École polytechnique de Louvain

# Automated monitoring of bat species in Belgium

Authors: **Maxime Franco, Corrado Lipani**
Supervisors: **Olivier Bonaventure, Siegfried Nijssen**
Reader: **Maxime Piraux**
Academic year 2019–2020
Master [120] in Computer Science

# Contents

# List of Figures

# List of Tables

**Abstract**

Anthropogenic change has become a recurrent subject in today's society. The population's growth, pollution and the over-consumption of resources is heavily damaging the biodiversity. A way to identify these changes is by monitoring bioindicators. Bat species can fit this role as they can easily show qualitative information on the environment they are in. The monitoring of bats can be achieved through bioacoustics. In fact, bats produce ultrasounds for most of their activities. The objective of this work is to create a reliable Belgian bat species classification tool using machine learning techniques trained on bat calls audio files.

Data provided by the association Natagora was processed to obtain the most accurate data available to train our model. The project Batmen was created to answer this need with the help of a convolutional neural network. Audio files were transformed into spectrograms before being fed to the neural network. Our results show that the classification through convolutional neural networks was accurate provided that the dataset was big and balanced enough. Batmen showed average results for the classification of bat species using groups to gather species containing similarities. Improvements can be done with more accurate labeling, more balanced data and different multiclass classification strategies through convolutional neural networks improvements.

**Acknowledgements**

We would like to thank both our supervisors, Olivier Bonaventure and Siegfried Nijssen for their guidance throughout the process. Olivier provided us with his knowledge and the equipment needed for our work. Siegfried gave us valuable advice and suggestions on the subject of machine learning. Then, we would also like to thank Maxime Piraux for his precious comments on this thesis and overall guidance. Furthermore, we wish to thank Natagora and in particular Claire Brabant for giving us the opportunity to gather valuable knowledge on bats and for providing us with a large collection of data. Without them, we would not have been able to undertake this project.

Finally, we must express our very profound gratitude to our families for their unfailing support and encouragement especially during these tough times. This accomplishment would not have been possible without them. Thank you.

# Chapter 1

# Introduction

Before we demonstrate the elaboration and the usage of the bat classification, it is essential to take a step back and understand the context behind the monitoring and classification of bats. To that end, a quick discussion on the role of bioindicators and a motivation for building such monitoring tools is provided.

The focus of this thesis is to provide a reliable and efficient classification tool improving the study of bat species in Belgium. This is achieved through the use of machine learning and specifically convolutional neural networks. To this end, audio recordings are fed to the trained neural network that will analyse them as spectrograms and give a prediction on the species.

By combining both the fields of biology and computer science, the tool provides an alternative for the scientists in their bat monitoring work as well as a solution for different uses such as entertainment or education. Finally, the knowledge gathered throughout the process can serve as a base for future projects as this project is flexible enough to inspire other classification goals.

At the moment, human activity is one of the most destructive force happening on the Earth. The growth in human population, over-consumption in natural resources, pollution causes irrecoverable damage to the ecosystems and biodiversity. This behavior is likely to stay and increase, leading to the Earth's point of no return. Two keys aspects of this anthropogenic change are climate change and habitat conversion i.e., fundamental change in a natural habitat caused by human activity. Monitoring such broad and complex aspects requires natural tools: bioindicators. By monitoring specific species or group of species, crucial information can be gathered on the environment they are in.

Bats have shown to be sensitive to climate change (variation in temperature causes significant changes in the populations or hibernation cycles). Moreover, activities such as deforestation, urbanisation, pesticide use, wind turbines [21] and overall pollution also cause significant variations in population and species distribution [20].

Fortunately, bats also are very good bioindicators as they present interesting factors:

- Bats have very diverse species and are dispersed on all continents except Antarctica.
- Bats occupy a high trophic level or a high place in their food chain. Their dietary diversity makes them likely to show the effect of pollutants faster than organisms at lower trophic levels.
- Bats have slow reproductive rates, taking a long time to recover from declines which makes them easier to study and less subject to noise.
- Bats can show a wide variety of infectious diseases due to environmental stress.

A way to quantify some of those factors falls under the discipline of bioacoustics i.e., the branch that studies the production, dispersion and reception of animal sounds. Some bats in particular use a fascinating principle called echolocation to navigate and search for food in their environment. Recording and analysing such sounds opens the door to acquire relevant information needed for the monitoring process previously discussed.

In this chapter, we have explained the motivations behind the thesis, the context behind the monitoring of bat species and its usefulness. Our collaboration with the Natagora association is covered in the next chapter, as they provided us with an important set of data. Also, we introduce the different bat species identified in Belgium. Then, we describe the technical aspect of the echolocation principle as well as the bat calls structure, detection and analysis. Several supervised learning techniques are introduced and the motivation behind our choice in machine learning algorithm is given. Then, we review the project Bat detective that we have considered as state of the art for search-phase bat echolocation calls detection.

A discussion on the processing of the audio files and labels is held. Next, we introduce our implementation of the classification tool, Batmen, by explaining our method and the modifications made to the original Bat detective project. This is followed by an assessment on the performance of the tool. Then, we explain how to replicate the project. Finally, we end with a discussion on what can be improved and give a final conclusion.

# Chapter 2

# Bat species in Belgium

This chapter describes the twenty-three species that can be found in Belgium. Some background is given on their habitat and distribution inside the country. In addition, it introduces the group Plecotus from Natagora who is responsible for the monitoring of bats in Belgium and describes their methodology.

## 2.1 Presentation

There are to this day twenty-three species [37] identified in Belgium. The table 2.1 summarizes these species by giving their common names, scientific names and their individual codes which are used mainly in metadata and for short notations. In addition, even though twenty-three species are listed above, it is critical to note that some species such as species of the genus Pletocus or Myotis are particularly hard to identify. This has led scientists to build a new level of identification in the name of "groups" to improve the quality of identification while slightly reducing reliability [18].

| **Name** | **Scientific name** | **Code** | **Group** |
|---|---|---|---|
| Barbastelle bat | Barbastella barbastellus | Barbar | Barbar_G |
| Serotine bat | Eptesicus serotinus | Eptser | ENVsp |
| Alcathoe Whiskered bat | Myotis alcathoe | Myoalc | Myosp |
| Bechstein's bat | Myotis bechsteinii | Myobec | Myosp |
| Brandt's bat | Myotis brandtii | Myobra | Myosp |
| Pond bat | Myotis dasycneme | Myodas | Myosp |
| Daubenton's bat | Myotis daubentonii | Myodau | Myosp |
| Geoffroy's bat | Myotis emarginatus | Myoema | Myosp |
| Greater Mouse-eared bat | Myotis myotis | Myomyo | Myosp |
| Whiskered bat | Myotis mystacinus | Myomys | Myosp |
| Natterer's bat | Myotis nattereri | Myonat | Myosp |
| Greater noctule | Nyctalus lasiopterus | Nyclas | ENVsp |
| Leisler's bat | Nyctalus leisleri | Nyclei | ENVsp |
| Common noctule | Nyctalus noctula | Nycnoc | ENVsp |
| Kuhl's pipistrelle | Pipistrellus kulhii | Pipkuh | Pip35 |
| Nathusius' pipistrelle | Pipistrellus nathusii | Pipnat | Pip35 |
| Common pipistrelle | Pipistrellus pipistrellus | PippiT | Pip50 |
| Soprano pipistrelle | Pipistrellus pygmaeus | Pippyg | Pip50 |
| Brown Long-eared bat | Plecotus auritus | Pleaur | Plesp |
| Grey Long-eared bat | Plecotus austriacus | Pleaus | Plesp |
| Greater Horseshoe bat | Rhinolophus ferrumequinum | Rhifer | Rhisp |
| Lesser Horseshoe bat | Rhinolophus hipposideros | Rhihip | Rhisp |
| Particoloured bat | Vespertilio murinus | Vesmur | ENVsp |

Table 2.1: Bat species identified in Belgium.

### 2.1.1 Habitat

A common misconception about bats is that they only live in caves. While this assumption is mostly true, bats generally are looking for a place away from predators (or any disturbance), with a sufficient amount of food and stable temperature during the day.

In summer, bats regroup as breeding colonies and generally find home inside trees, buildings, churches (nursery roosts). Between May and June begins the parturition and the breastfeeding process until between July and August where the newborn can already fly. In autumn, bats gather their fat reserves before migrating towards their hibernation roosts.

In winter, bats are hibernating in their roosts, a calm, humid and cold (between 0°C and 10 °C) area. Commonly, they will be roosting sometimes inside of trees, sometimes inside caves, mines or tunnels. Finally, spring is the time when bats leave their roosts and start to hunt again and look for their next summer habitat.

| Species \ Roost | Summer | Winter |
|---|---|---|
| **Barbastelle bat** | Tree holes | Tree holes, underground |
| **Serotine bat** | Buildings, under roofs | Colder parts of caves, buildings (cavities) |
| **Alcathoe Whiskered bat** | Trees [1] | Trees [1] |
| **Bechstein's bat** | Tree holes | Tree holes, underground |
| **Brandt's bat** | Tree holes [1] | Underground [1] |
| **Pond bat** | Buildings, trees | Underground |
| **Daubenton's bat** | Arboreal cavities (near water) | Underground |
| **Geoffroy's bat** | Building (attics) | Underground |
| **Greater Mouse-eared bat** | Building (attics) | Underground |
| **Whiskered bat** | Human habitations | Underground |
| **Natterer's bat** | Tree holes | Underground |
| **Greater noctule** | Arboreal cavities | Arboreal cavities[1] |
| **Leisler's bat** | Arboreal cavities | Arboreal cavities |
| **Common noctule** | Arboreal cavities | Arboreal cavities |
| **Kuhl's pipistrelle** | Buildings (cavities)[1] | Buildings (cavities)[1] |
| **Nathusius' pipistrelle** | Arboreal cavities | Arboreal cavities |
| **Common pipistrelle** | Buildings (cavities) | Buildings (cavities) |
| **Soprano pipistrelle** | Buildings (cavities)[1] | Buildings (cavities), arboreal cavities[1] |
| **Brown Long-eared bat** | Building (attics), tree holes | Underground |
| **Grey Long-eared bat** | Building (attics), tree holes | Underground |
| **Greater Horseshoe bat** | Building (attics), warm underground sites | Underground |
| **Lesser Horseshoe bat** | Building (attics), warm underground sites | Underground |
| **Particoloured bat** | Cliff, wall cracks[1] | Cliff, wall cracks[1] |

Table 2.2: Summer & winter habitat of bats in Belgium.

The table 2.2 provides the usual locations for the species's summer and winter roosts. This information shows us the roost difference for a specific species and within the same genus.

[1]These are hypotheses due to the lack of observations of these species in Belgium

### 2.1.2 Geographic distribution

Several species can be found in specific regions of Belgium:

| Location / Species | Region |
|---|---|
| **Barbastelle bat** | Flanders, Brussels, Wallonia |
| **Serotine bat** | Flanders, Brussels, Wallonia |
| **Alcathoe Whiskered bat** | Wallonia |
| **Bechstein's bat** | Flanders, Brussels, Wallonia |
| **Brandt's bat** | Flanders, Brussels, Wallonia |
| **Pond bat** | Flanders, Brussels, Wallonia |
| **Daubenton's bat** | Flanders, Brussels, Wallonia |
| **Geoffroy's bat** | Flanders, Brussels, Wallonia |
| **Greater Mouse-eared bat** | Flanders, Brussels, Wallonia |
| **Whiskered bat** | Flanders, Brussels, Wallonia |
| **Natterer's bat** | Flanders, Brussels, Wallonia |
| **Greater noctule** | Flanders |
| **Leisler's bat** | Flanders, Brussels, Wallonia |
| **Common noctule** | Flanders, Brussels, Wallonia |
| **Kuhl's pipistrelle** | Brussels, Wallonia |
| **Nathusius' pipistrelle** | Flanders, Brussels, Wallonia |
| **Common pipistrelle** | Flanders, Brussels, Wallonia |
| **Soprano pipistrelle** | Flanders, Brussels, Wallonia |
| **Brown Long-eared bat** | Flanders, Brussels, Wallonia |
| **Grey Long-eared bat** | Flanders, Brussels, Wallonia |
| **Greater Horseshoe bat** | Flanders, Wallonia |
| **Lesser Horseshoe bat** | Flanders, Wallonia |
| **Particoloured bat** | Flanders, Brussels, Wallonia |

Table 2.3: Geographic distribution of bat species in Belgium.

As we can see from the table 2.3, most bats have been observed all over Belgium. It is important to note that the species populations can vary significantly throughout the years, e.g., the Lesser Horseshoe bat went from 300.000 individuals in 1950 to 500-1000 individuals in 2017 [26].

## 2.2 Natagora

Natagora is a Belgian non-profit organization that was founded in 2003 by the two associations Aves and Réserves Naturelles RNOB. Their main goals are to protect, study and redeploy biodiversity in harmony with human activity especially in Wallonia and Brussels. To make this happen, Natagora provides conferences, animations and guided tours as well as fieldwork throughout the year. In particular, the organization happens to own a group named Plecotus working specifically on bats. The group accepted a collaboration between both parties and allowed the acquisition of context, knowledge and most importantly usable data for the project.

## 2.3 Data collection

### 2.3.1 Devices

The Plecotus group uses several devices to listen to and store bat calls:

- Song Meter SM2 by Wildlife Acoustics
- Song Meter SM4BAT by Wildlife Acoustics

The devices handle both .wav and .wac formats. The latter is a proprietary compressed format that allows options like GPS tracking every second. Also, recording can be achieved with triggers (recording only when activity is detected) on different audio channels. In fact, it is useful to save space or to record multiple animals at the same time [2].

### 2.3.2 Data processing

To collect sounds without disturbing the bats, the Plecotus group disposes the Song Meter near the bat's habitat and it is parametrized to start collecting data between 10 pm and 5 am. After leaving the microphone for one to three days around the habitat, the group can start the analysis of the recordings. First, Kaleidoscope [1] is used to separate .wac files into .wav files of 5 seconds, it also performs a first sort between files containing assumed bat calls and noise. Then, the SonoChiro software [7] is used on sounds to detect and identify the bats. The results are stored inside a csv file containing the species with a confidence index. Finally, the group validates manually the calls with the higher confidence indexes (at least one per species) with the BatSound software. This software allows them to visualise and to listen to the sounds in time expansion to determine the species more easily.

## 2.4 Conclusion

This chapter has shown us that Belgium has a good variety of bat species. Each of these species have very different behaviors in terms of roosting and overall location. Organizations such as Natagora go through a long and demanding process to gather and process bat calls data. First, there is a need for specific ultrasonic recorders that have to be placed near the bat's habitats. Then, the data have to be cut and visualised to allow their classification. Finally, a manual identification has to be done to confirm the previous predictions.

# Chapter 3

# Classical approaches

In this chapter, the bat echolocation principle is described. A more technical description of the detection techniques of these calls is provided as scientists use signal processing to their advantage (heterodyne, frequency division, time expansion). In addition, as bat calls also depend on the species and the environment, their structure might vary a lot. Finally, we will see how the bat calls are analysed and how the structure is helpful for classification.

## 3.1 Echolocation principle

During the fly, a bat may encounter obstacles that are fixed or mobile. To visualise its environment, the bat emits shouts named calls. These calls are attenuated with the distance traveled. When a call hits an obstacle (prey or objects), the signal is reflected. The reflection can be modified by the structure of the object, e.g., wing, hair. When the object is a flying animal, the reflection by the body and the wings are different because of the Doppler effect. The bat compares the initial call with the reflection to obtain information from the environment.

Social calls are emitted by a bat in order to be understood by its peers. That implies that they use a code to understand each other. Social calls can be used to mark a territory or lead a group in fly. In the case of a sonar call, there is less social specificity as it is not meant to be understood by the others.

Each species of bat has a different call because the species has its own characteristics. The size has repercussions on frequencies: little bats like the Pipistrellus have a louder call than the Myotis that are taller. The physiology of the bat also has a role because some species like the Vespertilio emit calls with the mouth while others like the Horseshoe bats emit with the nose. Finally, if the place is known or if it happens during the hunt, the call will be different. For all these reasons, we can link a call to a species.

## 3.2 Types of structure

As explained in the previous point, bat calls have different structures depending on the species and the environment. You can see in the figure 3.1 the four existing structures. The first type is the FM for frequency modulation. It is a call that hits a large frequency range in a short time (see the section 3.2.1 to have more information). The second type is the NCF call, in contrast to the FM call, the call hits a very small frequency range (see the section 3.2.2 to have more information). Then, the third type is the FM-NCF which is a mixture of the two first type (see the section 3.2.3 to have more information). The last type, is the CF call only used by the Rhinolophidae species (see the section 3.2.4 to have more information).

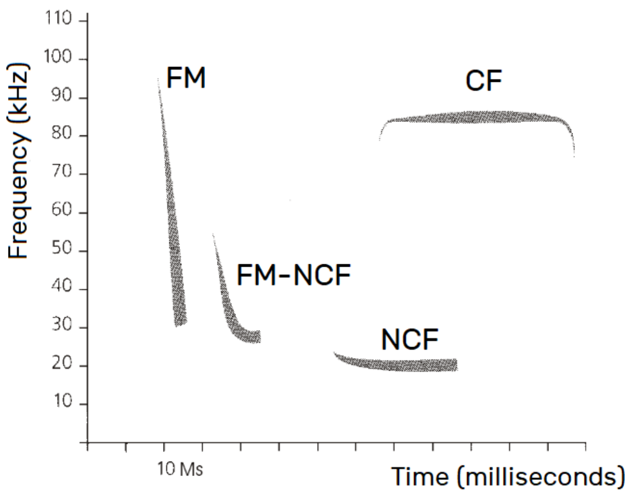


Figure 3.1: Spectrogram of different structure of bat call.

### 3.2.1 Frequency modulation (FM) call

A frequency modulation call is a signal that decreases significantly in frequencies (e.g., a Myotis myotis usually produces calls that go from 110Khz to 13Khz in 5ms). This kind of call is very expensive for the bat but it retrieves a lot of information thanks to the large range of frequencies used. High frequencies give information about the location of an object in a 2D space while low frequencies give information about the distance between the bat and the object. The lower in frequency the signal is, the more the distance is accurately estimated by the bat. This type of structure is used by forest bats to get more information about foliage and obstacles to avoid.

The slope is significant and can be linear, concave, convex or sigmoid. This kind of sound has a range of 10 meters approximately [6]. The slope depends on the goal of the call (increase the range of the signal, nature of information to get [6]).

### 3.2.2 Nearly constant frequency (NCF) call

This kind of sound is used by bats during the fly only and in an open environment to see the presence of objects. By convention the bandwidth is less than 5kHz for these calls. The range of a nearly constant frequency can be more than 100 meters. The structure of horseshoe bat call looks like a NFC call, but it is considered as a Doppler call, because it uses this technology to visualise its environment (see 3.3.4).

### 3.2.3 Hockey stick calls (FM then NCF)

Bats using this type of call are bats living in tree edges. Therefore, the call is a compromise between the FM call and the NCF call. The bat call can be divided into these two structures and the bat has a total control of the parameters of both of them (duration of the call, bandwidth, ...). All species using mostly this structure can use the previous structure depending on the needs. The name “Hockey stick calls” comes from the shape on the spectrogram.

The figure 3.2 shows how a call can evolve depending on the environment. The left part of the spectrogram shows a more congested space while the right part shows a more empty space. We can see that in a congested space, the call looks like a FM call to identify with more precision the object, while the NCF call, is used during fly to hunt or move.

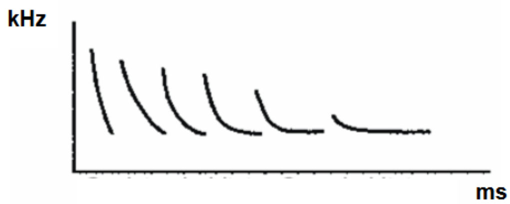


Figure 3.2: Spectrogram of the evolution of calls depending of the environment.

### 3.2.4 Doppler call or Constant Frequency (CF)

This structure is used by Horseshoe bats and is based on highlighting the movement of the objects. The call is divided into three parts. The first in a little modulated up frequency, followed by a long constant frequency and finished by a modulated low frequency. The Doppler effect is created by the long constant frequency. If the echo of the call is a high-pitched sound, it means that the bat gets closer to the object, this is created by the Doppler effect because the Doppler effect is a distortion of a wave length for a source in movement [6]. The first and the last part of the sound are used by the bat to identify the position of the object in the space.

## 3.3 Detector types

It exists different type of detectors depending on the needs:

### 3.3.1 Heterodyne

Heterodyning is a signal processing technique that creates a new signal by multiplying an input signal (e.g., coming from the air) and a signal created by an oscillator inside the system. The output signal is a beat [6] resulting of the combination of these two signals. At most these signals are close, at most the listened sound is low, and when they are equal, there is no more sound and this is called "zero beat" [6]. Detectors using this technique have a range of $\pm$ 5kHz around the referred frequency. For example, if the detector is specified for 120kHz then the trigger data are around 115 kHz and 125 kHz. Some recent detectors allow the variation of the local oscillator which is interesting for the analysis of calls that are not in constant frequency.

From a mathematical point of view, heterodyning is based on the trigonometric identity defined by

$$\sin\theta \times \sin\varphi = \frac{1}{2}\cos(\theta-\varphi) - \frac{1}{2}\cos(\varphi+\theta)$$

and adapted for sinusoidal signals with $\theta = 2\pi f_1 t$ and $\varphi = 2\pi f_2 t$ we have

$$\sin(2\pi f_1 t) \times \sin(2\pi f_2 t) = \frac{1}{2}\cos[2\pi(f_1 - f_2)t] - \frac{1}{2}\cos[2\pi(f_1 + f_2)] \tag{3.1}$$

The left side shows the modulation of a signal by an other by multiplying the sinus and the right side shows the resulting signals where the first is the difference of frequencies and the other is the sum of frequencies and they can be considered as distinct signals. One of these signals is kept and the other is removed by filtering.

The figure 3.3 shows how the heterodyne works. We have the two input signals, the first from the environment and the second from the oscillator inside the detector. These two signals are combined inside the mixer which gives in output two signals as described by the formula 3.1. Then a filter removes the unwanted frequency to keep the desired frequency.

This technique works well with the Horseshoe bats because they emit sound in constant frequency during a long time, you would have to set the detector at the frequency of the specified bat to get a good rate of detection.

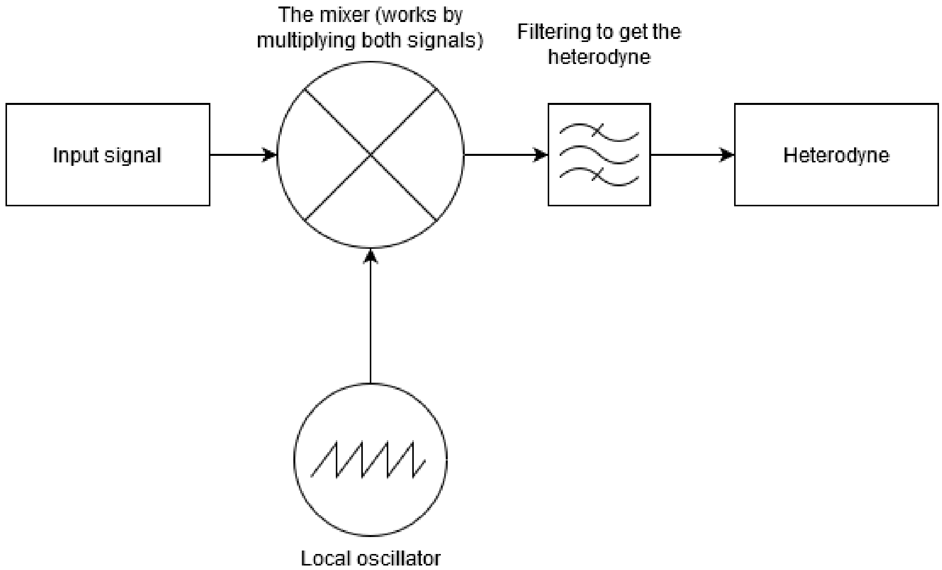


Figure 3.3: Schematic diagram of heterodyne.

### 3.3.2 Frequency division

Frequency division aims to transform bat sounds so that they can be heard by humans. To achieve this, the frequency of the call is divided by a parameter to match the range of the human ear. The human hearing acuity is situated between 20 Hz to 20 kHz in theory but in practice some people don't perceive sounds at the frequencies higher than 8 Khz and it can sometimes be complicated to specify a real range [44]. For example, a bat call around 75 000 Hz will be decreased to 7 500Hz with a parameter of 10 (this parameter is the most used inside the model of detectors by frequency division). In fact, a call containing 100 oscillations in 1 second will contain after the division only 10 during the same period.

To get that result, the computer creates the oscillogram of the call, then for every x oscillation depending of the parameter, the computer creates a unique oscillation in the output call. With this technique, the duration and the emit rhythm are not modified, but the structure is impacted and the whole acoustic domain can be accessed.

### 3.3.3 Time expansion

The principle of the time expansion is to digitalise a sound and play it at a slower speed by decreasing the sampling rate. The sampling rate is equal to the number of data saved by second. By default, the sampling rate must be equal to the double of the frequency to be listened by the detector to avoid aliasing. Thanks to this, the detector will save enough data to represent only one possible signal [43].

Like the heterodyne analysis, the time expansion analysis works by analysing the evolution of the frequencies over time, but the criteria are not all the same. There is no zero beat in time expansion, but it is replaced by the bandwidth and the bandheight. The sonority is used instead of the tone (multitoneal for Horseshoe bat, nasal for Plecotus and Barbastella, and finally whistled for the others). In addition of these criteria, there are new ones like the duration and the energy distribution that gives information about the structure.

### 3.3.4 Comparative table

The detectors can be compared on different characteristics depending on your need. If you need a detector to analyse the sound in a close bandwidth without spurious sounds, you need to use a heterodyne detector. On the other hand, if you need to analyse all the frequencies in real time by accepting a loss of detail then you need to use a frequency division detector. And lastly, if you need the full details sounds without it being in real time you need to use a time-expansion detector. All the information can be found in the table 3.1.

| | Heterodyne | Frequency division | Time expansion |
|---|---|---|---|
| Bandwidth | close to the local oscillator | All | All |
| One-way process | Yes | Yes | No |
| Good for | Constant frequency (Horseshoe bat) | Nearly constant frequency Hockey stick calls | all |
| Bad for | Frequency Modulation Hockey stick calls | Frequency Modulation | |
| Advantages | No spurious sounds | Real time | Full sound details |
| Limitation | close bandwidth | Lack of detail (caused by the reduction of frequencies) | Not in real time |

Table 3.1: Comparative table for the different detectors.

## 3.4 Computer analysis

With the computer analysis, you can analyse a call depending on the three possible representations of a sound, these representations will be introduced later. Also, by combining the information you get from the three representations, you can find the species of a bat.

### 3.4.1 Harmonic

The sound is created by a succession of closing and opening of the vocal folds generating vibrations in the air during exhalations. It is composed of a fundamental tone (the first harmonic) and overtone (all next harmonics). The overtones appear on several of the fundamental tone frequency. When the sound is inside the pharynx and the nasal cavity for the Rhinolophus, Plecotus and Barbastella or inside the oral cavity for the Vespertillio, Myotis and Molossidae, it will be modified. The cavity will distribute the intensity on one or multiple harmonics. On a spectrogram, these overtones look like the fundamental, however they have a slope of the fundamental tone multiplied by its harmonic number. Only the fundamental tones must be analysed to classify bats.

### 3.4.2 Spectrogram and sonogram

The first representation is the spectrogram. It is the evolution of the frequencies (ordinate) over the time (abscissa). It allows to find out what the type of structure of the call is. The first thing to do is to harmonise the temporal window approximately to 50 ms. Thanks to this, the structure is correctly drawn on the screen. If not, you cannot see the difference between a frequency modulation steep and flattened. The temporal window is linked to the Fast Fourier Transform, the narrower it is, better the resolution in duration it is, but at the expense of the resolution of frequencies. Depending on the structure of the call, you must choose a narrow windows (for FM) or wide windows (for NCF).

The sonogram is a variant of a spectrogram containing a third axis that represents the intensity power of the call.

### 3.4.3 Oscillogram

The oscillogram is the second possible representation. It gives information about the variation of the intensity over time. This view is limited because it depends on the quality of the record, background noises impact the oscillogram even when the noises are not in the frequency range [6]. But the oscillogram is important to find the species when the criteria of the intensity is needed to find the species.

The oscillogram is important to identify the energy distribution during a call because some species from the same family can only be classified with this process e.g., Myotis. As shown in the figure 3.4, the energy distribution can be classified in three ways. The first is a progressive start when the energy grow up for several milliseconds. The second distribution is the explosive start when the energy grow up quickly (< 1 millisecond). The last distribution, the sinusoidal modulation, indicates that the call was reflected on a water surface.

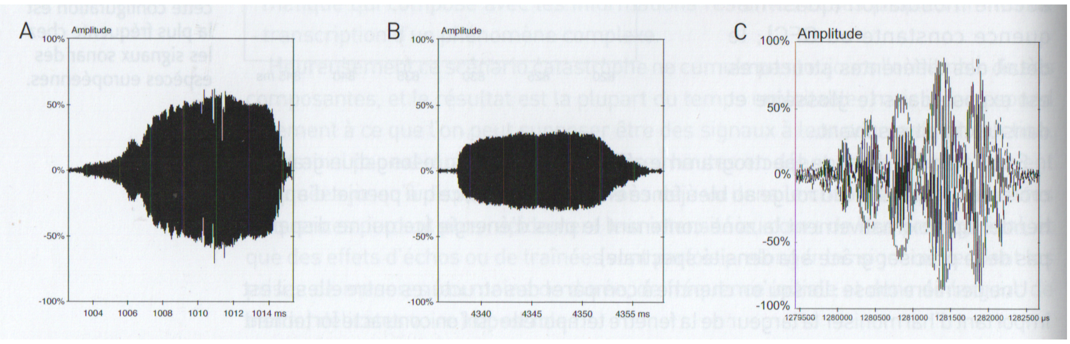


Figure 3.4: Oscillogram of different energy distribution. A: progressive start, B: explosive start, C: sinusoidal modulation [6].

### 3.4.4 Spectral density

The spectral density represents the evolution of the intensity over the frequency range, the time notion is not present in this representation. The abscissa refers to the frequencies and the ordinate refers to the decibel. The decibel is an anthropomorphic unity representing the ratio of the powers between the measured quantity and the reference value of a sound that cannot be heard by humans. Because of anthropomorphism, the values in ordinate are directly linked to the quality of the audio. For example, if you record a sound next to the detector, the value in decibel will be bigger than if the sound source is 15 meters away from the detector. Therefore, you have to analyse the amplitude peaks or domes related to the abscissa to get all the possible information. The initial frequency is found at the end of the peak (or dome) while the final frequency is found at the start of the peak (or dome). The maximum energy frequency is found at the highest point of the peak (or dome).

Nearly constant frequency and Doppler calls have peaks on the spectral density because the calls are around a frequency while the others calls create domes on the spectral density because they hit a bigger frequency range.

## 3.5 Analysis & classification

### 3.5.1 Heterodyne Analysis

For the analysis, the value of the detected call is the value of the zero beat. Heterodyning deforms the sound, but it can give a lot of information about the call. The criteria to be used for identification are the number of calls per second, the frequency of the zero beat, the tone of the call (by onomatopoeia: “tic”, “poét”, "top”, “tchoc”, ...), the regularity and finally the intensity in the case where the distance between the bat and the microphone is known. Species can be classified into four groups depending of the structure of the call, then with the number of calls per second and the frequency it is possible to identify a species. One has to be careful because some species cannot be identified with this technique (e.g., Myotis mystacinus, Myotis brandtii) [41]. For some calls, an other criteria is also needed: it is the maximum energy frequency (MEF), but it is calculated by spectral density so you need another software to obtain it.

The table 3.2 gives the species related to the information obtained by the analysis of the heterodyne. For the CF and NCF based on the MEF groups, only two species can get mixed up [6], but for the FM group, there are a lot of species corresponding to the 25 to 90 kHz, because a lot of calls do not contain a zero beat, so there is a great risk of confusion. For the NCF and FM-NCF group, all the criteria are needed to find the species related to the call because of the overlapping of the zero beats.

This analysis is hard to establish because the person must train his or her ears a lot to the different tones and the different intensity and that takes time.

| **Groups** | **Zero beat** | **Species** |
|---|---|---|
| CF | 76 to 85 kHz<br>92 to 98 kHz<br>100 to 105 kHz (confusion)<br>104 to 111 kHz (confusion)<br>107 to 116 kHz | Rhinolophus Ferrumequinum<br>Rhinolophus blasii<br>Rhinolophus euryale<br>Rhinolophus mehelyi<br>Rhinolophus hipposideros |
| NCF based on the MEF | 13 to 21 kHz (confusion)<br>18 to 21 kHz (confusion)<br>21 to 25 kHz | Nyctalus lasiopterus<br>Nyctalus noctula<br>Nyctalus Leisleri |
| NCF and FM-NCF | 9 to 13 kHz NCF<br>21 to 25 kHz NCF irregular rhythm<br>22 to 27 kHz NCF regular rhythm<br>23 to 27 kHz FM-NCF irregular rhythm<br>27 to 29 kHz NCF QFC irregular rhythm<br>30 to 35 kHz NCF<br>32 to 37 kHz FM-NCF<br>34 to 38.5 kHz NCF<br>37 to 40 kHz FM-NCF simple social call<br>38.5 to 42.5 kHz NCF doubled social call<br>41 to 47 kHz NCF<br>44 to 50 kHz FM-NCF<br>47 to 52 NCF fast rhythm<br>50 to 55 FM-NCF<br>50 to 57 NCF<br>52 to 58 kHz FM-NCF | Tadarida teniotis<br>Eptesicus isabellinus<br>Vespertilio murinus<br>Eptesicus serotinus<br>Eptesicus nilssonii<br>Hypsuga savii<br>Myotis dasycneme<br>Pipistrellus kuhlii<br>Pipistrellus kuhlii<br>Pipistrellus nathusii<br>Pipistrellus nathusii<br>Pipistrellus pipistrellus<br>Pipistrellus pipistrellus<br>Miniopterus schreibersii<br>Miniopterus schreibersii<br>Pipistrellus pygmaeus |
| FM | 25 to 28 kHz regular rhythm<br>25 to 90 kHz<br>41 to 43 kHz with a lot call with different sounds | Myotis myotis<br>oxygnathus<br>Plecotus sp.<br>Myotis sp.<br>Barbastella barbastellus<br>other species with NCF call<br>Barbastella barbastellus |

Table 3.2: Criteria to the species.

### 3.5.2 Frequency division analysis

Because of the reduction in frequency and the loss of quality from this type of detector (the call can sound thin and quiet), this analysis gives bad results compared to heterodyne or time expansion detector [17]. So it is not advisable to use it.

### 3.5.3 Time expansion analysis

As the time expansion is not in real time which is necessary to do a computer analysis to get the different criteria, it is very hard to do it during a walk.

The figure 3.5 is the flowchart you follow to find the species of a bat call from a time expanded sound using computer analysis. The first step is to use the spectrogram to get the duration and the structure of the call. Then, you have to use the spectral density to find the initial, final frequency, the maximum energy frequency and the spectrogram or the oscillogram to get the duration. In the case of multiple calls, the duration of interval between calls is an information to take. If the structure is a frequency modulation, it is important to analyse the sonority of the call and to use the oscillogram to get the type of the primer (explosive or progressive). Finally, once you get all these information, you have to use references such as the tables in the Barataud book [6], and look inside the table related to the structure and the MEF of the call and search for the species that respects all the other criteria you calculate from the call. The table looks like 3.3, this table is for the FM structure with a nasal sonority and explosive primer used for the example at the section 3.5.4.

| **Species** | **Duration of intervals (ms)** | | **Call duration (ms)** | | **Initial Frequency (kHz)** | | **Final Frequency (kHz)** | | **MEF (kHz)** | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **AVG** | **SD** | **AVG** | **SD** | **AVG** | **SD** | **AVG** | **SD** | **AVG** | **SD** |
| **P. auritus** | 96.3 | 62.6 | 3.4 | 1.6 | 50.6 | 4.8 | 23.0 | 3.8 | 57.9 | 6.7 |
| **P. austriacus** | 104.1 | 81.2 | 3.3 | 1.8 | 42.1 | 3.6 | 22.4 | 2.4 | 53.8 | 3.6 |
| **P. macrobullaris** | 100.7 | 47.3 | 3.8 | 1.5 | 45.9 | 3.6 | 20.6 | 3.2 | 53.9 | 7.4 |

Table 3.3: Species related to the criteria.

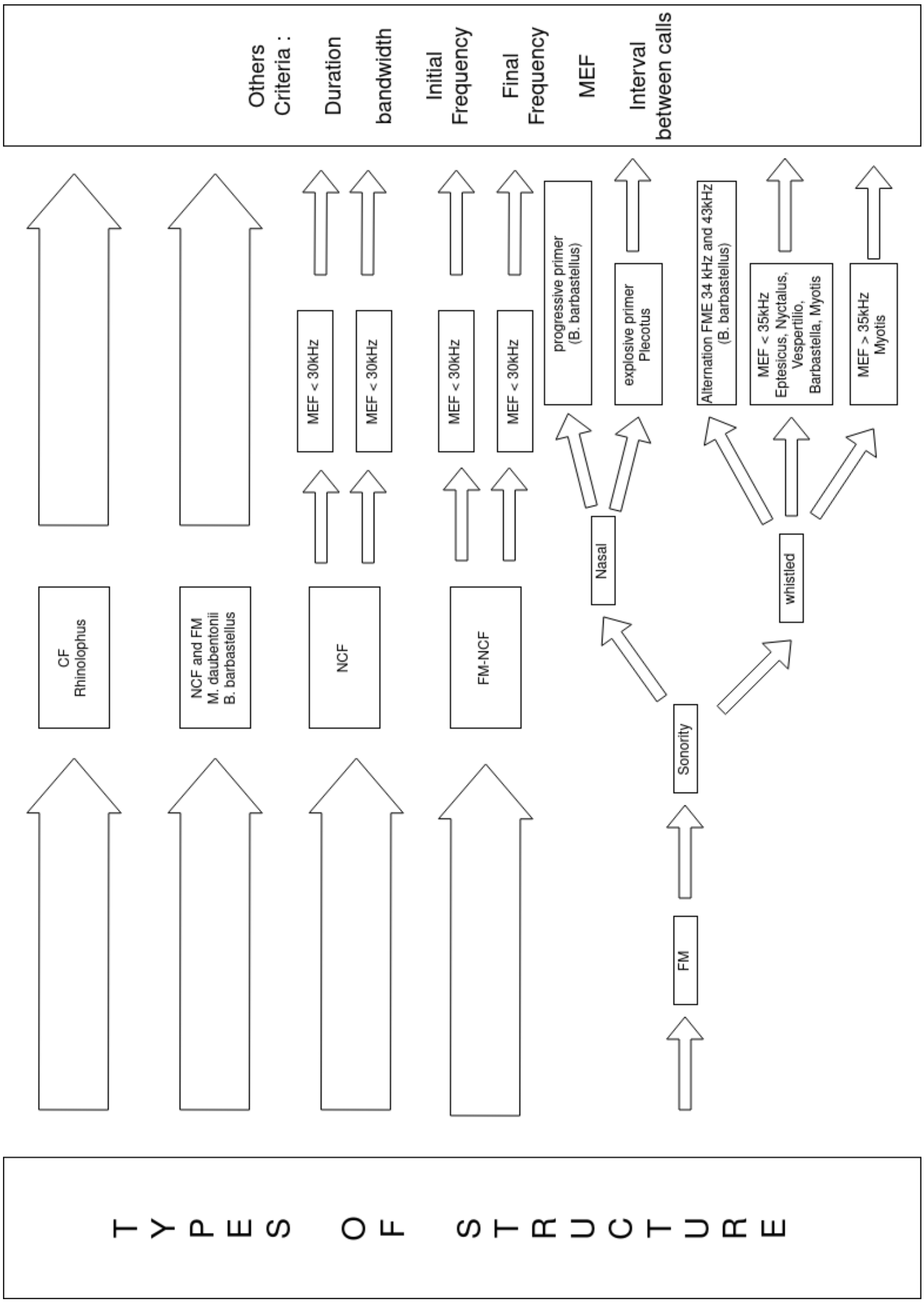


Figure 3.5: Flowchart for the time expansion analysis.

### 3.5.4 Example of analyse with time expansion analysis

The figure 3.6 is an example of a call during a prey capture. You can see the three different representations of the same call. With the oscillogram, we know that the call is an explosive start because the energy increases in less than one second, the spectral density gives a MEF near 64 kHz and the spectrogram shows a frequency modulation structure. The initial frequency is near 50 kHz and the final frequency near 28 kHz. You have to be careful because the MEF is based on the most powerful harmonic and that is why the MEF is higher than the initial frequency. The duration is lower than 1 second. With the figure 3.5 and if we listen to the call linked to the publication, we listen a nasal sonority. With all of this information, we can look in the table 3.3 given by the book Barataud [6] to identify the species and it is the Plecotus auritus who respects the most the different criteria which is confirmed by the publication [19].

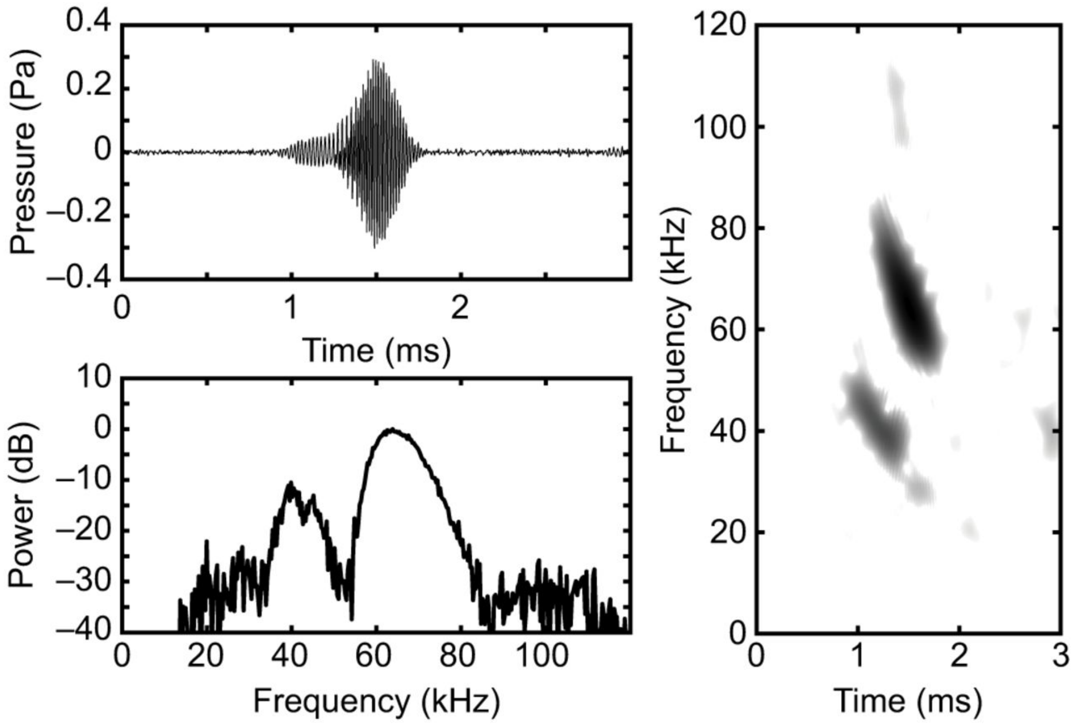


Figure 3.6: Oscillogram (top left), spectral density (bottom left) and spectrogram (right) of an exemplary call emitted during prey capture in the wind tunnel [19].

## 3.6 Conclusion

This chapter shows that each species of bat has its own call but which can be similar. Furthermore, depending on its environment a species can switch from a type of call to an other, increasing even more the difficulty in recognizing a species if we do not know the environment is which the call is caught.

Another important information of this chapter is that the choice of the detector is very important because the recording of the call will be different and has to be analysed differently if we want to identify the species.

Finally, the manual analysis or the computer assisted analysis requires a lot of knowledge, a fine and trained hearing and a great concentration because an imprecision in measurement can cause a bad classification. This is why an efficient supervised learning solution to classify bats is required to help people during the process.

# Chapter 4

# Supervised learning techniques

This chapter introduces previous works and covers several supervised learning techniques for classification (discriminant function analysis, random forests, support vector machines). Then, a discussion is held on the technique chosen for the Batmen project.

## 4.1 First approaches

With the significant amount of bioacoustic data that can be recorded and stored nowadays, recent researches considered the use of supervised learning techniques to provide a way to classify different animal species. In particular, David W. Armitage and Holly K. Ober uses random forests, support vector machines, artificial neural networks, and discriminant function analysis on frequency division data [5]. On the other hand, Nuno Cláudio [13] also experimented with Extreme Gradient Boosting [11]. Both researches were targeting bats.

Before we discuss the effectiveness of these techniques and give a decision on the one used for Batmen, we will give a short introduction of the most used techniques in the field in the next sections.

## 4.2 Discriminant function analysis (DFA)

Discriminant function analysis (DFA) is a statistical method that builds a model using one (for only two classes) or more combinations of predictor variables (or features) that best discriminate between the classes. Discriminant functions are based on these combinations and ranked by significance. They are used on newer cases that provide these variables but have no class membership. In our case, classification can be achieved using the maximum likelihood estimation.

Nevertheless, this method implies some assumptions:

- The cases should belong to only one group
- There should be more samples than there are independent variables
- The variables should follow a normal distribution
- There should be no outlier (distant from the other) in the data
- The cases should be independent

## 4.3 Random forests (RF)

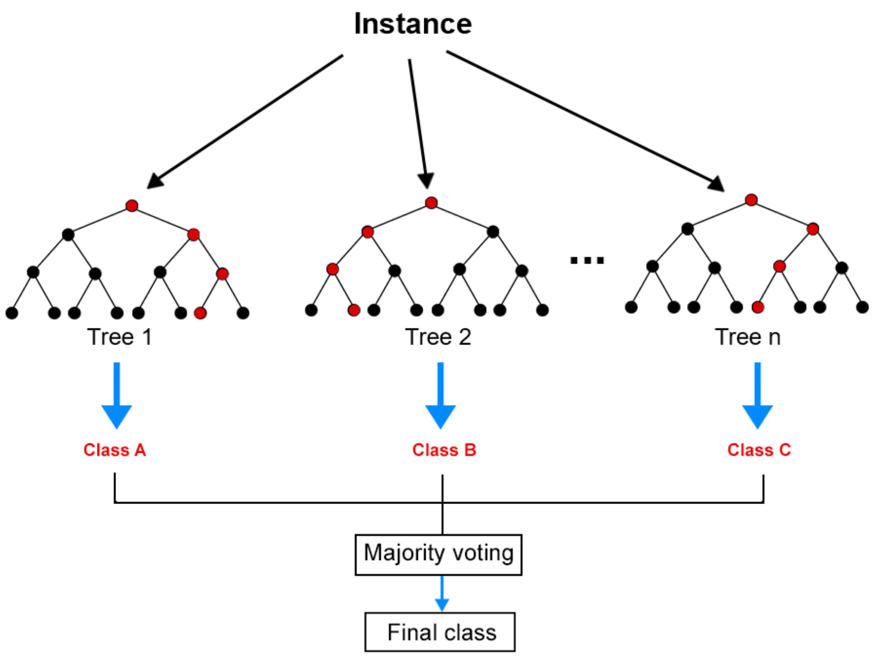


Figure 4.1: Random forest example.

Random forest (RF) [9] is an ensemble learning method that builds several decision trees at training time. A decision tree is a structure composed of nodes and directed edges where questions related to the features are asked at each decision until it leads to an actual class label. In the end, the model's prediction will be the most voted class label amongst every individual tree as seen in the figure 4.1.

Random forests also uses Bootstrap aggregation (or Bagging) to reduce variance of the model: each tree is trained on random samples of the training set with replacement. The dataset is split in

2/3 of the data for training and 1/3 for the testing (“out-of-bag”). Moreover, a random subset of features is also selected at each split. Making these changes greatly reduces the high correlation between trees and consequently making them less sensitive to the actual training set.

Finally, random forests can handle large, imbalanced or sometimes faulty data sets very well with good accuracy. The major drawback of this method is the huge computational and memory requirements when a large number of deep trees are created.

## 4.4 Support vector machines (SVM)

Support vector machines (SVM) [14] are models (originally for binary classification) that constructs a N-dimension (for N features) hyperplane that best separates data points into classes. This is achieved by finding the maximum margin between the closest data points of each classes (support vectors).

If the problem is not a binary classification problem, two additional methods can be used:

- One-against-all: $k$ SVM models are built for the $k$ classes. Each model is trained with data from one class as positive labels and every other class as negative.
- One-against-one: $k(k-1)/2$ SVM models are built as binary classifications for each pair of classes.

In the first case, the class that classifies test data with the best margin. In the second case, it would be the class selected by the most classifiers.

The support vector machines are really efficient on higher dimensions and when classes are clearly separated. On the other hand, it does not handle overlapping classes and larger data sets.

## 4.5 Discussion

It is important to note that some parameters can make the comparison difficult. In fact, different bat species are found all over the globe and their populations can vary significantly. The detection technique also plays a role in the data that will be available, as well as the involvement of the researchers. These parameters lead to data of different quality and quantity. We also have seen that each approach has its own constraints and methods. In the end, we chose to believe that random forest was the best technique overall as it was the most accurate and quick to train [5].

Nevertheless, a project named Bat detective [25] showed that convolutional neural network could outperform the previous techniques. In fact, we decided to try this approach as the implementation of Bat detective is open-source and available for research purpose. The system provided seemed scalable and flexible enough to handle different datasets through minor processing.

# Chapter 5

# State of the art

This chapter describes a research article published in the PLOS Biology journal. The article provides an in-depth insight on the conception of deep learning tools for bat signal acoustics. The process of data collection of European bat audio files is described and an overview of the neural network architecture that was built for the project is given. In addition, the algorithms, the different libraries and training parameters are described. Finally, a discussion is held on the performance comparison of the given algorithms considered in the paper.

## 5.1 Presentation

Bat detective is a set of deep learning tools for bat acoustic signal detection. This project was lead by the University College London, the Zoological Society of London, the Bat Conservation Trust, Batlife Europe and the University of Auckland. The need for efficient, scalable, noise-resistant, open-source bat search-phase (used to detect its preys) echolocation call detection tools has lead the researchers to design several machine learning algorithms, in particular convolutional neural networks, and compare them to the current state of the art.

## 5.2 Convolutional neural networks

Before we go through the different aspects of the project, it is important to understand the basics behind convolutional neural networks.

Convolutional neural networks [24] or CNNs are networks specialised in processing data in the form of arrays such as sequences or signals which are 1D, images and audio spectrograms which are 2D and finally, videos for the 3D.

Essentially, the CNN is inspired by the connectivity pattern between neurons inside the human brain, specifically the visual cortex. In terms of architecture, the multilayer network can be separated in two parts: the feature extraction through convolution, pooling operations and the classification part with fully connected layers.

### 5.2.1 Convolution layer

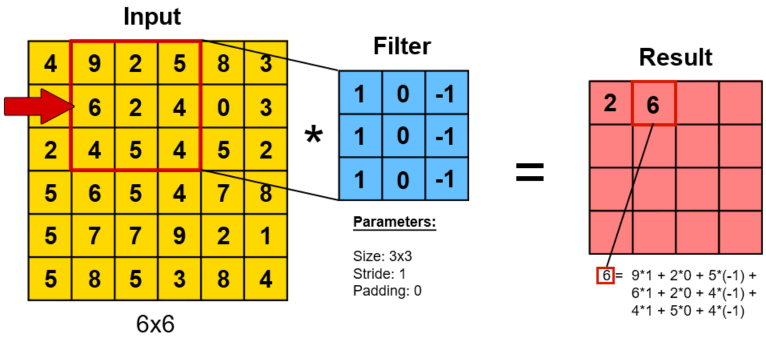


Figure 5.1: Dot operation between the input and the filter.

A convolution operation, as seen in the figure 5.1, involves a sliding dot operation between the input data and a filter (or kernel) that produces a feature map and passes it to the next layer. The sliding operation is dictated by the stride parameter which is the amount by which the filter shifts through the input. The padding parameter is also really important if you want to preserve a certain amount of information and specific dimensions: zero-padding times the padding value is added around the input.

### 5.2.2 Pooling layer

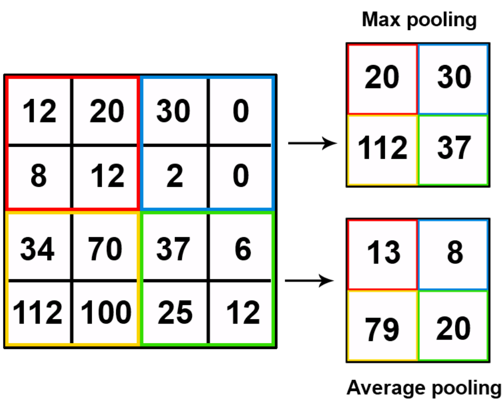


Figure 5.2: Max pooling and average pooling with a 2x2 filter.

The pooling operation is similar to convolution as a filter is applied to the input and uses the stride parameter. This time, an average operation or a maximum operation (e.g., figure 5.2) is performed thus reducing the spatial dimensions of the input. The purpose of this operation is to limit the computation cost and reduce overfitting.

### 5.2.3 Activation functions

An activation function is another building block of the neural network. The function is applied to each neuron and decides if the neuron should fire or not, it introduces nonlinearity. The output of the function is then sent to the next layer.

The most commonly used activation functions are:

- ReLU (or rectified linear unit): $f(x) = \max(0,x)$ which is the most commonly used activation function since it's so computationally efficient and it deals well with the vanishing gradient problem (the inability for the network to propagate useful gradient information thus limiting the network's training).
- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$ which allows the normalization of the outputs between 0 and 1 and tends to be a great classifier.
- Softmax: $f(x_i) = \frac{e^{x_i}}{\sum_{k=1} Ke^{x_k}}$ which allows the normalization of the outputs between 0 and 1 for each class in a multi-class problem. It also helps classify inputs into classes by giving their probability of belonging to the classes.

### 5.2.4 Fully connected layer

The fully connected layer is the classification part of the network: it takes the output from the last pooling layer and flattens it into a vector. The neurons are multiplied by weights and a ReLU activation function before passing its output to the output layer that will determine the classification using typically a Softmax activation function.

## 5.3 Data

Bat detective was trained on a subset of a full-spectrum time-expanded (with factor 10) ultrasonic acoustic data collected between 2005 and 2011. The data were provided by scientists belonging to the Indicator Bats Programme (iBats) [39]. The individual 3.84 seconds recordings were later uploaded to the Zooniverse [45] platform for the scientists to annotate them. A subset of the 3,364 files provided by the top performing annotating user was selected as the primary dataset as the authors decided to settle for a train test split. Each file is labelled with a sequence of zeros and ones where ones appear when a call is present, zeros when it is not.

Here is a summary of the datasets used for the project:

| **Location/Name** | **Period** | **Device** | **Training set files** | **Test set files** |
|---|---|---|---|---|
| Bulgaria & Romania | 2006-2011 | Tranquility Transect (Courtplan Design Ltd, UK) | 2,812 | 500 |
| Norfolk (UK) | 2015 | SM2BAT+ Song Meter (Wildlife Acoustics) | 2,812 | 500 |
| UK | 2005–2011 | SM2BAT+ Song Meter (Wildlife Acoustics) | 2,812 | 434 |

Table 5.1: Bat detective dataset information.

| **Location/Name** | **# training calls** | **# testing calls** |
|---|---|---|
| Bulgaria & Romania | 4,782 | 842 |
| Norfolk, UK | 4,782 | 1,345 |
| UK | 5,624 | 842 |

Table 5.2: Bat detective individual calls.

## 5.4 Method

The algorithm consists in four different steps.

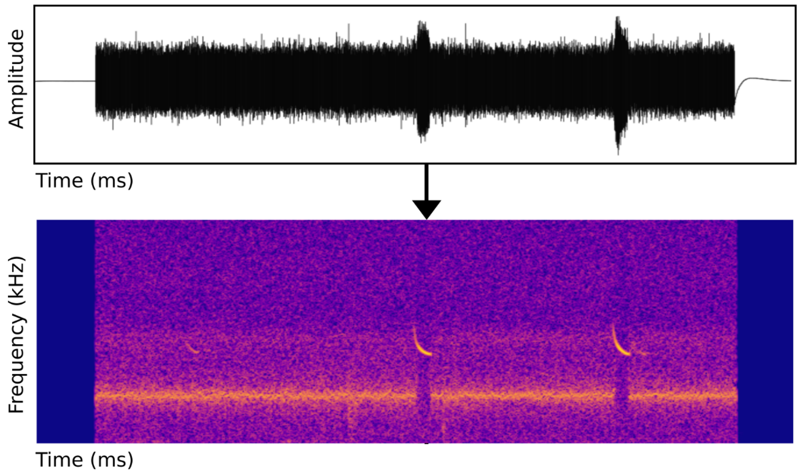


Figure 5.3: Fast Fourier transform [25].

To begin with, raw audio files are converted into log magnitude spectrograms using Fast Fourier Transform (FFT window size 2.3 milliseconds, overlap of 75%), retaining the frequency bands between 5kHz and 135kHz. An example transformation can be seen in figure 5.3.

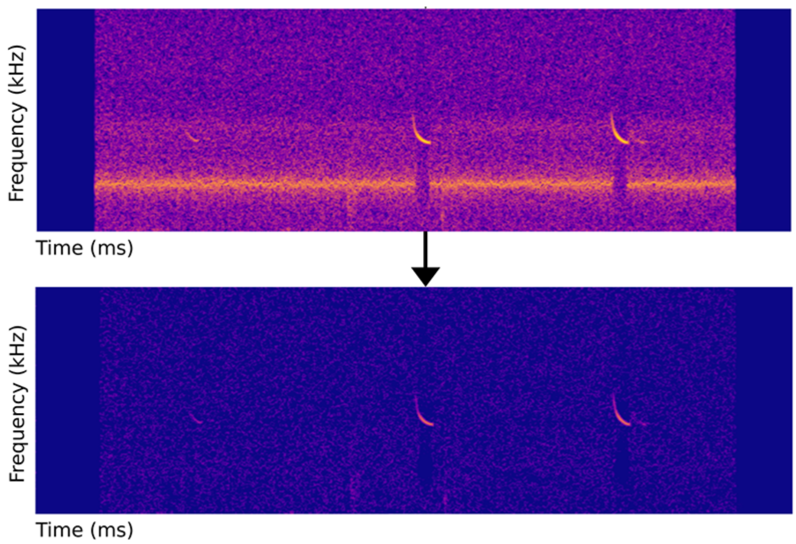


Figure 5.4: Denoising example [25].

Since the audio files might contain a large amount of unwanted background noise (mostly due to human activities), a denoising technique [3] was applied to the frequency-time matrix from the previous step. The mean intensity value in each frequency band is used to generate a threshold: any input signals over this threshold (or a certain percentage over the threshold) are kept as they will represent an acoustic event. The figure 5.4 shows the same spectrogram before (above) and after the denoising technique is applied (below), eliminating the background noise but keeping bat calls intact.

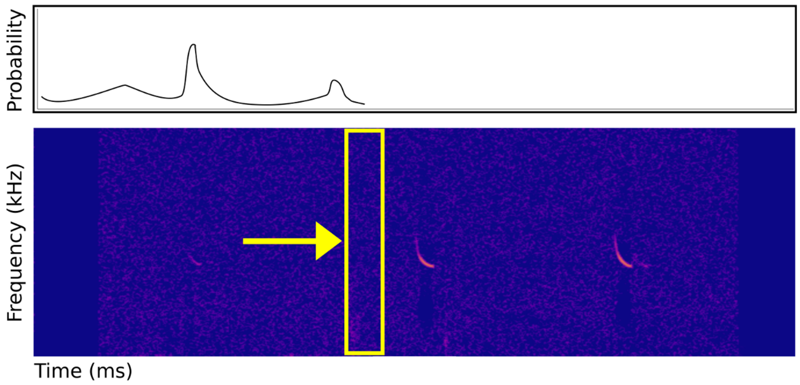


Figure 5.5: Sliding window [25].

The spectrograms are reduced in size by half, giving to the neural network an input of size 130 (frequency bins) by 20 time steps. The spectrograms are read through by a sliding window of size 23 ms predicting at each instance the presence of calls. In the figure 5.5, the yellow rectangle represents a window and shows the variation in probability throughout the process.

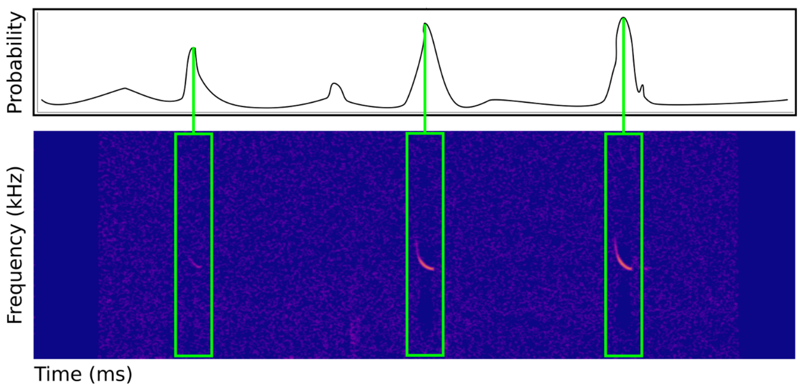


Figure 5.6: Non-maximum suppression [25].

Finally, the non-maximum suppression technique is applied to the predictions. Finding a local maximum at each peak in prediction corresponds to finding the precise time location for the start of the calls. The figure 5.6 shows the acquisition of three bat calls depicted by the green lines.

## 5.5 Architectures

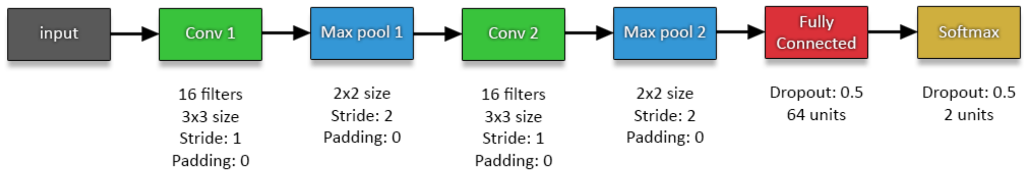


Figure 5.7: Small neural network architecture.

Bat detective provides two versions of a convolutional neural network. The smaller network depicted in figure 5.7 consists in two convolution layers with 3x3 filters followed by ReLU and two 2x2 maximum pooling layers (reducing the dimensions by half). A dropout with probability of 0.5 is applied in the 64 units fully connected layer. Finally, the same dropout is applied in a 2-unit softmax layer.

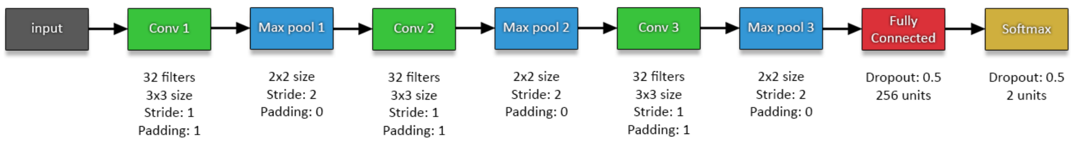


Figure 5.8: Big neural network architecture.

The bigger neural network depicted in figure 5.8 is quite similar in terms of architecture except that instead of a set of two convolutions plus ReLU and pooling, there are three. Also, the number of filters is upped to 32 and the padding to 1 for the convolutional layers. Finally, fully connected layer is upped to 256 units.

## 5.6 Output processing

As mentioned in section 5.4, Bat detective uses the non-maximum suppression technique (NMS) to obtain the precise position of a call. Non-maximum suppression is a widely used preprocessing technique in Computer vision applications that was introduced by John F. Canny with his work on the Canny edge detector [10]. The idea is to thin the edges of an image by eliminating points that are not local maxima as these local maxima indicate the largest change in intensity value. This technique is generally applied after noise filtering techniques, e.g., Gaussian filters, as edges are blurred.

As for Bat detective, the same concept is applied to the output predictions coming from the neural network. In the same manner, the one-dimensional vector containing the predictions of the detected calls is smoothed by a Gaussian filter. This filter removes noise, detail through convolution. The 1D Gaussian distribution is given in the form of:

$$g(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

where $\sigma$ is the standard deviation of the distribution.

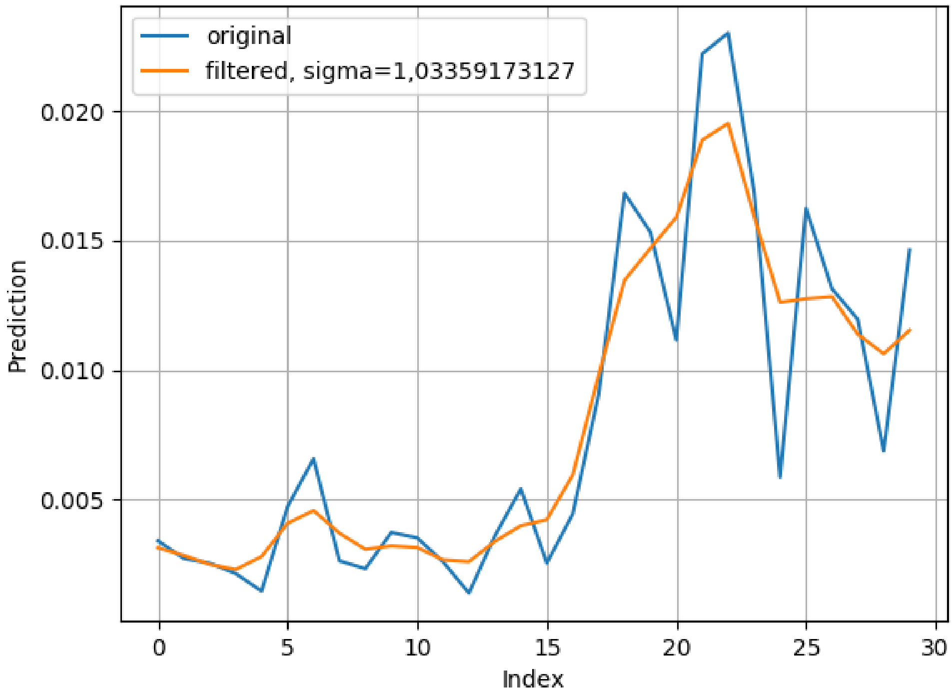


Figure 5.9: 1D Gaussian filtering example for the thirty first indexes.

The figure 5.9 shows an example of 1D Gaussian filtering on the thirty first elements of a vector of predictions.

After the Gaussian filter, the non-maximum suppression is applied to the 1D vector of probability. As we have seen earlier, NMS looks for local maxima and tries to isolate them. In this case, local maxima are found in each window of a given size inside the vector as depicted in 5.6.

Moreover, the authors chose to implement the algorithm in Cython rather than in Python. Cython allows more efficient code as it generates efficient C code and gives better performance.

## 5.7 Implementation

The project is implemented in Python 2.7. It obviously requires several other libraries for the network's training and other mathematical computations:

### 5.7.1 Theano

Theano [38] is a library that helps define, optimise and evaluate mathematical expressions involving multi-dimensional arrays efficiently. The major advantages of this library are its transparent use of the GPU for computations and its integration of numpy arrays in Theano-compiled functions.

### 5.7.2 Lasagne

Lasagne [15] is a lightweight library that allows the building and training of neural networks in Theano. The library supports the training on GPU but only on NVIDIA GPUs with CUDA support. In particular, the package layers contains the implementation of layers and helper functions for the neural network.

### 5.7.3 NumPy

NumPy [30] is the most important library for the creation of N-dimensional array objects and the ability to operate on them.

### 5.7.4 Cython

Cython [8] is an optimising static compiler for both the Python language and the Cython language (superset of Python that supports calling C functions, declaring C types on variables and class attributes). The compiler then generates very efficient C code from Cython code.

### 5.7.5 Scipy

The Scipy [42] library provides a collection of mathematical algorithms and convenience built on NumPy. Inside the project, two packages are used:

- ndimage: functions for N-dimensional image processing (manipulate, add filters to spectrograms such as a gaussian filter)
- io.wavfile: reading .wav files to numpy arrays, writing numpy arrays to .wav files

### 5.7.6 Scikit-image

Scikit-image [40] also provides algorithms for image processing. The zoom and view_as_windows provide a way to cut and calibrate the spectrograms before feeding them to the neural network.

## 5.8 Training

### 5.8.1 Hyperparameters

The authors also use the following hyperparameters:

- Learning rate: 0.01
- Momentum: 0.9
- Batch size: 256
- Epochs: 50

The algorithm also uses gradient descent optimization in the form of Nesterov Accelerated Gradient [27].

### 5.8.2 Hard negative mining

In general, the hard negative mining [16] consists in training the model with a initial subset of negative examples. The incorrectly classified, high confidence negatives examples (= false positives) are re-added as hard negatives in the next training set and this process can be repeated several times. This helps increase the performance of the model but on the other hand, it requires more resources as you are training several times with bigger training sets.

In Bat detective, the model is trained a first time with the training data. Then, hard negative mining can be applied for $n$ rounds. At each round, using the model, the algorithm gets the predicted probabilities and positions of the calls. Then, for each of these positions, the probability is compared to a given detection threshold and every prediction over the threshold is kept. The positions are then compared to the ground truth by checking the distance between them. The predictions outside of those distances are candidates hard negatives as they appear as false positives. Finally, these candidates are re-added as negative examples (i.e., labelled 0) in the training set before training the model again.

## 5.9 Performance comparison

As discussed in chapter 4, Bat detective has shown that both convolutional neural networks are superior in comparison of other supervised learning techniques (segment [23] and random forest). They also compared their performances with commercial systems such as SCAN'R, SonoBat and Kaleidoscope.

The comparison is done using the measures precision and recall. The precision represents the accuracy of the model on the predicted positives and the recall that represents the number of predicted true positives in comparison of the actual number of true positives.

$$\textbf{Precision} = \frac{True\ positive}{False\ positive + True\ positive}$$

$$\textbf{Recall} = \frac{True\ positive}{False\ negative + True\ positive}$$

The precision-recall curve in the figure 5.10 lets us analyse the trade off between precision and recall at different thresholds. The threshold determines the probabilities of call detection taken into account and gives a new precision-recall value pair each time. We can see that as the threshold decreases, the precision decreases and the recall increases. In fact, this would increase the number of false positives and thus decrease the precision and on the other hand, we accept more true positive examples thus increasing the recall. We observe that the bigger neural network has a better precision-recall curve as it it is the closest to the upper right (the perfect curve), meaning that it has the better precision-recall values.

In addition, the figure 5.11 gives a numeric comparison of the different algorithms. It is obvious that both convolutional neural networks outperform the other algorithms overall. In general, other algorithms show low recall rates (high false negatives) except for segment which shows high recall rates but low precision rates (high false positives). Interestingly, we note that the smaller neural network outperforms the bigger one on the Norfolk dataset.

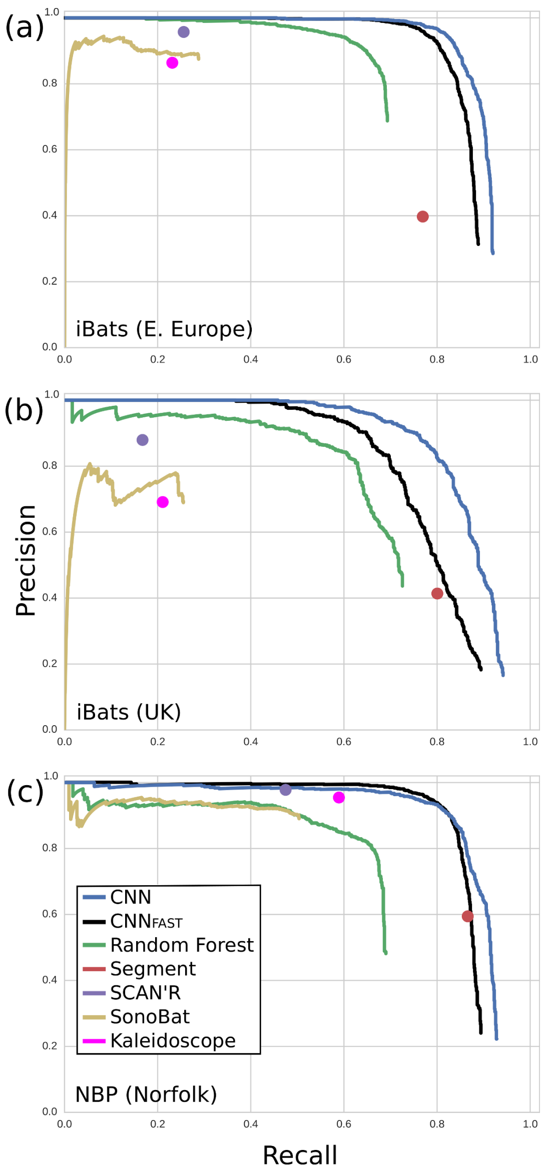


Figure 5.10: Precision-recall curves for the different algorithms [25].

| Detection Algorithms | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | BatDetect | | | |
| Average Precision | SonoBat | SCAN'R | Kaleidoscope | Segment | Random Forest | $CNN_{FAST}$ | $CNN_{FULL}$ |
| iBats (R&B) | 0.265 | 0.239 | 0.189 | 0.299 | 0.674 | 0.863 | 0.895 |
| iBats (UK) | 0.200 | 0.142 | 0.144 | 0.324 | 0.648 | 0.781 | 0.866 |
| NBP (Norfolk) | 0.473 | 0.456 | 0.553 | 0.506 | 0.630 | 0.861 | 0.882 |
| **Recall at 0.95** | | | | | | | |
| iBats (R&B) | 0 | 0.251 | 0 | 0 | 0.568 | 0.777 | 0.818 |
| iBats (UK) | 0 | 0 | 0 | 0 | 0.324 | 0.570 | 0.670 |
| NBP (Norfolk) | 0.184 | 0.470 | 0 | 0 | 0.049 | 0.781 | 0.754 |

Large numbers indicate better performance. Recall results are reported at 0.95 precision, where zero indicates that the detector algorithm was unable to achieve a precision greater than 0.95 at any recall level. The results for the best performing algorithm are underlined. Details of the test datasets and detection algorithms are given in the text.

https://doi.org/10.1371/journal.pcbi.1005995.t001

Figure 5.11: Performance comparison of algorithms [25].

## 5.10 Conclusion

Bat detective is a project implementing several learning techniques in Python 2.7 that help the detection of bat calls given bat calls audio files. The datasets were created all over Europe between 2005 and 2011 and labelled by expert volunteers. The audio files are transformed into spectrograms using Fast Fourier Transform and a denoising algorithm is applied to it. The most accurate technique, convolutional neural network, goes through the spectrogram with a sliding window and makes predictions on the detection of calls. Also, the output prediction goes through Gaussian filtering and non-maximum suppression. Moreover, hard negative mining is applied during training to increase the performance. Finally, the results show that convolutional neural networks outperforms other techniques such as random forest and commercials detection systems.

# Chapter 6

# Data collection

This chapter describes how the dataset for Batmen was created from the labelled data provided by Natagora. Also, the rejection and the acceptance of specific files is described inside this chapter. Furthermore, an explanation on the processing of the audio files is given as it was needed to be able to work with Batmen. Finally, we discuss the lack of data for some species inside the dataset.

## 6.1 Dataset for the new IA

### 6.1.1 Compilation

The dataset for Batmen is a compilation of labelled sounds from the different projects of the Plecotus group that were collected between 2013 and 2018 (see the tables 6.1 and 6.2 to get the papers linked to the directory). These files were stored in three formats (.wav in real time, .wav with time expansion with a parameter of 10 and .wac files) and the format of the filename is $A-B\_C\_D\_E.wav$ with:

- A represents the code of the Song Meter if there are several for the project.
- B represents the project code, for example (BARBA for Barbalux).
- C represents the date when the sound was captured, for example 20170815 for 15/08/2017.
- D represents the sound capture time, for example 124537 for 12h45m37s.
- E represents the milliseconds of the capture time, for example 487.

A possible name could be A-BARBA-20170815_124537_478.wav. But projects of 2013 do not contain a Song Meter code and a project code, so the filename matches with the date when the file was recorded.

| Directory | Publication |
|---|---|
| 2017_Dasy_Escaut_Benoit_Gauquie | Etude acoustique des chauves-souris dans le bassin de l'Escaut, Hainaut occidental. [29] |
| Barbalux_2016 | A la recherche des Barbastrelles en province du Luxembourg. [32] |
| Barba_Lux_2017 | A la recherche des Barbastrelles en province du Luxembourg. [34] |
| Barba_Lux_2018 | A la recherche des Barbastrelles en province du Luxembourg. [35] |
| CavesPahautAutomne2013 | Project not found |
| Copie_de_Brabantwallon204 | Participation à la surveillance des populations de chiroptères : inventaires estivaux de chauves-souris dans quelques sites Natura 2000 [28] |
| Copie_de_Vesdre2014 | Participation à la surveillance des populations de chiroptères : inventaires estivaux de chauves-souris dans quelques sites Natura 2000 [28] |
| Ecoduc20150801-04-CecileHerr | Plecobrux : Monitoring des zones natura de Bruxelles. [36] |
| études_complementaires | Project not found |
| GueuleTOUT | Etude acoustique des chauves-souris dans la vallée de la gueule, Pays de herve.[29] |
| JenneretTOUT | Evaluation des actions du LIFE Prairies Bocagères en faveur des populations de petits rhinolophes, grand rhinolophes et murins à oreilles échancrées sur 10 sites Natura 2000 de Fagne et Famenne [29] |
| LIFEPrairiesBocageres_2013_TOUT | Evaluation des actions du LIFE Prairies Bocagères en faveur des populations de petits rhinolophes, grand rhinolophes et murins à oreilles échancrées sur 10 sites Natura 2000 de Fagne et Famenne [31] |
| LIFEPrairiesBocageres_2014 | Evaluation des actions du LIFE Prairies Bocagères en faveur des populations de petits rhinolophes, grand rhinolophes et murins à oreilles échancrées sur 10 sites Natura 2000 de Fagne et Famenne [31] |
| LPB2018 | Evaluation des actions du LIFE Prairies Bocagères en faveur des populations de petits rhinolophes, grand rhinolophes et murins à oreilles échancrées sur 10 sites Natura 2000 de Fagne et Famenne [33] |

Table 6.1: Listing of directory received from Natagora related to the publications.

| Directory | Publication |
|---|---|
| my_test | Evaluation des actions du LIFE Prairies Bocagères en faveur des populations de petits rhinolophes, grand rhinolophes et murins à oreilles échancrées sur 10 sites Natura 2000 de Fagne et Famenne citation |
| plateau_Engeland | Plecobrux : Monitoring des zones natura de Bruxelles. [36] |
| Plecobrux_ligne_161_tout | Plecobrux : Monitoring des zones natura de Bruxelles. [36] |
| Plecobrux_SM4_cécile | Plecobrux : Monitoring des zones natura de Bruxelles. [36] |
| Pleco_Lux_2016 | A la recherche des Barbastrelles en province du Luxembourg. [32] |
| Plecolux_2016-TOUT | A la recherche des Barbastrelles en province du Luxembourg. [32] |
| SM2_Escaut2015 | Etude acoustique des chauves-souris dans le bassin de l'Escaut, Hainaut occidental [29] |
| SM2_Hautes-Fagnes2014-TOUT | Etude acoustique des chauves-souris dans la réserve naturelle domaniale des Hautes-Fagnes [31] |
| SM2-Ligne161 | Plecobrux : Monitoring des zones natura de Bruxelles. [36] |
| Tournay-Solvay-DataSM2 | Plecobrux : Monitoring des zones natura de Bruxelles. [36] |
| TrouPicotIntérieurJuin2014 | Participation à la surveillance des populations de chiroptères : inventaires estivaux de chauves-souris dans quelques sites Natura 2000 [28] |

Table 6.2: Listing of directory received from Natagora related to the publications.

### 6.1.2 Selected files

The first step was to create a unique csv containing all the entries of each project, to know the diversity of data. The data were split into nine groups, the first seven match with the seven groups of bats Barba_G, ENVsp, Myosp, Pip35, Pip50, Plesp and Rhisp. Then the eighth group contains all sounds that triggered a detector without containing a bat, and finally the nineth group contains bat calls that are not identified.

Then, from this amount of sounds, the step 2 was to consider only calls labelled by SonoChiro and validated by a member of Plecotus, because only these files were corrected if SonoChiro was wrong.

The step 3 was to remove sounds containing multiple species from the dataset because each call was not related to the correct species and therefore it would be unusable by Batmen because an example cannot contain more than one class.

The fourth step was to check if every entry in the csv matches with a file inside the storage. And that was not always the case for three reasons.

- In some cases the pattern between B and C were correct in the filename, but in the csv, the pattern was different. It was *B_0_C* instead of *B_C*.(some files of the Plecolux_2016-TOUT project).
- The file was compressed into a wac file. In this case, the wac file associated to the wav file must be found and that is easy because the name of the wac file follows the same format as the wav file but without the millisecond. Once the list of wac is obtained, the wac file associated to the wav file is the closest in past file related to the date of the wav file. Then, the wac is decompressed and the wav file is obtained with the good filename. In some cases, the extraction of the wac file could give a wav file with a time rounded to the higher milliseconds.
- The file does not exist.

On this step, the group eight and the group nine were removed from the dataset because they are not needed to train the network.

Finally, for the last step, files were saved in time expansion with a parameter of 10 with the SoX software[12], because Batmen needs to use sounds in time expansion to work correctly. Files with a duration longer than 49 seconds were split into files with a maximum duration of 45 seconds using the algorithm described in the next section because the graphic card does not have enough RAM to process sounds that are too long.

### 6.1.3 Cutting audio files

As explained before, files with a longer duration than 50 seconds are too large for the graphic card, so these files were split into several files depending on the location of the bat call to avoid cutting a file during a call.

The algorithm works for an audio file and the input data are the duration in milliseconds in an integer format and the calls list containing each call inside the audio file. A call inside the list is a dictionary containing the key start for the start of the call and end for the end of the call. The output data is a list named timer containing the splitted timers. The algorithm is of our own creation and is not related to a known algorithm. It was designed for our problem which is cutting audio file without split a call inside both part.

The calls are going to be covered one by one (Listing 6.1 line 6). The line 7 to 10 ensures that the timers of the call are lower than the max duration and that it starts before it ends. Then, if the call starts further back in time than the max_timer, as many intervals will be added to make sure that the call is not between two intervals. But if the call starts before the end of a interval and ends after, the interval will end in the half of time between the previous and the current call. And as no interval was created before and that the current call start before the max_timer, we know that we can create this interval respecting the max_timer. Finally, from the line 25 to 27, intervals respecting the condition are created until the end of the wav file.

```
1  def split_audio(duration, calls):
2      timer = [0]
3      max_timer = 45
4      previous_start_call = 0
5      previous_end_call = 0
6      for call in calls:
7          if call[end] < call[start]:
8              return []
9          elif call[end] - call[start] > max_timer:
10             return []
11         elif call[start] - timer[-1] > max_timer:
12             count = (call[start] - timer[-1]) / max_timer
13             while count > 0:
14                 timer.append(timer[-1] + max_timer)
15                 count -= 1
16             if call[end] - timer[-1] > max_timer:
17                 interval = call[start]- (max_timer -(call[end] - call[start]))/2
18                 timer.append(interval)
19         elif call[start] - timer[-1] < max_timer and call[end] - timer [-1] > max_timer:
20             interval = (previous_end_call + call[start]) /2
21             timer.append(interval)
22         previous_start_call = call[start]
23         previous_end_call = call[end]
24
25     while duration - timer[-1] > max_timer:
26          timer.append(timer[-1] + max_timer)
27     timer.append(duration)
28     return timer
```

Listing 6.1: Cutting audio files.

### 6.1.4 Labelling of call

The data coming from Natagora were labelled only for the species and not for the times of the calls. We had an approximation of the number of calls inside the csv, because with the analysis of Kaleidoscope and SonoChiro, they gave the number of calls that they detected in the audio files. Therefore, to get the timer of the calls, we used Bat detective. The number of calls found by Bat detective was always below the number of calls found by Kaleidoscope and SonoChiro. This should have an impact on Batmen's performance.

The choice of labelling is very hard to do. In fact, a human labelling, besides being tedious and time consuming, would have allowed to label more calls, but some of them would have been deleted by the denoising algorithm which would cause a raise of false positives. As they are deleted, they are not visible anymore on the spectrogram but are still labelled and therefore undetectable. While the Bat detective labelling would have less false positives, it is possible that some calls that were not labelled would still be visible after the denoising algorithm and that would also have an impact on performance. This can create false positives if the call is detected but not labelled, or we could have the training of the network on the non-presence of a call while there is one.

### 6.1.5 Final dataset

The table 6.3 shows the evolution of the number of files for each group as well as their proportions since the first step. We observe that the Pip50 is the most represented group in Belgium with more than 50% of the trigger record followed by the group 9, which contains unidentified bats, with 19.55%, that shows the need of a better way to classify bats. We notice that the Barba_G and Plesp groups are underrepresented but are more often labelled by a human when a file exists because of their rarities. On the other hand, the Pip50 group is more present in Belgium and is less labelled by hand. We see at step 5 that the ENVsp, Plesp and Barba_G trigger bigger files than the other groups. Finally, as explained before, the eighth and ninth groups are removed before searching their location inside the storage to avoid unnecessary calculations, and because they cannot be used properly by Batmen because of the labelling problem.

| Group | Step 1: Files in csv | | Step 2: Labelled by a human | | Step 3: Cleaning | | Step 4: Found in storage | | Step 5: File to be cut | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Qty | % | Qty | % | Qty | % | Qty | % | Qty | % |
| Barba_G | 194 | 100% | 90 | 46.39% | 16 | 8.25% | 12 | 6.19% | 7 | 3.61% |
| ENVsp | 13244 | 100% | 962 | 7,26% | 526 | 3.97% | 326 | 2.46% | 176 | 1.33% |
| Myosp | 62163 | 100% | 1653 | 2.66% | 953 | 1.53% | 886 | 1.42% | 320 | 0.51% |
| Pip35 | 7559 | 100% | 448 | 5.93% | 292 | 3.86% | 161 | 2.13% | 49 | 0.65% |
| Pip50 | 497662 | 100% | 592 | 0.12% | 380 | 0.08% | 257 | 0.05% | 107 | 0.02% |
| Plesp | 1330 | 100% | 150 | 11.28% | 95 | 7.14% | 61 | 4.59% | 35 | 2.63% |
| Rhisp | 23840 | 100% | 653 | 2.74% | 396 | 1.66% | 215 | 0.90% | 21 | 0.09% |
| Group 8 | 37502 | 100% | 1034 | 2.76% | 684 | 1.82% | 0 | 0% | 0 | 0% |
| Group 9 | 180659 | 100% | 1187 | 0.66% | 790 | 0.44% | 0 | 0% | 0 | 0% |
| Sum | 924153 | 100% | 6769 | 0.82% | 4132 | 0.50% | 1918 | 0.23% | 715 | 0.09% |

Table 6.3: Evolution of the number of files per group, depending on the step.

The splitting of the 715 files gave 1664 files. Some were divided into 20 files but for the most part they were divided into two files. And from this amount of files, only 1525 files were still containing calls and were kept for Batmen. The table 6.4 shows that the Barba_G and Plesp groups are underrepresented in the data and should give bad results with Batmen. We can also see that the number of calls per file varies depending on the group.

| | Rest of step 5 | Split files | Number of files | Number of calls | Average number of calls per file |
|---|---|---|---|---|---|
| Barba_G | 5 | 13 | 18 | 678 | 37 |
| ENVsp | 150 | 452 | 602 | 17782 | 29 |
| Myosp | 566 | 610 | 1176 | 25315 | 21 |
| Pip35 | 112 | 94 | 206 | 5332 | 25 |
| Pip50 | 150 | 249 | 399 | 8948 | 22 |
| Plesp | 25 | 64 | 89 | 1499 | 16 |
| Rhisp | 194 | 43 | 237 | 3800 | 16 |

Table 6.4: Table representing the proportion between split and common files.

| Species | Training | | Testing | |
|---|---|---|---|---|
| | Files | Calls | Files | Calls |
| **Barba_G** | 16 | 625 | 2 | 53 |
| **Barba_G** | **16** | **625** | **2** | **53** |
| **Nycnoc** | 74 | 1522 | 9 | 115 |
| **Envsp** | 213 | 7109 | 29 | 1029 |
| **Eptnil** | 2 | 108 | 0 | 0 |
| **Eptser** | 87 | 2400 | 8 | 231 |
| **Nyclei** | 165 | 4899 | 15 | 369 |
| **Envsp** | **541** | **16038** | **61** | **1744** |
| **Myodau** | 130 | 4936 | 11 | 340 |
| **Myonat** | 37 | 1026 | 5 | 55 |
| **Myodas** | 13 | 503 | 1 | 37 |
| **Myoalc** | 18 | 334 | 3 | 63 |
| **Myobra** | 17 | 601 | 2 | 54 |
| **Myoema** | 65 | 1950 | 8 | 154 |
| **Myosp** | 600 | 9937 | 66 | 1205 |
| **Myobec** | 2 | 45 | 0 | 0 |
| **Myomys** | 86 | 2141 | 11 | 339 |
| **Myomyo** | 90 | 1413 | 11 | 182 |
| **Myosp** | **1058** | **22886** | **118** | **2429** |
| **Pip35** | 32 | 767 | 4 | 129 |
| **Pipnat** | 153 | 3776 | 17 | 660 |
| **Pip35** | **185** | **4543** | **21** | **789** |
| **Pip50** | 39 | 730 | 5 | 87 |
| **PippiT** | 303 | 7055 | 35 | 701 |
| **Pippyg** | 17 | 375 | 0 | 0 |
| **Pip50** | **359** | **8160** | **40** | **788** |
| **Pleaus** | 28 | 542 | 1 | 18 |
| **Pleaur** | 19 | 275 | 5 | 134 |
| **Plesp** | 33 | 448 | 3 | 82 |
| **Plesp** | **80** | **1265** | **9** | **234** |
| **Rhihip** | 119 | 1545 | 12 | 250 |
| **Rhifer** | 94 | 1849 | 12 | 156 |
| **Rhisp** | **213** | **3394** | **24** | **406** |

Table 6.5: Table representing the proportion between training and testing files.

The dataset, represented by the table 6.5, was divided into two groups for the neural network. One is for the training and one is for the testing in the following proportions: 9 files for the training per 1 file for the testing. The sum for the different groups is represented by the cell with a grey background color, and each line between these sums represent all species contained inside the group. We can see that a lot of species are underrepresented in the data e.g., Eptnil, Myobec, while they belong in a highly represented group. Other groups like Barba_G and Plesp do not contain enough data to train effectively Batmen. We notice that some groups contain their own group, it is because some calls were identified by their groups and not the species.

## 6.2 Conclusion

The dataset to train Batmen is based on data received from Natagora. They collected the data all over Wallonia and Brussels. Some of these calls were labelled by hand in addition to some analysis by a software. Only this data was used because it was more likely to be properly labeled even if in some cases, some data were labelled two times for different species. Then, we remove from this amount of data, those containing more than one species because we had to label manually the timer of the calls and check for each of them from what species it comes from. Obviously, this was not feasible with the amount of files and calls (684 files containing at least 2 species and an average number of 25 calls per file). Those not containing calls or those not containing enough information to be classified inside the correct group were removed as well. In fact, we had to manually check if each file does not contain calls that could be detected as noise or if we needed to classify it. Finally, the removed data represent 35,67% of the data that could have been used.

# Chapter 7

# Implementation

This chapter covers the changes that were made to the bigger network from the Bat detective project. First, the goal of this extension are described. Then, the choices that were made in terms of class labelling, network architecture and classification strategy are explained. Finally, the training times and performance measures are given and discussed.

## 7.1 Goal

The purpose of this extension is to provide the classification of bat calls instead of just detecting the presence of calls. To this end, we took the open-source implementation of [25] and modified some aspects. In fact, the base project is implemented as a binary classification problem i. e., whether at a certain time, there is a bat call or there is not. The goal is transform the binary classification problem into a multiclass classification problem. The reason is that we would like to retrieve the individual bat species responsible for the calls inside the audio files. These modifications involve changes in the architecture of the neural network as well as how the dataset is trained, treated and how we measure the performance.

## 7.2 Classes choice

| Class label | Group |
|---|---|
| 0 | No calls |
| 1 | Barba_G |
| 2 | ENVsp |
| 3 | Myosp |
| 4 | Pip35 |
| 5 | Pip50 |
| 6 | Plesp |
| 7 | Rhisp |

Table 7.1: Class numbers relative to bat groups.

The initial problem is a binary classification problem. This means that at each position, there is either no bat calls which corresponds to the class label 0 or an actual call, corresponding to the class label 1. As discussed in chapter 6, due to the lack of data for specific species, we chose to represent each class by a bat group. In fact, the data available show some imbalance between the different species. The choice of groups follows the standard that was decided by the researchers as they had trouble identify certain species as discussed in chapter 2.

The table 7.1 depicts the association of class labels from one to eight with the different groups. There are eight groups as we want to keep the distinction between finding a call at a certain time interval and not finding one. Then, the call can be classified between the seven classes remaining.

## 7.3 Architecture & strategy

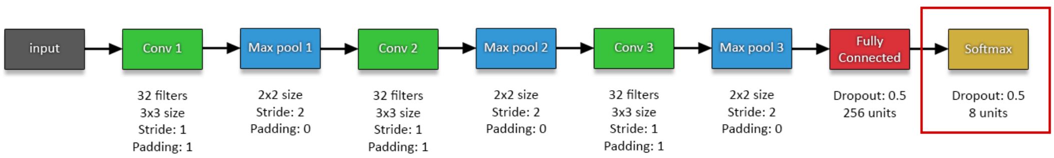


Figure 7.1: Batmen neural network architecture.

The figure 7.1 shows the modifications to the bigger network that were applied to the previous neural network from Bat detective. For our multiclass classification, the idea is to associate a time window in the spectrogram with the presence of the call and a reference to the predicted class.

We achieve this behavior by using a softmax output layer with $n$ units [22] where $n$ is the number of classes. The output values obtained range from 0 to 1 and are used as probabilities. In fact, the highest probability will determine the chosen class. In our case, we separate the class 0 from the other classes first as we only care about actual calls and we associate a class for each time some call is found. Then, with non maximum suppression, we associate out of these pairings a more precise time when the actual call starts with a prediction on the class.

The rest of the network architecture remains similar to the Bat detective project as the modifications were targeted on the last layer.

## 7.4 Training

The model was trained in the same conditions as the Bat detective algorithm. Nevertheless, the chosen dataset is the one that was created for Batmen and that we discussed in chapter 6. The specifications of our training computer are the following:

- CPU: Intel(R) Core(TM) i7-9800X CPU @ 3.80GHz
- RAM: 32 GB
- GPU: GeForce RTX 2080 SUPER
- OS: CentOS 8
- CUDA version: 10.2

The table 7.2 provides the times needed for our training computer to train the Batmen neural network. The hard negative mining was chosen to be two rounds as it is resource and time consuming.

| **Algorithm** / **Training** | **Batmen (no hnm)** | **Batmen (2 hnm rounds)** |
|---|---|---|
| **Time** | 46 min | 3h15 |

Table 7.2: Times for Bat detective and Batmen algorithms.

## 7.5 Results

Before we assess the performance of Batmen's classification. It is interesting to compare the performance on the detection of calls for our dataset in comparison with the measures of the three Bat detective datasets in figure 5.10.

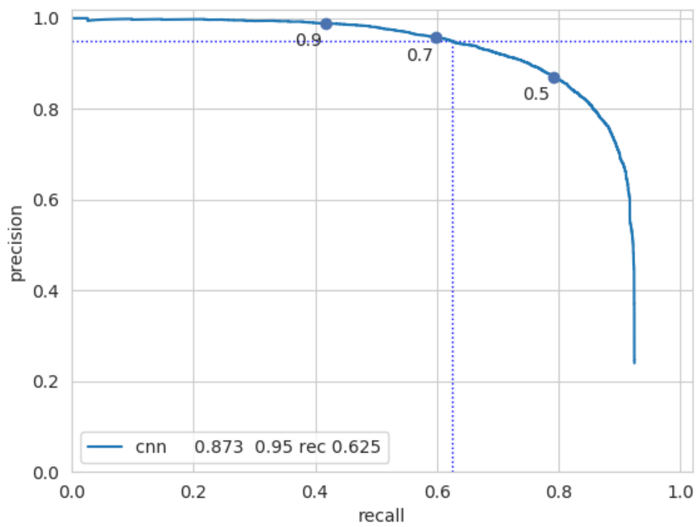


Figure 7.2: Precision-recall curve for Batman dataset detection.

| Dataset \ Measure | Precision | Recall at 0.95 |
|---|---|---|
| **Batmen** | 0.873 | 0.625 |
| **UK** | 0.866 | 0.670 |
| **Norfolk** | 0.882 | 0.754 |
| **Romania & Bulgaria** | 0.895 | 0.818 |

Table 7.3: Performance comparison for Batmen dataset and previous datasets.

The figure 7.2 shows the precision-recall curve for our dataset. As a reminder, the recall is calculated at 0.95 precision, a metric that measures the fraction of calls that were detected while accepting a false positive rate of 5%. We can observe with the table 7.3 that the detection performs slightly worse in terms of recall (0.625) compared to the three Bat detective datasets.

As discussed previously, the classification goal is for several bat species groups. Therefore, since we are working with a multiclass classification problem, the performance measure is provided by a confusion matrix of size $n$ where $n$ is the number of relevant classes. This helps us measure the performance for each class individually and then overall. The matrix $C$ below represents the individual calls found and how the model classifies them. The diagonal in the matrix gives us the true positives for each class. The other elements in the matrix are given for each class $i$ and n=6 as follows :

$$TP_i = C_{ii}$$

$$FP_i = \sum_{k=0}^{n} C_{ki} - C_{ii}$$

$$FN_i = \sum_{k=0}^{n} C_{ik} - C_{ii}$$

$$TN_i = \sum_{k=0}^{n} \sum_{l=0}^{n} C_{kl} - TP_i - FP_i - FN_i$$

where :

TP = True positive

FP = False positive

FN = False negative

TN = True negative

Three performance measures are used for our problem. The accuracy defines how well the model's predictions are compared to the total predictions. Then, the precision which defines the ratio of true predicted positives over the total predicted positives, i.e., how well the model predicts positives. Finally, the recall gives the ratio of predicted true positives over the actual positives, i.e., how well the model identified the actual positives. The three measures are given by:

$$\textbf{Accuracy} = \frac{True\,positive + True\,negative}{False\,negative + True\,positive + False\,positive + True\,negative}$$

$$\textbf{Precision} = \frac{True\,positive}{False\,positive + True\,positive}$$

$$\textbf{Recall} = \frac{True\,positive}{False\,negative + True\,positive}$$

The elements given in the confusion matrix are the calls detected and classified by the algorithm. In fact, the idea is to first eliminate the highest predictions where there are no calls present i.e., labelled 0. This allows us to get the predictions for the actual classes needed for our classification problem. Then, the highest predictions and their corresponding class labels are processed with a Gaussian filter and non-maximum suppression to obtain the precise call positions, their probabilities and their class labels.

### 7.5.1 Without hard negative mining

First, the matrix below represents the classification for our dataset without the use of hard negative mining:

$$
\begin{array}{c|ccccccc}
 & Barba_G & ENVsp & Myosp & Pip35 & Pip50 & Plesp & Rhisp \\
\hline
Barba_G & \mathbf{34} & 0 & 170 & 0 & 1 & 18 & 0 \\
ENVsp & 0 & \mathbf{3185} & 2034 & 30 & 174 & 14 & 52 \\
Myosp & 1 & 143 & \mathbf{10379} & 12 & 206 & 6 & 111 \\
Pip35 & 0 & 129 & 1193 & \mathbf{746} & 286 & 13 & 7 \\
Pip50 & 37 & 144 & 1539 & 57 & \mathbf{1391} & 72 & 55 \\
Plesp & 0 & 107 & 650 & 3 & 66 & \mathbf{156} & 0 \\
Rhisp & 0 & 35 & 301 & 27 & 134 & 0 & \mathbf{1406}
\end{array}
$$

| Performance \ Group | Barba_G | ENVsp | Myosp | Pip35 | Pip50 | Plesp | Rhisp |
|---|---|---|---|---|---|---|---|
| **True positive** | 34 | 3185 | 10379 | 746 | 1391 | 156 | 1406 |
| **False positive** | 38 | 558 | 5887 | 129 | 867 | 123 | 225 |
| **False negative** | 189 | 2304 | 479 | 1628 | 1904 | 826 | 497 |
| **True negative** | 24863 | 19077 | 8379 | 22621 | 20962 | 24019 | 22996 |
| **Accuracy** | 0.9909 | 0.8861 | 0.7466 | 0.9301 | 0.8897 | 0.9622 | 0.9713 |
| **Precision** | 0.4722 | 0.8509 | 0.6381 | 0.8526 | 0.6161 | 0.5591 | 0.8620 |
| **Recall** | 0.1525 | 0.5802 | 0.9559 | 0.3142 | 0.4221 | 0.1588 | 0.7388 |

Table 7.4: Performance measure for each class label.

The table 7.4 represents for each bat group how each individual “call” was classified and their accuracy, precision and recall.

Immediately, we can observe that the accuracy is high overall. Nevertheless, several groups suffer from low precision rates and low recall rates. For the Barba_G, both measures, 0.4722 and 0.1525 respectively, are lower compared to the other groups. This can be attributed to the lack of overall data available for the group (and species since there is only one in the group) as discussed in chapter 6. Interestingly, the Myosp group which is the group with the most data available results in a quite low precision and accuracy but shows high recall. A justification for this behavior might come from the imprecision in the automatic labelling process, creating higher number of false positives in the detection. This echoes on the classification part.

Moreover, we can see that the recall is low for other groups such as Plesp, Pip35 or Pip50. We can observe that other than the data availability issue, this behavior might come from the similarity is call structures as the Rhisp group, whose call structure is CF, thus very different from the other groups, shows good rates for precision and recall.

### 7.5.2 With hard negative mining

The same testing is done with this time two rounds of hard negative mining. This training technique was kept from the Bat detective project as it also gave an increase in performance for this project.

$$
\begin{array}{c c}
 & \begin{array}{ccccccc} Barba_G & ENVsp & Myosp & Pip35 & Pip50 & Plesp & Rhisp \end{array} \\
\begin{array}{c} Barba_G \\ ENVsp \\ Myosp \\ Pip35 \\ Pip50 \\ Plesp \\ Rhisp \end{array} &
\begin{pmatrix}
\mathbf{69} & 1 & 166 & 0 & 0 & 1 & 0 \\
1 & \mathbf{2978} & 2122 & 33 & 183 & 20 & 52 \\
4 & 153 & \mathbf{10193} & 20 & 249 & 6 & 108 \\
1 & 158 & 1213 & \mathbf{673} & 296 & 0 & 6 \\
39 & 143 & 1441 & 44 & \mathbf{1511} & 61 & 62 \\
0 & 101 & 680 & 22 & 26 & \mathbf{150} & 0 \\
0 & 28 & 248 & 15 & 158 & 0 & \mathbf{1402}
\end{pmatrix}
\end{array}
$$

| Performance \ Group | **Barba_G** | **ENVsp** | **Myosp** | **Pip35** | **Pip50** | **Plesp** | **Rhisp** |
|---|---|---|---|---|---|---|---|
| **True positive** | 69 | 2978 | 10193 | 673 | 1511 | 150 | 1402 |
| **False positive** | 45 | 584 | 5870 | 134 | 912 | 88 | 228 |
| **False negative** | 168 | 2411 | 540 | 1674 | 1790 | 829 | 449 |
| **True negative** | 24555 | 18864 | 8234 | 22356 | 20624 | 23770 | 22758 |
| **Accuracy** | 0.9914 | 0.8794 | 0.7419 | 0.9272 | 0.8912 | 0.9631 | 0.9727 |
| **Precision** | 0.6053 | 0.8360 | 0.6346 | 0.8339 | 0.6236 | 0.6302 | 0.8601 |
| **Recall** | 0.2911 | 0.5526 | 0.9497 | 0.2867 | 0.4574 | 0.1532 | 0.7574 |

Table 7.5: Performance measure for each class label.

The addition of hard negative mining does not change the overall behavior of the model as we can see from 7.5. However, in the same manner as for Bat detective, the technique improved the performance measures by a small margin. For example, the previous problematic group, Barba_G gets an increase in precision from 0.4722 to 0.6053 and from 0.1525 to 0.2911 for the recall.

### 7.5.3 Comparison

Finally, we provide the same performance measures in average for both approaches. This is an average over every class we previously measured.

The table 7.6 gives the average accuracy, precision and recall for our model with two rounds of hard negative mining and without hard negative mining. As expected, both approaches give similar results in average but importantly, hard negative mining provides a small improvements on the precision and recall.

| Algorithm / Performance | With hard negative mining | Without hard negative mining |
|---|---|---|
| **Average accuracy** | 0.9096 | 0.9110 |
| **Average precision** | 0.71767 | 0.693 |
| **Average recall** | 0.49265 | 0.4747 |

Table 7.6: Average performance for both approaches.

## 7.6 Conclusion

Batmen is a project that builds on the code provided by Bat detective. It transforms the previous binary classification problem, i.e., the detection of a call or not, into a multiclass classification problem. Each class represents a bat species group as it balances the dataset more. The architecture of the Batmen's neural network is a slightly modified version of the bigger convolutional neural network from Bat detective. In fact, the last layer has to change as the output must predict more than two classes. The performance of Batmen's classification were tested with and without hard negative and has shown that hard negative mining still gives better results in the same manner as Bat detective. Finally, we saw that the imbalance aspect of the dataset shows problem in the dataset and needs further investigation and changes both in the training-testing strategy and in the network's architecture.

# Chapter 8

# Replications

This chapter will explain how to get and install Batmen. After that, we show how to produce a dataset file containing the link between the labelled calls and the network for the training.

## 8.1 Installation

The code for Batmen can be found at this link https://github.com/c-lipani/batmen. Be aware that Batmen needs a lot of Python libraries to work. The best way to install them is to use the Anaconda software [4], as you can directly install the same version of each package with the configuration file available at this link https://github.com/FrancoMaxime/batconfig.

## 8.2 NPZ File

To train Batmen, you need to create a NPZ file to make the link between the location of the audio file, the calls and the network. A NPZ file is a NumPy file that saves objects, the file used by Batmen must contain 8 objects. Four NumPy Array objects for the training and Four NumPy array objects for the testing. Each object for the training contains the same number of entry because they are interconnected via the indexes.

1. train_durations: An array containing the duration of each audio file.
2. train_files: An array containing the location of each audio file.
3. train_pos: An array containing a NumPy array containing the timer of each call.
4. train_class: An array containing for each audio file the class of the bat that emitted the call.

The four other objects are pretty much the same as the training objects but for the testing and follows the same construction.

1. test_durations
2. test_files
3. test_pos
4. test_class

## 8.3 Discussion

As the data provided by Natagora are private, we cannot share them, but you can use your own data and use the previous point to train Batmen to get your own results.

# Chapter 9

# Conclusion

This thesis aimed to build a machine learning tool that was able to classify bat species in Belgium through the use of a convolutional neural network. As we have seen before, anthropogenic change calls for better means of monitoring climate change and habitat conversion. Monitoring bats is actually an excellent way of providing such a thing.

The twenty-three species that are currently identified in Belgium individually have very different behaviors in terms of roosting and even with the use of their echolocation patterns. The lack of information surrounding these Belgian species has lead volunteer group works such as Natagora to collect more and more data, giving a better idea of the bat's role and presence in this country.

The different methods to detect calls were introduced and compared. The time expansion detector is the most powerful detector if real time is not needed because it records calls with full details while the other detectors have a more or less significant loss. Furthermore, we saw inside the chapter 3, that the computer analysis of time expansion call were the most powerful way to analyse and classify bats. The methods to analyse a call such as the analysis of the structure of the call or the spectral density were described and used in an example concluding the chapter.

Even though several supervised learning techniques are presented, it is hard to find a flexible and scalable approach to the bat classification problem. In fact, all these approaches have their own constraints and it is hard to assess which one would perform the best for a specific problem and for a specific dataset such as bat calls datasets. The project Bat detective was the closest open-source tool available that could support the modifications needed for our classification problem.

A first improvement for the dataset would be to cut the sounds around the call, to decrease the number of calls that were not labelled by Bat detective. This should slightly increase the performance of Batmen if there are a lot of unlabelled calls that are detected by Batmen. Also, this would accelerate the training time. An other improvement for the dataset would be to separate the data for the testing and the training depending on the number of call instead of the number of file. This would allows us to have a number of calls for the training and the testing that respects even more the division of nine elements for the training for one for the testing.

As for the Batmen performance, different factors impacted it. First, the labelling of the call timers was made with the Bat detective algorithm and it found less calls than predicted by Kaleidoscope and SonoChiro, the number of false positives for Batmen thus increased. Moreover, the dataset is not balanced because of the available data on Belgian bat species even though groups were used to reduce the inequality between the species. It is important to note that inside an individual group, there can be a disparity in terms of proportion between the species. If a highly present species in a group has a similar call structure with another species from a different group, then it can have a big impact on the predictions.

In the future, an other change could be to over-sample minority classes or under-sample majority classes as we get new data. A hierarchy of classes could be built in several multiclass sub-problems and classify in a tree-like approach. Moreover, the approach taken to build a classification tool over the Bat detective project with one neural network was one idea. First, there could be several changes to be brought to the network's architecture and hyperparameters. Another approach would be to build two neural networks, with one feeding on the other as they have different goals. This could lead to more control over the parameters and overall higher performance.

Finally, the work that was done during this thesis can help the next generation of tools that could be built in the same classification goals. In fact, the knowledge taken from the experiment and the data processing methods throughout the project can always be valuable.

# Bibliography

[1] W. Acoustics. Kaleidoscope pro. https://www.wildlifeacoustics.com/products/kaleidoscope-pro, 2020.

[2] W. Acoustics. How shoudl i save, in .wac or in .wav? https://www.wildlifeacoustics.com/resources/blog/how-should-i-save-in-wac-or-in-wav, 2020.

[3] T. M. Aide, C. Corrada-Bravo, M. Campos-Cerqueira, C. Milan, G. Vega, and R. Alvarez. Real-time bioacoustics monitoring and automated species identification. *PeerJ*, 1:e103, 2013.

[4] I. Anaconda. Anaconda software. https://www.anaconda.com/products/individual, 2020.

[5] D. Armitage and H. Ober. A comparison of supervised learning techniques in the classification of bat echolocation calls. *Ecological Informatics*, 5:465–473, 11 2010. doi: 10.1016/j.ecoinf.2010.08.001.

[6] M. Barataud. *Écologie acoustique des Chiroptères d'Europe, identification des espèces, étude de leurs habitats et comportements de chasse.* Biotope, Mèze, 3. ed. edition, 2015. ISBN 9782366621426. URL https://leclub-biotope.com/fr/librairie-naturaliste/772-ecologie-acoustique-des-chiropteres-d-europe.

[7] Biotope. Sonochiro software. https://sonochiro.biotope.fr/, 2020.

[8] R. Bradshaw, S. Behnel, D. S. Seljebotn, G. Ewing, et al. The cython compilers. http://cython.org, 2020.

[9] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[10] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[11] T. Chen and C. Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

[12] e. a. Chris Bagwell. Sox - sound exchange. http://sox.sourceforge.net/, 2019.

[13] N. Cláudio. Automatic identification of bat species using supervised learning. 2017.

[14] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[15] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, et al. Lasagne: First release., Aug. 2015. URL http://dx.doi.org/10.5281/zenodo.27878.

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[17] D. B. group. Bat detectors: Frequency division. http://www.durhambats.org.uk/fd.shtml, 2015.

[18] A. Haquart. Sonochiro® 4.0 : Les classificateurs. https://sonochiro.biotope.fr/doc/FR/4-sonochiro-classificateurs.html, 2017.

[19] L. Jakobsen, J. Hallam, C. F. Moss, and A. Hedenström. Directionality of nose-emitted echolocation calls from bats without a nose leaf (plecotus auritus). *Journal of Experimental Biology*, 221(3), 2018. ISSN 0022-0949. doi: 10.1242/jeb.171926. URL https://jeb.biologists.org/content/221/3/jeb171926.

[20] G. Jones, D. Jacobs, T. Kunz, and P. Racey. Carpe noctem: The importance of bats as bioindicators. *Endangered Species Research*, 8:93–115, 07 2009. doi: 10.3354/esr00182.

[21] S. L. S. L. J.-P. B. Jérémy SIMAR, Thierry KERVYN. Projets éoliens note de référence pour la prise en compte dela biodiversité. https://www.bats.org.uk/our-work/national-bat-monitoring-programme/past-projects/ibats, 2012.

[22] Lasagne. Tutorial - convolutional neural network (cnn). https://lasagne.readthedocs.io/en/latest/user/tutorial.html, 2020.

[23] M. Lasseck. Large-scale identification of birds in audio recordings. 09 2014.

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[25] O. Mac Aodha, R. Gibb, K. E. Barlow, E. Browning, M. Firman, R. Freeman, B. Harder, L. Kinsey, G. R. Mead, S. E. Newson, I. Pandourski, S. Parsons, J. Russ, A. Szodoray-Paradi, F. Szodoray-Paradi, E. Tilova, M. Girolami, G. Brostow, and K. E. Jones. Bat detective—deep learning tools for bat acoustic signal detection. *PLOS Computational Biology*, 14(3):1–19, 03 2018. doi: 10.1371/journal.pcbi.1005995. URL https://doi.org/10.1371/journal.pcbi.1005995.

[26] Natagora. Suivi des populations de chauves-souris en belgique grâce aux recensements hivernaux. https://plecotus.natagora.be/fileadmin/Plecotus/inventaires_chiro_hivernal.pdf, 2017.

[27] Y. E. NESTEROV. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983. URL https://ci.nii.ac.jp/naid/10029946121/en/.

[28] P. Nyssen. Participation à la surveillance des populations de chiroptères en wallonie : inventaires estivaux de chauves-souris dans quelques sites natura 2000 - résultats des recensements réalisés au cours de l'été 2014. 02 2015. doi: 10.13140/RG.2.1.4245.9689.

[29] P. Nyssen. Etude acoustique des chauves-souris dans le bassin de l'escaut, hainaut occidental - rapport - résultats des enregistrements automatiques sm2bat+ sur l'été 2015. 03 2017.

[30] T. E. Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[31] Plecotus. Etude acoustique des chauves-souris dans la réserve naturelle domaniale des hautes-fagnes. https://plecotus.natagora.be/fileadmin/Plecotus/Echo/RAPPORTSM2HautesFagnes2014.pdf, 2015.

[32] Plecotus. A la recherche des barbastrelles en province du luxembourg. 2016. https://plecotus.natagora.be/fileadmin/Plecotus/site/RapportBarba2016.pdf, 2017.

[33] Plecotus. Evaluation des actions du life prairies bocagères en faveur des populations de petits rhinolophes, grand rhinolophes et murins à oreilles échancrées sur 10 sites natura 2000 de fagne et famenne. https://plecotus.natagora.be/fileadmin/Plecotus/Documentation/LIFEPB_Rapport_chauves-souris_2018.pdf, 2018.

[34] Plecotus. A la recherche des barbastrelles en province du luxembourg. 2017. https://plecotus.natagora.be/fileadmin/Plecotus/Documentation/RapportBarba2017.pdf, 2018.

[35] Plecotus. A la recherche des barbastrelles en province du luxembourg. 2018. https://plecotus.natagora.be/fileadmin/Plecotus/RapportBarba2018.pdf, 2019.

[36] Plecotus. Plecobrux : Monitoring des zones natura de bruxelles. https://plecotus.natagora.be/index.php?id=723, 2019.

[37] Plecotus. Les différentes espèces de chauves-souris. https://plecotus.natagora.be/index.php?id=700, 2020.

[38] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. URL http://arxiv.org/abs/1605.02688.

[39] B. C. Trust. ibats project. https://www.bats.org.uk/our-work/national-bat-monitoring-programme/past-projects/ibats, 2005-2006.

[40] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL https://doi.org/10.7717/peerj.453.

[41] S. Verkem. Manuel d'utilisation du détecteur d'ultrasons hétérodyne pour débutants. https://plecotus.natagora.be/fileadmin/Plecotus/Documentation/Manuel_heterodyne_versionMars2014.pdf, 2014.

[42] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: https://doi.org/10.1038/s41592-019-0686-2.

[43] Wikipedia. Nyquist–shannon sampling theorem. https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem, 2020.

[44] Wikipedia. Acuité auditive. https://fr.wikipedia.org/wiki/Acuit%C3%A9_auditive, 2020.

[45] Zooniverse. Zooniverse. www.zooniverse.org, 2020.

**UNIVERSITÉ CATHOLIQUE DE LOUVAIN**
**École polytechnique de Louvain**
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**