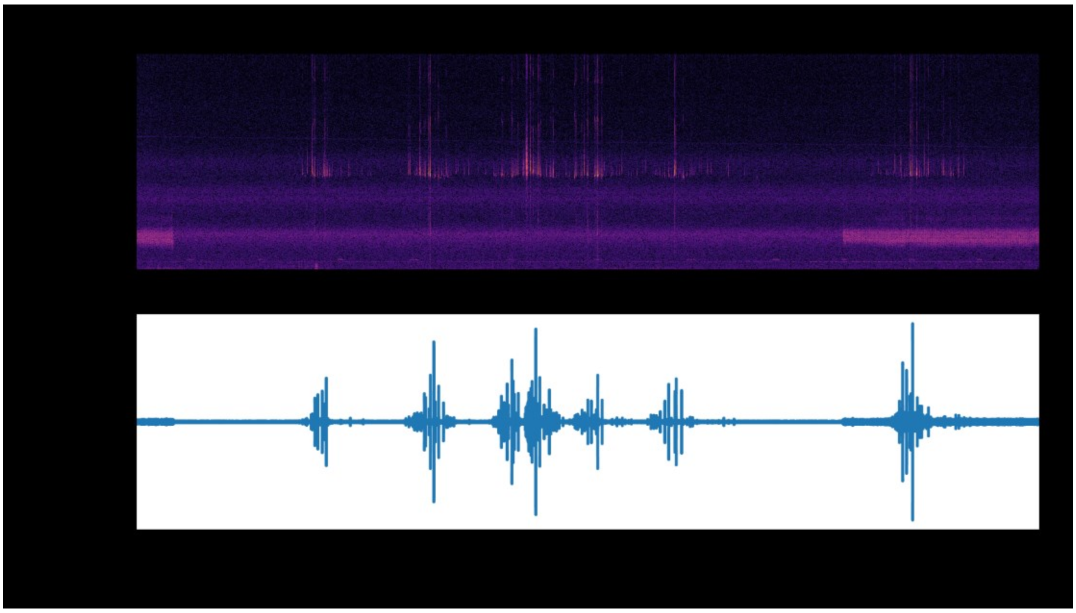


Figure 3: The Waveform and Corresponding Spectrogram for a Sample Recording from Dedan Kimathi University of Technology (June 2022) using AudioMoth recorder

5.1 Call Activity Detection

Figure 4 illustrates the call detection process. The first subplot shows the pulses present in this section. In the second plot, short pulses are eliminated. In the subsequent subplot, we group closer pulses together. Finally, pulses that do not belong to a group are dropped. Table 2 shows the parameters used in this process. The results are a binary mask outlining audio sections containing calls.

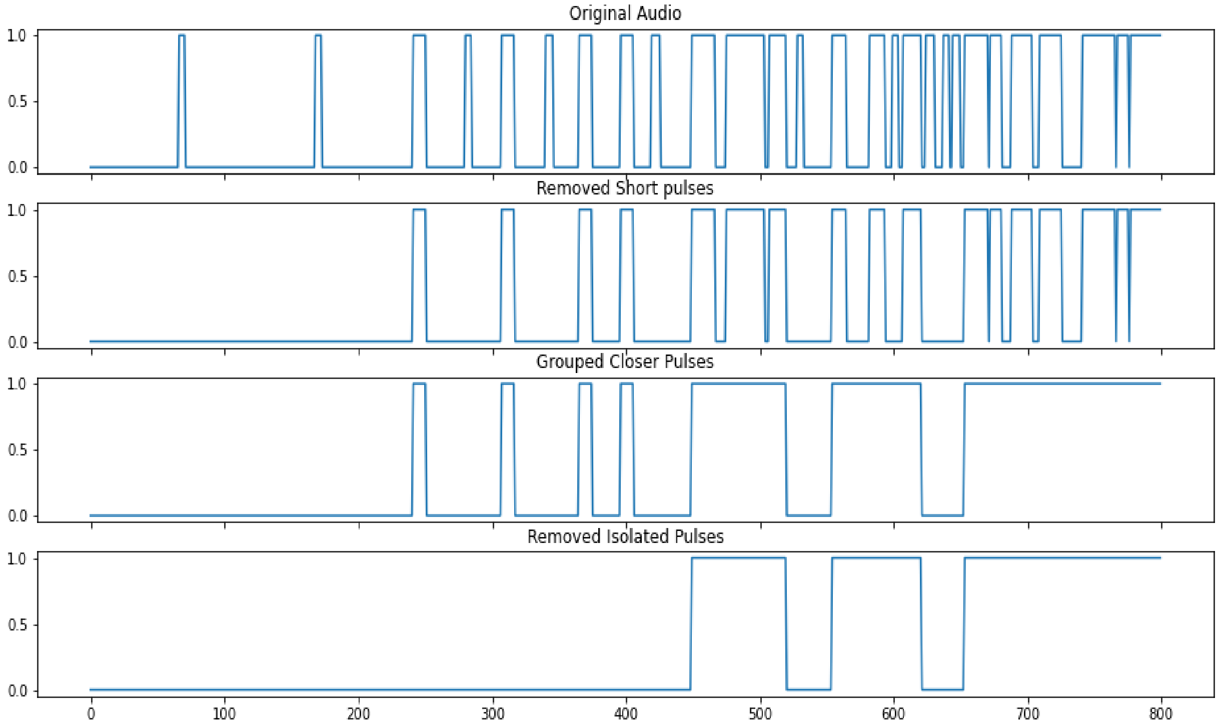


Figure 4: Pulses Extraction Process

Applying the mask obtained above on the recording results in audio segments where the calls are present. This results in shorter audio segments. Figure 5 illustrates a segment after filtering for bat calls. The locations highlighted in green show bat calls.

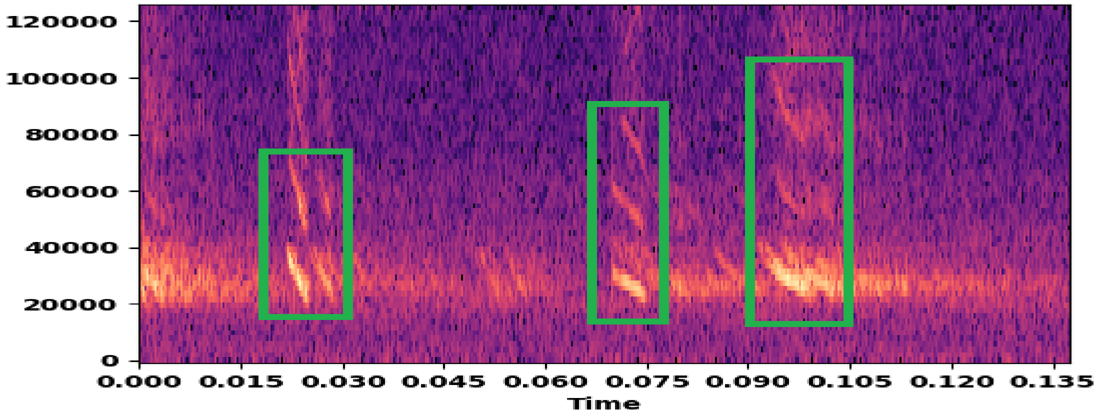


Figure 5: Spectrogram for a Section of the Processed Recording

For precision, the segments were reprocessed using finer parameters i.e. instead of using a 10ms long window for spectrogram computation; we used a window 1ms long. This resulted in parameters shown in Table 3. The goal was to obtain pulses within a group.

Table 3: Parameters Used for Finer Computation

Parameter	Sampling rate	Window	Window length	Window length	Overlap	Hop length
Value	250 kHz	hann	1 Ms	250 samples	75%	62

The spectrogram for an audio segment (Figure 6) represents pulses within this segment while the last plot illustrates pulses after elimination of short pulses. This allows for sequential pulse identification and obtaining call sequence characteristics such as inter pulse duration.

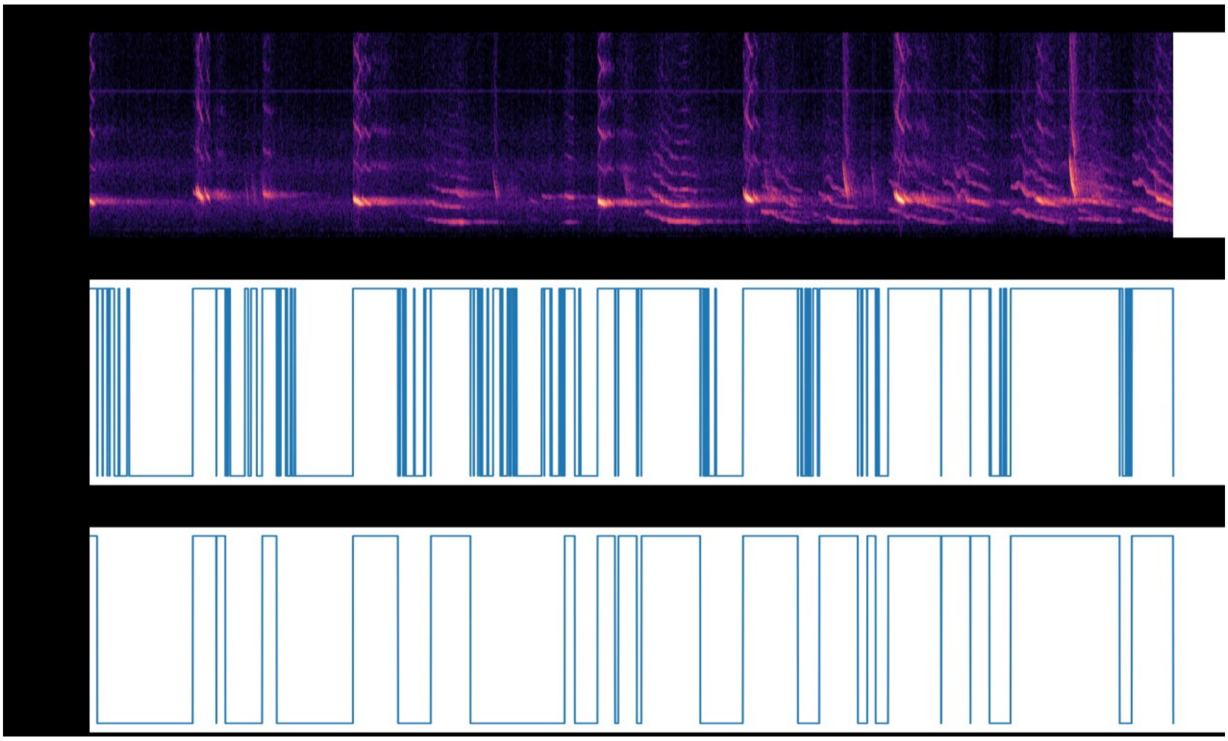


Figure 6: Pulses Re-Computation for an Audio Segment to Extract Sequential Pulses

5.2 Parameter Extraction

The YIN algorithm was used to estimate the fundamental frequencies. From this, the maximum frequency, minimum frequency, mean frequency and the pulse duration were calculated per pulse. Figure 7 shows an audio segment and the estimated frequency. Table 4 illustrates the extracted parameters for this segment.

Table 4: Call Pulse Parameters Derived from Re-Computation from the Calls Recorded within Dedan Kimathi University of Technology.

Pulse	Min frequency (Hz)	Max frequency (Hz)	Average frequency (Hz)	Pulse duration (Ms)
1	10000	37389.64	17232.88	0.01984
2	10274.69	37522.89	23173.78	0.010416
3	10000	34572.42	11868.97	0.062992

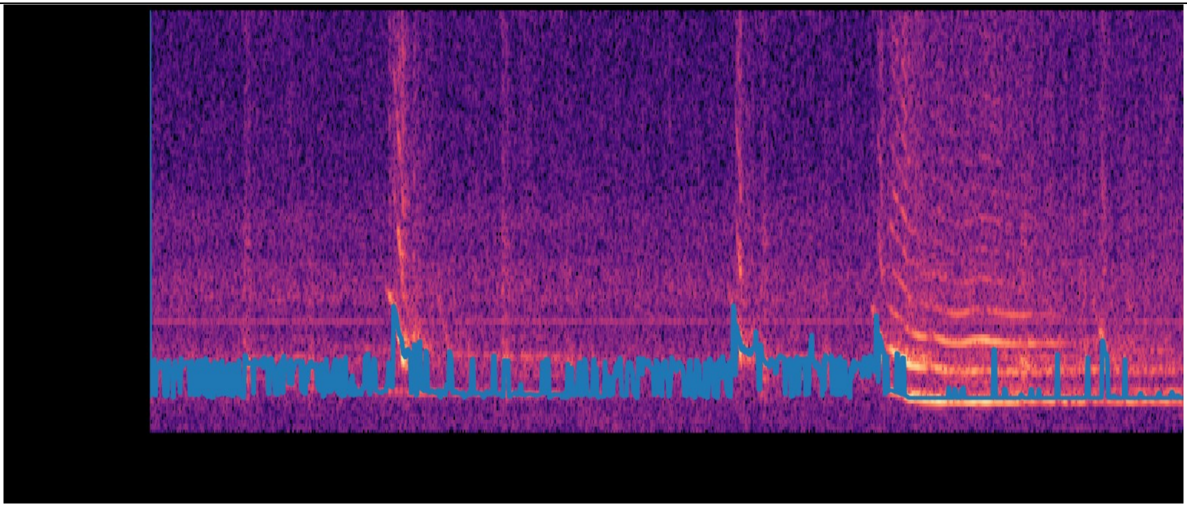
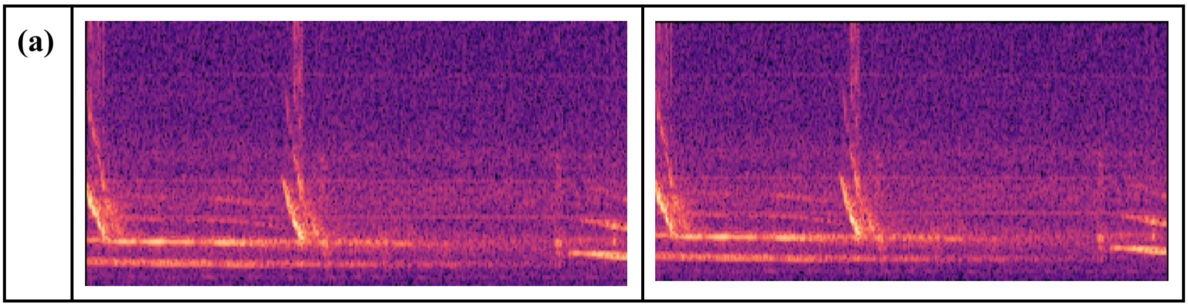


Figure 7: Frequency Estimation Using YIN [10]

5.3 Clustering

We trained a Gaussian mixture model to cluster the recordings based on the extracted features. Figure 8 shows spectrograms for random file recordings drawn from two different clusters. Spectrograms in **(a)** are for files from the same cluster, the pulses appear similar in their structure. The same can be said for spectrograms in **(b)** which are for two recordings from the same cluster. These similar structures in pulses and frequency components infer that the audio recordings emanated from the same species.



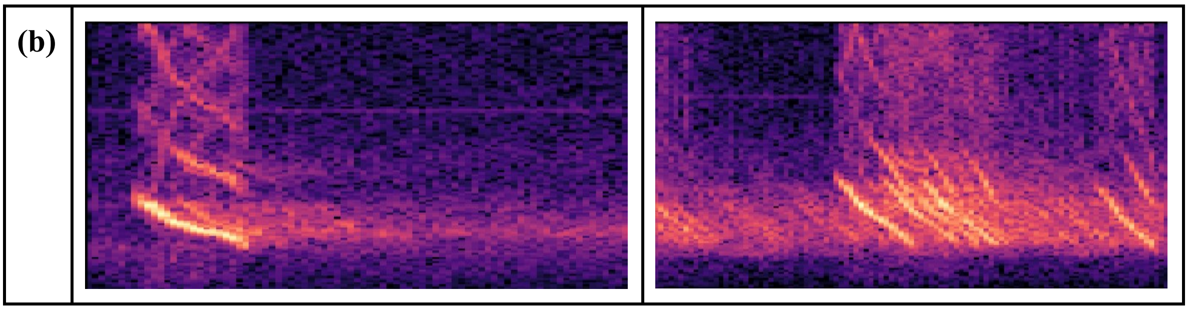


Figure 8: Spectrograms for Random Files in a Cluster

6. Business Benefits

Acoustics recordings obtained by ecologists contain extensive noise. Often, the desired calls are present in a few sections of the recording. Processing of these recordings manually is daunting and time consuming. Furthermore, storage of full spectrum audio is expensive. Machine learning models are adversely affected by this noise and thus an attempt to draw patterns from this noisy data may be intractable. By setting up a rapid screening pipeline, the resulting data is of good quality thus better suited for machine learning modelling. It also consumes less storage space.

This open source pipeline presents a cheaper method for bats' acoustic processing. It is tailored to help ecologists in processing of bats' recordings for species identification and classification, population monitoring and ultimately ecological monitoring. Development of this pipeline took a timeframe of six months. Extra three months will be required for development of a user interface before the product is availed for use.

7. Conclusions

The focus in this study was on development of a processing pipeline to help detect and localise bats' calls from audio recordings. We developed a pipeline capable of detecting and localising bat calls from noisy recordings. The pipeline was based on the median thresholding method. We also extracted call parameters i.e. peak frequency, minimum frequency, characteristic frequency and duration for every detected call that we used in training a clustering model for species detection. As illustrated, the pipeline was able to detect and localise bat calls as well as classify bats species.

Our next steps will be to build a user interface and annotate datasets for pipeline verification. For future work, more noise reduction methods can be explored; it would also be necessary to build bat classifiers using machine-learning methods and deploy them for automatic classification of bat species in real time.

Acknowledgement

We would like to thank Data Science Africa for support through the Affiliated Centre Program as well as Canada's International Development Research Centre (IDRC) and the Swedish International Cooperation Agency (SIDA) through the Artificial Intelligence Program for Development in Africa (AI4D Africa) Program.

References

[1] Jones, G., Jacobs, D. S., Kunz, T. H., Willig, M. R., & Racey, P. A. (2009). Carpe noctem: the importance of bats as bioindicators. Endangered Species Research, 8(1-2), 93-115.

[2] Schnitzler H-U, Moss CF, Denzinger A. From spatial orientation to food acquisition in echolocating bats. 2003 Trends in Ecology & Evolution., 18(8):386±94. http://dx.doi.org/10.1016/S0169-5347(03)00185-X.

[3] Walters CL, Collen A, Lucas T, Mroz K, Sayer CA, Jones KE. (2013) Challenges of Using Bioacoustics

to Globally Monitor Bats. In: Adams RA, Pedersen SC, editors. Bat Evolution, Ecology, and Conservation. New York, NY: Springer New York; p. 479±99.

[4] Rannisto, M. (2020). Detecting Bat Calls from Audio Recordings.

[5] Sainburg, T. (2022) Noise reduction using spectral gating in python. https://timsainburg.com/noise-reduction-python.html. Accessed: 2020-10-20.

[6] Sprengel, E., Jaggi, M., Kilcher, Y., & Hofmann, T. (2016). Audio based bird species identification using deep learning techniques (No. CONF, pp. 547-559).

[7] D. M. Kiapuchinski, C. R. E. Lima, and C. A. A. Kaestner. (2012) Spectral noise gate technique applied to birdsong preprocessing on embedded units. In 2012 IEEE International Symposium on Multimedia, pages 24–27, 2012.

[8] Hill, A. P., Prince, P., Piña Covarrubias, E., Doncaster, C. P., Snaddon, J. L., & 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.

[9] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015, July). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (Vol. 8, pp. 18-25).

[10] De Cheveigné, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111(4), 1917-1930.