Whistler Waves Detection

Investigation of modern machine learning techniques to detect and characterise whistler waves

Presented by:
Othniel J.E.Y. Konan
KNNOTH001
Dept. of Electrical and Electronics Engineering
University of Cape Town

Prepared for:
Dr. Amit Mishra
Dept. of Electrical and Electronics Engineering
University of Cape Town
and
Dr. Stefan Lotz
Science Research and Applications
South African National Space Agency

A dissertation submitted to the Department of Electrical Engineering,
University of Cape Town in partial,
in partial fulfillment of the academic requirements for the
Master of Science in Electrical Engineering specialising in
Radar and Electronic Defense

July 6, 2021

Key Words: Very Low Frequency Waves, Whistler Radio Waves, CFAR, Object detection

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.

3. This report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signed by candidate

Othniel J. E. Y. Konan

Cape Town
July 6, 2021

Abstract

Lightning strokes create powerful electromagnetic pulses that routinely cause very low frequency (VLF) waves to propagate across hemispheres along geomagnetic field lines. VLF antenna receivers can be used to detect these whistler waves generated by these lightning strokes.

The particular time/frequency dependence of the received whistler wave enables the estimation of electron density in the plasmasphere region of the magnetosphere. Therefore the identification and characterisation of whistlers are important tasks to monitor the plasmasphere in real time and to build large databases of events to be used for statistical studies.

The current state of the art in detecting whistler is the Automatic Whistler Detection (AWD) method developed by Lichtenberger (2009) [1]. This method is based on image correlation in 2 dimensions and requires significant computing hardware situated at the VLF receiver antennas (e.g. in Antarctica).

The aim of this work is to develop a machine learning based model capable of automatically detecting whistlers in the data provided by the VLF receivers. The approach is to use a combination of image classification and localisation on the spectrogram data generated by the VLF receivers to identify and localise each whistler. The data at hand has around 2300 events identified by AWD at SANAE and Marion and will be used as training, validation, and testing data.

Three detector designs have been proposed. The first one using a similar method to AWD, the second using image classification on regions of interest extracted from a spectrogram, and the last one using YOLO, the current state of the art in object detection. It has been shown that these detectors can achieve a misdetection and false alarm rate, respectively, of less than 15% on Marion's dataset.

It is important to note that the ground truth (initial whistler label) for data used in this study was generated using AWD. Moreover, SANAE IV data was small and did not provide much content in the study.

Acknowledgments

I was assisted by many people throughout the course of this research. First and foremost, I would like to express my gratitude to both my supervisors, Dr Amit Mishra from the University of Cape Town (UCT) and Dr Stefan Lotz from the South African Space Agency (SANSA) for their advice and guidance throughout the course of this project. Dr Amit Mishra, for making the time to get weekly updates on the project, for providing direction to the research. Dr Stefan Lotz, for providing the data required for this research and other resources, for introducing me to the Machine Learning Conference in Helio Physic which provided me with a deeper insight into this project.

I would also like to thank Dr Steven Pain and doctor-to-be, Darryn Jordan from the Radar Remote Sensing Group (RRSG) of UCT for their insight on this project and also for providing help with setting up the servers. To the other colleagues from the RRSG, I would like to express my gratitude for the time you have made helping me understand new facets of the research.

I would like to thank the Mandela Rhodes Foundation for their financial and leadership support for the two years of master degree.

Finally, I would like to express my gratitude to friends and family and especially my girlfriend Michaela for her support and for proofreading this dissertation.

Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analogue to Digital Converter
ANN	Artificial Neural Network
AWDAnet	Automatic Whistler Detector Analysis Network
AWD	Automatic Whistler Detector
CFAR	Constant False Alarm Rate
CNN	Convolutional Neural Network
DTFT	Discrete Time Fourier Transform
FFT	Fast Fourier Transform
RoI	Region of Interest
SANSA	South African National Space Agency
SNR	Signal to Noise Ratio
SOTA	State of the art
STFT	Short-Time Fourier Transform
VLF	Very Low Frequency
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Background to Project

Lightning strokes create powerful electromagnetic pulses that result in Very Low Frequency (VLF) [1] waves propagating along the Earth's magnetic field lines. Due to the dipole shape of the geomagnetic field, these waves travel upward from the stroke location out through portions of the plasmasphere and back to the Earth's surface at the field line foot point in the opposite hemisphere. VLF antenna receivers set up at various high and middle latitude locations can detect whistler waves generated by these lightning strokes.

The propagation time delay of these waves is dependent on the plasma density along the propagation path. This enables the use of whistler wave observations for characterising the plasmasphere in terms of particle number and energy density. The dynamics of energetic particle populations in the plasmasphere are an important factor in characterising the risk to spacecraft in orbit around Earth.

Annual global lightning flash rates are on the order of 45 flash/s [5]. The resulting high occurrence rate of whistler events makes it impossible to identify and characterise them in a reasonable time. Therefore the automatic detection and characterisation of whistler are valuable to the study of energetic particle dynamics in the plasmasphere and to develop models for operational use.

Lichtenberger [1] developed an automatic detector and analyser based on the Appleton-Hartree dispersion relation and experimental models of particle density distribution. Recent advances in artificial neural network-based image processing methods – for example, convolutional networks [6] may be able to provide an alternative method for the automatic identification and characterisation of whistler events in broadband VLF spectra.

Model development is based on training a neural-network-based model on a set of spectrograms with whistler events identified by the nodes of the Automatic Whistler Detection and Analysis Network (AWDAnet [7]). Spectrograms will be presented in the form of images (Figure 1.1) to take advantage of the wide range of image-processing techniques available for this type of object identification.

[1] Very Low Frequency waves are those in the 3 to 30 kHz range.

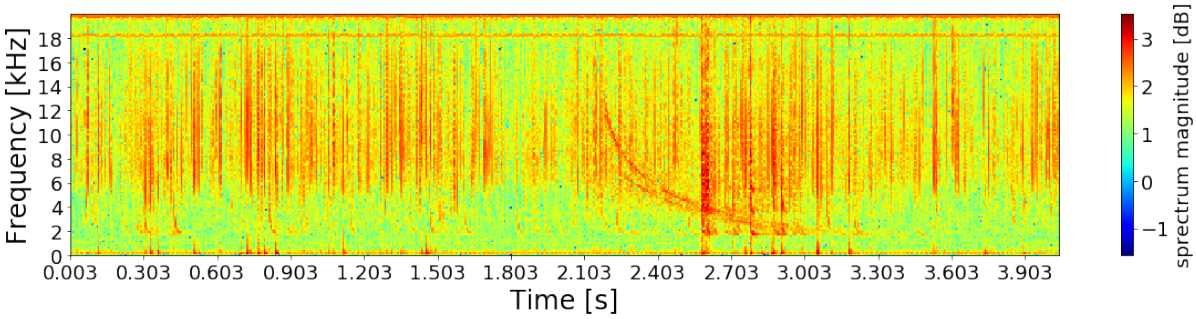


Figure 1.1: Example of whistler wave trace in spectrogram. This whistler starts at 2.2s and ends at around 3.1s

1.2 User Requirements

This project provides a proof of concept of the application of machine learning techniques to the problem of diagnosing the plasmasphere using the information provided by Whistler waves. The designs proposed in this dissertation focus on localising whistlers given a dataset of collected whistlers.

1.3 Objectives of Study

1.3.1 Problems to be Investigated

This project investigates the feasibility of finding and implementing methods - including the one proposed by Lichtenberger [1] - to localise whistler radio waves observed in the plasmasphere. This problem encompasses the following questions:

1. Is it possible to develop a new method in detecting whistlers different from the SOTA (State Of The Art)?
2. Can any of the new methods localise the start and end time of the whistler while providing a figure of merit on the localisation?
3. Do the new methods perform better than the current SOTA?

1.3.2 Purpose of Study

The purpose of this study is to develop and implement a few methods to localise any whistlers in given VLF spectra. By localisation, it is implied that the starting time and ending time of each whistler should be specified within a range of uncertainty.

1.3.3 Scope and Limitations

The scope of this project is to design and implement a method capable of localising whistlers in given VLF spectra.

A major limitation of this project is the availability and diversity of the data. The dataset used for this project has 2571 observations in total collected in 2013 with 2196 samples from Marion Island and 375 from the fourth base of the South African National Antarctic Expedition (SANAE IV).

Another limitation is the verification of the result of the proposed method which is limited by the labelled data provided by the SOTA. The SOTA provides a time location of each whistler at $5kHz$ and therefore not the starting point. Thus, it is difficult to use the SOTA as a benchmark for the evaluation of the starting and ending time of the detected whistler.

1.4 Dissertation Overview

Chapter 1 provides a background to whistler radio waves and to their crucial importance as an alternative economical method used in the diagnosis of the Earth's plasmasphere. Moreover, it brings forth the need of an automatic way of detecting and analysing these whistlers as a reason for this investigation of new whistler waves detection algorithm. Despite the limitations on the available data, such as lack of diversity (with the data only collected at two sites), the limited number of samples collected per sites (less than 2500 samples), and the poor quality of data collected a some site (such as SANAE IV), these methods can be extended to new data collected at different sites. The requirements for these new methods as stated by the user as well as the objectives of this study are presented. Each chapter of the dissertation is summarised to provide an overall understanding of the research.

Chapter 2 reviews the appropriate literature on the detection and localisation of whistler radio waves. This includes the origin and formation of such waves, their characteristics in the time and time-frequency domain, their modelling using the whistler dispersion model developed by Bernard [8], and their importance in diagnosing the Earth's plasmasphere. In addition, an overview of the current state of the art in whistler wave detection is provided. The literature also provides insight into different techniques. This covers some signal processing techniques such as cross-correlation used in target detection, an adaptive threshold based detector such as Constant False Alarm Rate (CFAR) and its derivatives. The literature provides the fundamentals of machine learning and some insight into a recent object detection technique known as You Only Look Once (YOLO) developed by Redmon et al. [2].

Chapter 3 provides insight into the methodology used in the investigation. This methodology involves firstly, collecting information from the literature. Secondly, understanding the project requirements by breaking them down into their analysis, system requirements, and qualification and acceptance strategy. Two requirements were imposed by the user, firstly, the system should be able to count the number and time occurrence of whistlers in a data file, and secondly, the system should be able to process any sample faster than the time it has taken to collect it. The third step in the methodology is the definition of the design procedures used for each proposed detector. And lastly, the selection of the appropriate software and hardware required for the investigation.

Chapter 4 focuses on the data provided for this investigation. An overview of the data collection process at different sites as well as the method of extraction of the raw vr2 data file is presented. This is followed by an explanation of the type of representation (time or time-frequency) used for this research. An in-depth statistical analysis of the samples collected at each site is performed followed by an assessment of the quality and quantity of the available data. The background of the method used to generate the labels associated with the data is presented, as well as its limitations. On the topic of the generation of the data, Chapter 4 presents a method used to simulate the whistler radio waves in their time-frequency domain. Lastly, a series of techniques used for preprocessing the data is presented as well as the their signal to noise ratio improvement.

Chapter 5 introduces the first whistler wave detector and localiser implemented in this research. This implementation uses cross-correlation of a preprocessed spectrogram with a simulated whistler generated using the Bernard whistler waves dispersion approximation to generate a time series representation of the similarities between the simulated whistler and the spectrogram. It continues with the introduction of a few detector methods such as CA-CFAR and its variations in the detection of section in the time series with high similarities with the simulated whistler. It also proposes a method of bounding the detected whistlers by cross-correlating an arbitrary bounding box with a series of simulated whistler. The chapter ends with the presentation and discussion of the results of this detector on a sample from the training set.

Chapter 6 introduces two new whistler wave detectors and localisers. These two implementations use the power of supervised machine learning to tackle the object detection problem. The first design proposes a method that uses an image classifier to classify a few generated regions of interest extracted from the spectrogram representation of the data. This results in a probability detection over time of the whistler in the spectrogram which is fed to a detector in charge of selecting the appropriate probability threshold and peaks. The second design propose the use of an adaptation of the state of the art in object detection, YOLO, to detect all whistlers at once. This method treats the object detection problem as a regression and therefore, alleviate the need of manual generations of these regions of interest.

Chapter 7 provides the analysis of the performance of the new whistler wave detectors on the data. This chapter starts by defining the benchmark and metrics used in the evaluation of the detectors. The selection of the best detector for each design is selected by tuning the detector's parameters. The chapter ends with the evaluation of the new models in terms of performance metric and processing time on the data.

Chapter 8 is a reflection of the process and results of this investigation. A few recommendations on the design and implementation process are given for subsequent research.

Chapter 2

Literature Review

This chapter covers the literature of the whistler radio waves, from their formation to simulation. In addition, it provides insights on the current state of the art in whistler wave detection. Moreover, different techniques from signal processing to object detection and machine learning used in this investigation are presented.

2.1 Whistler Radio Wave

Whistler Radio Waves also known as Whistling atmospherics [9] are very low-frequency electromagnetic waves propagating along the magnetic field of the earth. Although the terminologies used to refer to the whistlers differ from one field to another[1], in the domain of Space Science (domain of interest in this project), a whistler is defined as a electromagnetic wave excited by lightning and propagating through the magnetosphere and the ionosphere [10]. On a spectrogram, they are characterised by a decay in frequency (typically from 10 kHz to 1 kHz) over time and can last several seconds.

2.1.1 Whistlers formation

Lighting strikes occur at an annual rate of 45 flashes/s and dissipates each 10^{12}W into heated air, thunder, and VLF radio waves also known as radio atmospherics or spherics for short [11]. The VLF radio waves generated by lightning propagate through the earth ionosphere. The ionosphere is a layer of the Earth's atmosphere where both "ions and electrons are present in quantities sufficient to affect the propagation of radio waves [12]". These waves, travelling through the ionosphere can be so strong that they could be detected all over the planet. The particularity of these waves is their frequency components covering the VLF spectrum with high intensity as seen in Figure 1.1. Since the earth and ionosphere act as conducting surface and are only a few VLF wavelengths apart, they both behave as a parallel wave-guide for the

[1] The terminologies are defined based on the excitation mechanisms causing the radio wave, on the sound produced by those waves (whistlers are in the audio frequency range), and the characteristics of their spectrograms. Excitation different from lightning result in "whistler-mode-wave" which are referred to as hooks, roar, risers, hiss, saucers, triggered emissions, chorus, etc . . . depending

VLF radio waves as illustrated in Figure 2.1. The waves propagating between the earth surface and the ionosphere are said to propagate subionospherically. Part of these waves escapes the bottom of the ionosphere and travel through the earth plasmasphere where they interact with the energetic electron of the radiation belt. Among those, some travels along the magnetic field line of the earth in a duct and can be detected on the other side of the hemisphere (Figure 2.2). These radio waves pass through the Earth-ionosphere twice (on exit and entry), then twice through the ionosphere, and once or many time through the magnetosphere. Upon propagation through the atmospheric layers of the Earth, the frequency of these radio waves decay over time as shown in Figure 1.1, and fall nicely into human-audible frequency range producing a whistling sound and hence the name whistler. When collecting data on VLF radio signal after lightning strikes, the VLF events can be observed on a time-frequency plot known as a spectrogram. An example of spherics and whistlers detected at Marion island in 2013 are shown in Figure 2.3

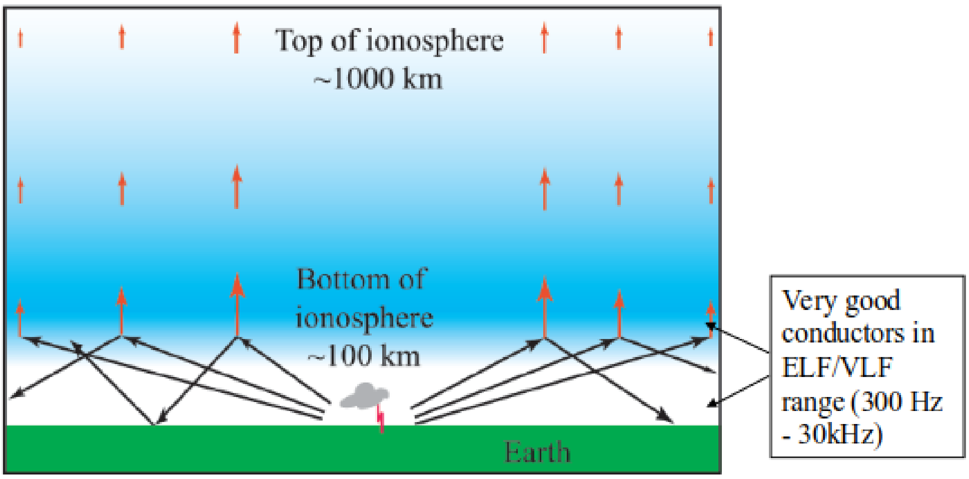


Figure 2.1: Path taken by the radiation emitted during lightning strikes [13].

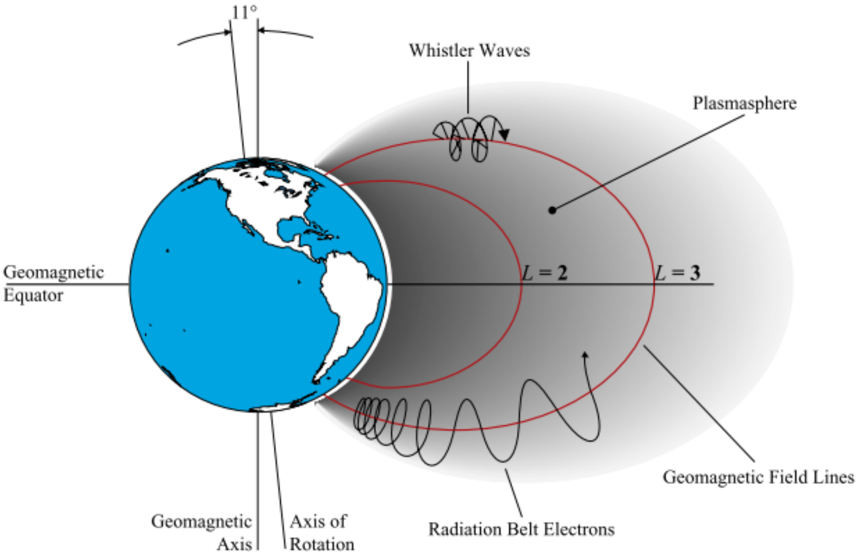


Figure 2.2: Whistler waves travelling in the earth ionosphere [13].

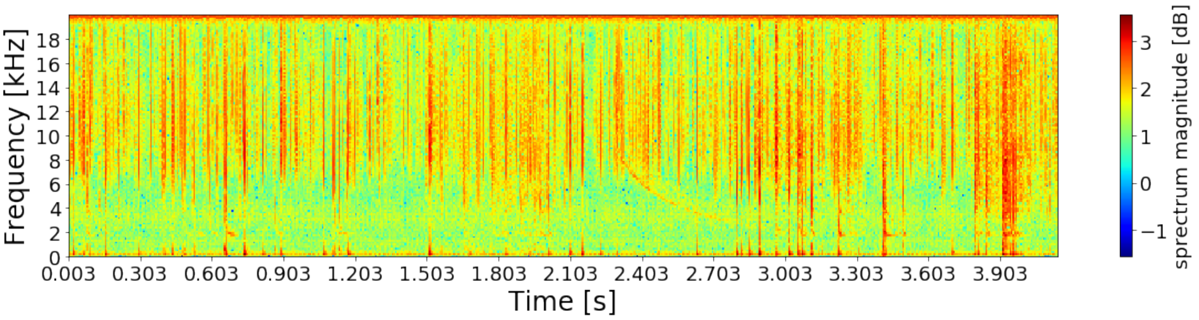


Figure 2.3: Spectrogram showing: a whistler starting at 2.2s and visible from 8kHz to 3kHz after 0.6s; many whistlers which appears as vertical line of high magnitude at random time; and one man-made signal at 20kHz spanning over the time range of the spectrogram.

2.1.2 Whistler Characteristics and importance

Whistlers are VLF radio waves that are characterised by the decay of their frequency over time. On a spectrogram as illustrated by Figure 2.3, a whistler takes the shape of an exponential decay signal lasting for few or several seconds [14]. Although the waves expand through most of the VLF spectrum, the whistlers are mostly observed below 20 kHz with their peak located between 3 to 6 kHz [13].

Since their discovery in the late 19th century, whistlers have been used to prove the existence of the protonsphere (now known as plasmasphere) [15] and later used to show that the propagation path delay of the whistlers and its plasma density are proportional [16]. Since then, whistlers have been used as a cheap method for diagnosing the Earth plasmasphere.

2.2 Automatic Whistler Detector and Analyser (AWDA)

Since the discovery of whistlers as a way to diagnose the plasmasphere, methods to analyse the whistlers have been developed. Prior to the work by Lichtenberger *et al.*, whistlers were detected and analysed manually. The process involved visual detection of whistler in raw data by skilled human and application of some models using scale *f-t* pairs to calculate the propagation and plasma parameters. In 2008, Lichtenberger et al. introduced a novel way for whistlers detection and analysis known as the Automatic Whistler Detector (AWD) whose purpose is to "(1) to automatically provide plasmaspheric electron densities extracted from whistlers and (2) collect statistical data for the investigation of whistler generation and propagation" [7]. Both detector and analyser were successfully implemented and performed with a 90% efficiency (only 10% of the whistlers in the data were not detected by AWD) and a 50 to 80% false detection efficacy rate (20 to 50% of the detection were not correct) [7].

Detecting a signal of known shape within another is a common problem that can easily be solved with correlating one signal with another. However, the spectrogram generated by the raw data contains noise in form of spheric, power line harmonics, VLF transmitted signals etc. To reduce the noise in the data, the AWD only uses whistlers in the range of 4.5 to 11.5 kHz since most of the whistlers recorded at mid-latitude span over this range. Cropping the spectrogram to this range, however, does not remove noise such as spherics which are superposed on the whistler signal. To reduce this form of noise, AWD uses a threshold method ”removing” all columns whose total power is higher than its threshold. The resulting cropped and filtered signal is shown in the middle image of Figure 2.5. The resulting spectrogram is then correlated with a simulated whistler shown in Figure 2.4 using a model generated by the Bernard approximation [8] to produce a 1D signal as the output of the detector. Finally, AWD uses an adaptive threshold to detect the presence of whistlers. The entire process is illustrated in Figure 2.5.

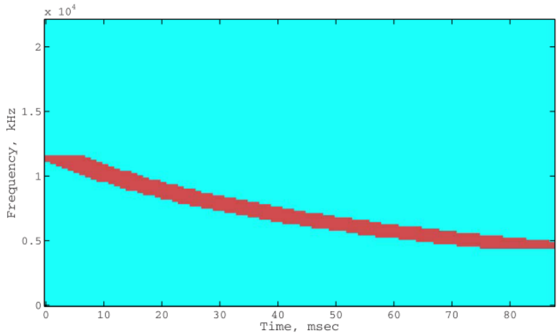


Figure 2.4: Whistler generated using the Bernard approximation and used by AWD [7] as a kernel for its detector.

2.3 Bernard Approximation Of Whistler Waves

The expression of the normalised travel time as a function of the normalised frequency is given by Equation 2.1 [8].

$$\frac{t}{t_n} = \frac{1}{2\sqrt{f/f_n}} \frac{(1+\Lambda_n) - (3\Lambda_n - 1)(f/f_n)}{1 - \Lambda_n(f/f_n)} \tag{2.1}$$

We know that the dispersion D of the whistler is given by:

$$D = t\sqrt{f} \tag{2.2}$$

therefore, the dispersion D_n at the nose frequency [2] is given by

$$D_n = t_n\sqrt{f_n} \tag{2.3}$$

The dispersion at the nose frequency as a function of the zero dispersion is given by

$$D_n = \frac{2D_0}{1+\Lambda_n} \tag{2.4}$$

Combining Equation 2.1, Equation 2.3, and Equation 2.4, we can derive Equation 2.5 which relates the whistler frequency range f, its time t, the zero-frequency or Eckersley dispersion D_0

[2] Starting frequency of the whistler

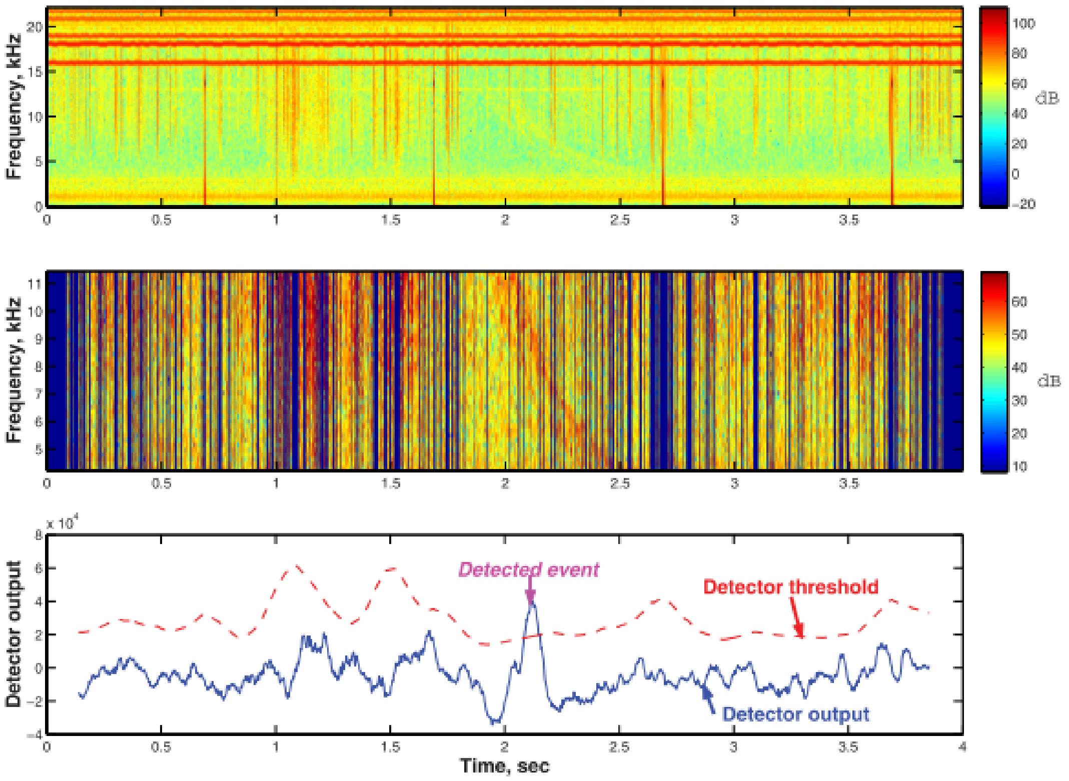


Figure 2.5: Summary of the process used by AWD in detecting whistlers [7]. (top) the generated spectrogram, (middle) the cropped and filtered spectrogram, (bottom) result of the correlation between the simulated whistler in Figure 2.4 and the cropped and filtered spectrogram.

[8], the nose frequency f_n, and the normalised equatorial electron gyrofrequency [3] Λ_n.

$$t = \frac{D_0}{(1+\Lambda_n)\sqrt{f}} \frac{(1+\Lambda_n) - (3\Lambda_n - 1)(f/f_n)}{1 - \Lambda_n(f/f_n)} \tag{2.5}$$

Equation 2.5 presents the time travel t of the whistler as a function of four parameters D_0, Λ_n, f_n, and f. D_0, Λ_n, f_n characterise the property of the whistler and are chosen experimentally. f is chosen based on the frequency range to be covered.

2.4 Cross-Correlation

In the domain of signal processing, cross-correlation refers to the measure of similarity between two signals over time. Let's assume two time signals $f(t)$ and $g(t)$, the mathematical expression of the cross-correlation between $f(t)$ and $g(t)$ is

$$c(t)(f \star g)(t) = \int_{-\infty}^{\infty} \bar{f}(\tau)g(t+\tau)\mathrm{d}\tau[17] \tag{2.6}$$

[3] frequency with which a charged particle (such as an electron) executes spiral gyrations in moving obliquely across a magnetic field

For discrete signal $f[n]$ and $g[n]$, the cross-correlation is defined as

$$c[n] = (f \star g)[n] = \sum_{m=-\infty}^{\infty} \bar{f}[m]g[m+n] \tag{2.7}$$

c is a function measuring the power of the signal g in f.
For two dimensional signals $f(m,n)$ and $g(p,q)$, their corresponding matrices are used instead. Let F and G the corresponding matrices of $f(m,n)$ and $g(p,q)$ such that F and G are respectively M by N and P by Q matrices. The full 2-D cross-correlation of F and G is the $M+P-1$ by $N+Q-1$ matrix C given by

$$C(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(m,n)\bar{G}(m-k,n-l) \quad , \quad \begin{cases} -(P-1) \leq k \leq M-1, \\ -(Q-q) \leq l \leq N-1 \end{cases} \quad [18] \tag{2.8}$$

This definition results in a matrix bigger than the original F matrix. To alleviate this issue, we define the valid cross-correlation C between F and G as

$$C(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(m,n)\bar{G}(m-k,n-l) \quad , \quad \begin{cases} 0 \leq k \leq M-P-1, \\ 0 \leq l \leq N-Q-1 \end{cases} \tag{2.9}$$

Furthermore, it is sometimes preferred to obtain a 1D matrix as result of the cross-correlation. In such cases, the matrix F and G must be respectively of size M by N and P by Q with $M = P$. This result in a matrix C defined as;

$$C(l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(m,n)\bar{G}(m,n-l) \quad , \quad 0 \leq l \leq N-Q-1 \tag{2.10}$$

2.5 Constant False Alarm Rate (CFAR)

Constant False Alarm Rate (CFAR) detectors are commonly used in the domain of Radar because of their ability to adapt their threshold based on different noise sources encountered [19]. A typical CFAR detector calculates the average noise power around the Cell Under Test (CUT), multiply this power by a constant obtained from statistical analysis of the targets and interference and make detection decisions based on this result and the power in the CUT.

CFAR detectors rely on the statistical characteristics of both interference and target plus interference to ensure a constant false alarm rate. In CFAR, one of these characteristics is that the interference is Rayleigh distributed in voltage [19]. Although the statistical characteristic is key in target identification in CFAR, CFAR detectors rely on the careful selection of the number of cells to be considered as noise and the number of guard cell around the cell under test. The latter is key in selecting the shape of the overall adaptive CFAR threshold, while the former only scaled the CFAR threshold. Consequently, a CFAR detector can be implemented even with no knowledge of the statistical characteristics of the target and interference, however, doing so will not ensure a constant false alarm rate.

2.5.1 Cell Averaging CFAR (CA-CFAR)

Cell averaging CFAR (Figure 2.6) is the most common CFAR detector and is used when the noise is homogeneous in the observed signal. The key parameters of CA-CFAR are the number of noise cells, the number of guard cells, and the scaled obtained from statistical analysis of the interference. The guard cells and noise cells should be chosen such that no targets are present in the noise cell, in other words, the number of guard cells plus one should be greater than the number of cells occupied by the target in the CUT.

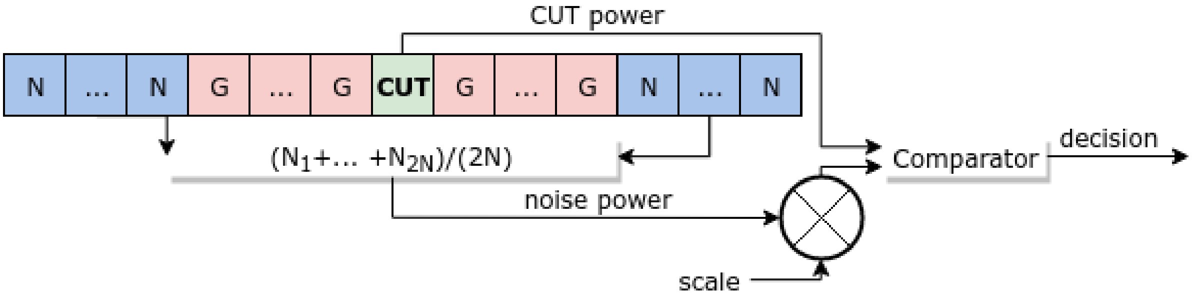


Figure 2.6: Block Diagram of CA-CFAR.

The scale factor is given by

$$T = N(P_{fa}^{-1/N} - 1) \tag{2.11}$$

with N, the total number of noise cells and P_{fa}, the probability of false alarm rate.

2.5.2 Order Statistic CFAR (OS-CFAR)

Order statistic CFAR (Figure 2.7) is used when there are many targets present in the CFAR window. Instead of calculating the average noise power, the power at the k^{th} ordered noise is used as noise power. When using OS-CFAR, k should be chosen such that $N - k \geq T$, where N is the number of noise cell, k the index of the noise cell to be selected, and T the number of expected targets.

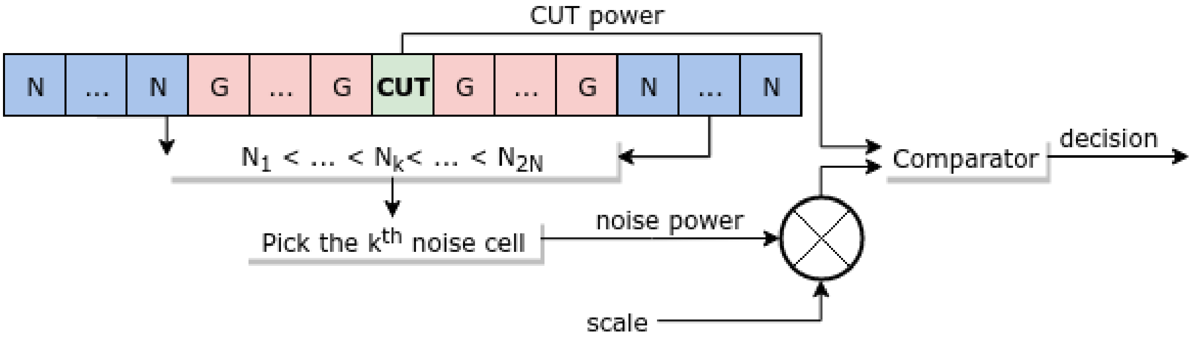


Figure 2.7: Block Diagram of OS-CFAR.

2.5.3 Trimmed Mean CFAR (TM-CFAR)

Trimmed Mean CFAR (Figure 2.8) ranks the noise and discards the T_1 smallest and T_2 largest sample before calculating the average noise power [19]. T_2 is chosen based on the assumption that the highest noise power are interfering targets. T_1 is chosen to remove clutter edge.

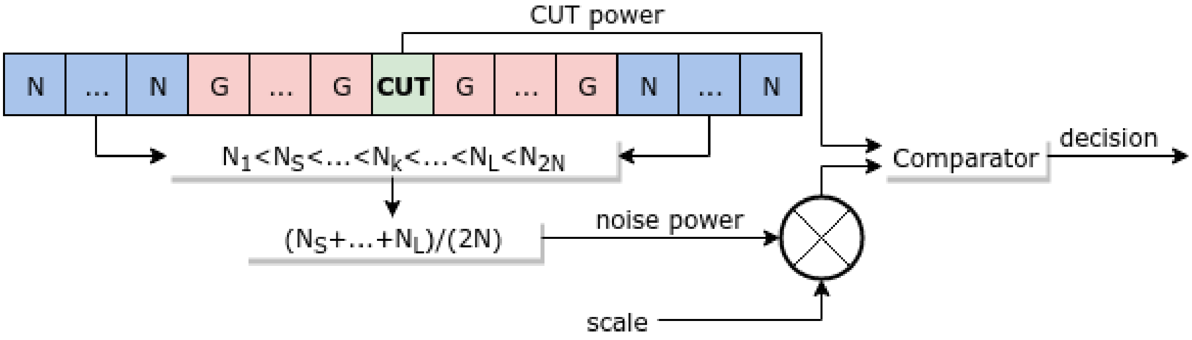


Figure 2.8: Block Diagram of TM-CFAR.

2.5.4 Linear Fusion CFAR detectors

A Linear Fusion CFAR detector (LF CFAR) as described by Ivkovic *et al.*[20] combines the decisions of three CFAR detector (CA CFAR, OS CFAR, and TM CFAR) to decide on the presence of targets in the CUT. LF CFAR makes a new decision by using the logic described in Figure 2.9.

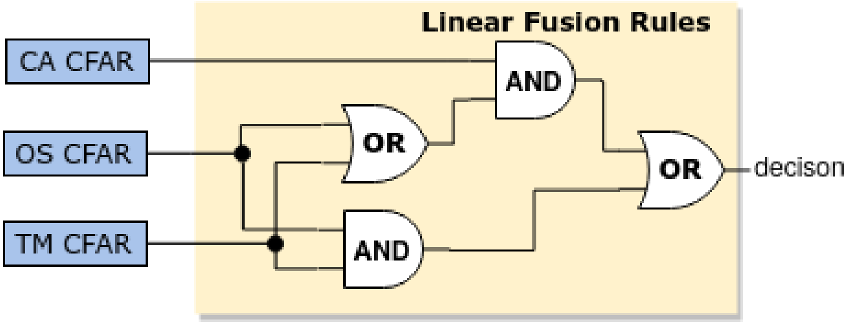


Figure 2.9: Block Diagram of LF-CFAR.

When CA CFAR is 0, there is a possibility that this is due to multiple interfering targets or strong clutter. Thus an AND logic is applied between OS CFAR and TM CFAR. When CA CFAR is 1, there is a possibility of false alarm caused by either a change in the clutter features or multiple targets interference. Consequently, an AND logic is applied on the CA CFAR output and the result of the OR logic between the output of OS CFAR and TM CFAR since both are designed considering multiple targets interference.

2.6 Fundamentals of Machine Learning

Machine learning is the science of programming computers so that they can learn from data [21]. An engineering-oriented definition of machine learning is given by Tom Michell [22] as "A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.".

According to Tom [22], three elements must be defined. The task T to be performed, the experience E accumulated by the learner, and the measure P of evaluation of the learning process. With advancements in the domain of machine learning, variations of P, T, and E have been developed and work best for certain tasks. Without going into much detail of decades of research, only a few aspects of the machine learning literature related to this investigation will be further elaborated.

Task T influence the architecture of the machine learning model. Common architectures harness the power of Artificial Neural Networks (ANN) . ANN is an information paradigm inspired by the way the brain processes information. ANNs are composed of many interconnected perceptrons that learn from given data to predict the outcome of unseen data. Figure 2.10 shows the general model of a perceptron.

A perceptron is a single-layer neural network associating an input vector to an output value using some weight and biases. Let's x be a vector consisting of n+1 axis (x_0,x_1,. . . ,x_n) and w a vector like x representing the associated weights (w_0,w_1,. . . ,w_n) to x, and finally b a bias (bias weight). A perceptron firstly calculate the dot product of x and w and add the bias value ($\Sigma = b + x.w = b + x_0w_0 + x_1w_1 + ... + x_nw_n$). The sum is then passed through an activation function (f) whose roles among others are to add non-linearity to the ANN and sometime restrict the output to a range (0 to 1 for probability).

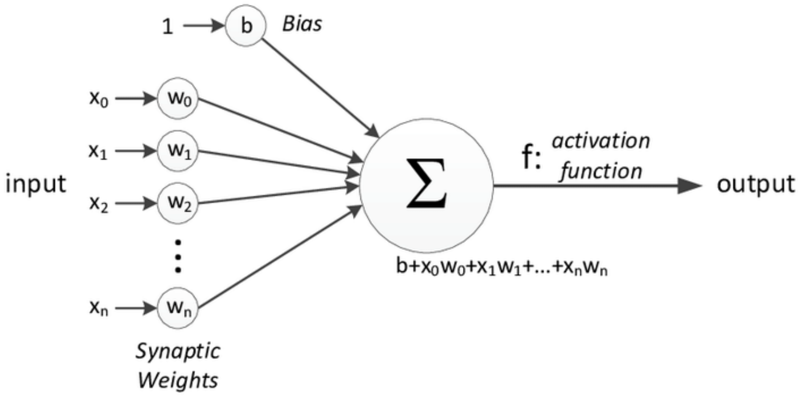


Figure 2.10: Block Diagram of a perceptron [23].

A complete ANN uses one or many layers of perceptrons, with each layer, having multiple perceptrons all interconnected to their inputs and the perceptrons from the following layer.

Another architecture is the Convolutional Neural Network which introduces the use of convolutional layers. These layers use a convolution of the input matrix and a few kernels to generate new layers. The purpose of these convolutions is to help the machine learning model to learn spacial

features of the input. These networks are thus preferable when dealing with images [2].
The second element of the definition given by Tom is the experience E. E implies that the computer program learns something about the data. In some cases, the data is labelled meaning that the outcome associated with each input data is known. In such a case, the model learns by minimising the error between the predicted and the actual outcome using backpropagation and an optimiser.
The last element is the performance measure P of the model. The performance measures of the model are defined by some metrics (see Chapter 7). These metrics are used to tune the training parameters of the network to achieve the best performance.

2.7 You Only Look Once (YOLO)

The field of computer vision has made significant progress since the development of Convolutional Neural Network. Among the different problems it focuses on, object detection has probably benefited most from CNN. In object detection, we try to find objects in an image by drawing bounding boxes around the region of interest (RoI) . As opposed to the standard CNN, object detection requires a variable number of outputs as the number of objects in an image is not known. Few CNN, such as Region-based CNN (R-CNN [24], Fast R-CNN [25], Faster RCNN [26]) and single-shot detectors (YOLO[2], SSD[27]) have been created to tackle this issue. Region-based CNNs require the generation of some RoI which are fed to one or more CNN. As a result, these methods do not look at the complete image but only the region with a high probability of containing objects. Single-shot detector treats object detection as a regression problem. As of 2018, the current state of the art, YOLO (You Only Look Once), uses a single CNN to predict the bounding boxes and the class probabilities [2]. Unlike R-CNN, YOLO perceives the image in its entirety during training and testing and encodes the class location and information. In YOLO, every input image is divided into $S \times S$ cells. A cell is responsible for an object if its centre falls within the cell. Each cell generates B bounding boxes (x, y, w, h) with their confidence score c which is a measure of the validity of the predicted bounding box (Figure 2.11). Moreover, each cell predicts C conditional class probabilities. In total $S \times S \times (B * 5 + C)$ tensors are used per image to make the final prediction.

2.8 Summary

This chapter provided insight into the literature covered for this research. It started by informing the reader of the properties of whistler radio waves, their origin, and characteristics. The current state of the art in whistler radio waves detection developed by Lichtenberger et al. [7] was also introduced. Moreover, the approximation made by Bernard [8] was rewritten to find a frequency-time approximation of the dispersion of the whistlers based on their nose frequency, their zero dispersion and the normalised gyrofrequency for all L-shells. A few techniques in target detection using signal processing, such as cross-correlation and constant false alarm rate detector were covered. The literature ended with an overview of the fundamentals of machine

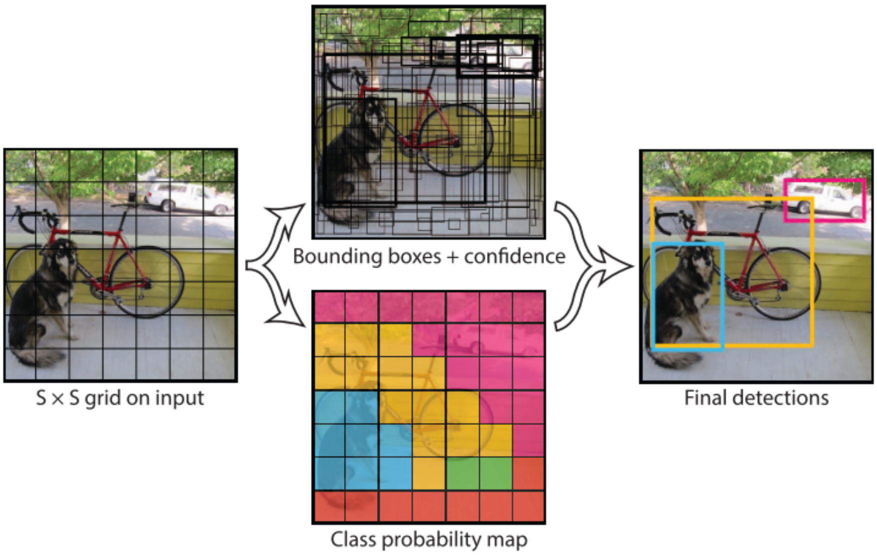


Figure 2.11: YOLO model [2].

learning and YOLO as of 2018 which is the very known for its overall great performance in object detection.

Chapter 3

Methodology

In this chapter, the methodology used in this study is presented. The methodology provides insight on the literature outlined in Chapter 2 by analysing the project requirements to deduce the traceability table, defining the design procedure and selecting the appropriate tools for this investigation.

3.1 Project Requirements

The alternative method for whistlers identification is intended to detect and localise whistlers within any given raw VLF data in reasonable time. In our context, detection means finding the number of whistlers in the raw data, while localisation means providing a bounding box coordinate of the location of each whistler in a frequency-time plot of the raw data. Following the criteria of AWD, the detection and localisation should be done faster than the collection time of the data on an average computer.

3.1.1 Requirement 1: Whistler Detection and Localisation

The system should be able to count the number and time occurrence of whistlers in a data file.

Requirement Analysis

The particular time-frequency dependence of the received whistler wave enables the estimation of electron density in the plasmasphere region of the Earth's magnetic field. The data files provided by the South African National Space Agency (SANSA) contain at least one target per observation. The whistlers can be visually seen on the data file spectrogram as curved lines in the spectrogram in Figure 1.1. So far, the AWD algorithm is capable of detecting the whistlers in a data file. The new method will be developed based on a set of whistlers identified by the Automatic Whistler Detection (AWD) algorithm developed by Lichtenberger (2009).

System Requirements

The system requirement for requirement 1 (R1) are tabulated in Table 3.1. R1 entails that a software must be written that given a data file, outputs the location (starting time and duration) of whistlers as well as a figure of merit on the accuracy of the detection.

Table 3.1: System requirements for requirement 1 (R_{1x} where x is a sub-requirement)

Requirements	**Description**
R_{11}	Create software that given a data file (.vr2 file) ouputs a list of detected whistlers
R_{12}	Each whistler in the output must be followed by a start and end time
R_{13}	Each whistler event in the output file must have a figure of merit for its accuracy

Qualification and Acceptance Strategy

Since the system will be trained from the detection made by the AWD algorithm, the accuracy of the detection of the system will be tested by comparing its performance to the one of the AWD on a large set of data.

3.1.2 Requirement 2: Real time

The system should process any data faster than the generating time of the data, *i.e.* a data of length t should have a processing time of $t_{processing} \leq t$.

Requirement Analysis

Global annual average lighting strokes are about 45/second. Ideally, the system should be able to process 45 data files per second. The focus of this project is to evaluate the performance of machine learning compared to the existing AWD method, therefore, real-time processing is not a point of emphasis. It is assumed that a report on the whistlers occurrence is required at the end of every day, therefore, a 24 hours delay is assumed to be the real-time of the system [1].

System Requirements

The system requirement for requirement 2 (R2) is tabulated in Table 3.2. The system has 24 hours to process the data collected in a day, therefore, it is assumed that the average processing time for a sample of duration t should be less than t.

[1] Real time implies an known duration of the program execution.

Table 3.2: System requirements for requirement 2

Requirements	Description
R_2	Data processing time of a collected sample should be less than its collection time

Qualification and Acceptance Strategy

The system is to be implemented as a computer program. The CPU time will be used to determine the processing time $t_{processing}$ of the detection. The acceptance strategy consists of evaluation the processing ratio defined as:

$$r = \frac{t_{processing}}{t}$$

with t, the duration of the sample. A ratio less or equal to one implies that the processing time is smaller than the collection time of the sample.

3.1.3 Specifications

The specifications of the whistler detector and localiser are defined in Table 3.3. This research focuses on the ability of alternatives methods to detect whistler radio waves. Finer detection resolutions are not of interest since the focus is on the presence of the radio waves. Therefore, the resolution of the detector should be 100 milliseconds per detection. The AWD achieved a misdetection of 10% and a false alarm rate of between 20 to 50% [7], the goal in this research is to achieve less than 20% for both misdetection and false alarm rate. Finally, the detection should happen in real-time, which implies that the processing time should be less than the collection time.

Table 3.3: System specifications

Specification	Description
S_1	The resolution of the detection should be 100ms per detection
S_2	The false alarm and misdetection rate of the whistlers should be less than 20%.
S_3	The detection of whistlers from a data file should be done faster than the time duration of the data

3.1.4 Acceptance Test

The list of acceptance tests is shown in Table 3.4. The resolution of the detection is a design choice and cannot be compared to any benchmark. The resolution can be double-checked by ensuring that the result of the detection only has one decimal number. The dataset must be divided into a training and testing set. The evaluation of the performance of the detectors must be done on the testing set which has not been used during the development of these detectors. Computer's clocks are quite reliable, therefore, the time taken for the detector to process a sample on the architecture of choice will be used.

Table 3.4: Acceptance tests

Acceptance Tests	**Description**
AT_1	The starting and ending time of detected whistlers should have one decimal number
AT_2	The algorithm will be verified using datasets not used for training the system. The performance on the datasets will be the result used in this test
AT_3	The time taken to run the program on the architecture of choice will be used for this test

3.1.5 Traceability Table

The traceability table is shown in Table 3.5. This table maps each requirement with its specification and its acceptance test.

Table 3.5: Traceability table

Requirements	**Specifications**	**Acceptance Tests**
R_1	S_1	AT_1
	S_2	AT_2
R_2	S_3	AT_3

3.2 Design Procedure

This research involves the design and implementation of various techniques to detect and localise whistler radio waves in a raw VLF data file. Despite each technique having different design intricacies, there is a common theme shared among them. Prior to starting the design, the available data is analysed using some statistical evaluation. It is also shuffled and randomly divided into two sets, namely, the training set and the testing set. The training set is used during the design process of each detector while the testing set is used only once to evaluate each final design. Each design starts with an overview of the detector with a clear flowchart of the training and testing pipeline. The different building blocks forming the pipeline are analysed and the results of the final detector are shown for the same sample.

3.3 Tools and Setup

The current SOTA (AWD) is deployed on an x86 PC (PIII 600MHz with 256MByte RAM is the minimum, P4 with 512M RAM is preferable for later extension). To harness the power of new techniques such as machine learning, it is required to use a GPU. This investigation was performed on a GPU server hosted by the Chemical Department at the University of Cape Town. The server runs on a 5.3.0-24generic x86_64 Linux kernel 24GB of RAMs, 8 x 2726MHz quad-cores Intel core I7 960 CPUs and 2 NVIDIA GM200 GPUs.

Python has many libraries for data analysis, data and signal processing and more importantly machine learning so was the language of choice for this investigation. To visualise the effect of processing the data easily, a Jupyter Notebook using Python.

Few basic Python libraries such as Scipy, Matplotlib, and Numpy are used for processing the data. For machine learning, Keras was the library of choice since neural networks can be trained and evaluated in few steps.

The entire implementation of the detectors can be found on Github.

3.4 Summary

This chapter presented the methodology followed in this research. For this investigation, the literature around whistlers and the different techniques used for their detection and localisation were reviewed. This was followed by refinement of the project requirements into subsystem requirements, specification of these requirements and their acceptance tests. An overview of the design procedure was provided with a few insights into the division of the available data for training and testing the new detectors. Lastly, the selection process of the appropriate python libraries was detailed.

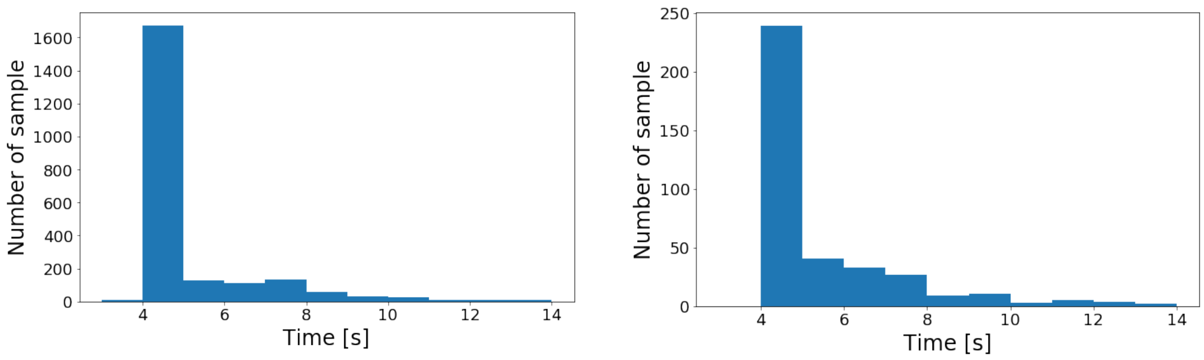
Chapter 4

Whistler Data

This chapter focuses on the data provided for this project. It starts by giving a background of the data collection process, followed by the choice of the type of data representations, the different data partitions, the statistical analysis required before processing the data, and the label used for the data. Furthermore, it provides insight into the simulation of the data using Bernard whistler's dispersion approximation. Finally, we propose a few pre-processing techniques to increase the signal to noise ratio in the data.

4.1 Data

The development of the whistler detector method presented by Lichtenberger et al. [7] led to the creation of a global network of Automatic Whistlers Detector and Analyser (AWDA) known as Automatic Whistlers Detector and Analyser Network (AWDAnet) which is made of AWD nodes. Each AWD node or system is composed of four elements, an antenna and a preamplifier which are site-dependent and a VR2 VLF sampler, and the AWD software running on a computer. The VR2 VLF sampler is a high-precision sampler unit controlled by an analogue device ADSP BF352 DSP processor. Its sample rate varies from 2kHz to 200kHz and its sample have a maximum value of 32765 (only for high spherics). The system collects data hourly and processes it using the AWD software to detect whistlers and generates an output file for the detection that is 4 seconds long (2 seconds before and after detection), however, it can be extended if multiple events at proximity are found (the 2-2 seconds margin is kept). The distribution of the duration of the data provided for this research is presented in Figure 4.1.
The AWD system collects and saves the data as vr2 files which are set of frames with each frame consisting of 4103 words (7 words for headers and 4096 words of data).
The data provided for this research was collected at two nodes, one at Marion Island in 2013 (2196 files with data sampled at 40kHz) and another at SANAE IV in 2012 (375 files with data sampled at 20kHz). Each data file contains the value of the signal received by the North/South (NS) and East/West (EW) pointing orthogonal loop antennas used at the site.



(a) Marion

(b) SANAE IV

Figure 4.1: Duration of data collected at Marion and SANAE IV. The output files of the AWD are in most case around 4s long with only few file which are up to 14s long.

4.1.1 Data Extraction

The AWD system collects and saves the data as vr2 files. Each vr2 file contains a set of frames. Each frame has a 7 words header and 4096 words of data for both channels, totalling to 4103 words in a frame. Each word is 2 bytes long (one low (lo) byte and one high (hi) byte) using the least significant bit (LSB) format.

The frame's header has the following format:

word 0 Sync word 0xA116, it appears as 16 A1 displaying the header in hexadecimal format

word 1 year (lo byte, without century, i.e. 06 in 2006), month (hi byte)

word 2 day (lo byte), hour (hi byte)

word 3 minute (lo byte), second (hi byte)

word 4 Hardware timer lower 16 bits

word 5 Hardware time upper 8 bits (lo byte), hi byte = 0

word 6 Status word

The status word has the following format:

bit 15 GPS navigation (1=navigate, 0=no)

bit 14 . . . 10 unused

bit 9 CH2 sampling status (1=on, 0=off)

bit 8 CH1 sampling status (1=on, 0=off)

bit 7. . . 0 divider values

The sampling frequency of each vr2 file is obtained by

$$f_s = \frac{200kHz}{divider + 1}$$

with *divider*, the value of the first 8 bits of the status word. The data is collected by two channels and comprises both channel's data (2048 words each) such that for any two consecutive words, one belongs to channel one (North/South) and the other for channel two (East/West).

4.1.2 Data Representation

Time Representation

The data provided by the AWD nodes are on average 4 to 5 seconds long. Each sample value is within ±32765 bits (only strong whistlers reach those values) and was collected for each North-South and East-West polarity. The basic representation of the data is a plot of the value of each sample collected over time. Figure 4.2 shows the 16 bits ADC (Analogue to Digital Converter) values vs time of the data collected at Marion during June 2013. This data sample has a mean of -15, a standard deviation of 2555, a minimum value of -32586, and a maximum value of 31993. Its overall distribution is shown in Figure 4.3. The data contains signals of magnitude more than 5 times above the standard deviation. Even though the time plot of the data shows the presence of these signals as spikes and can be interpreted as spherics, it does not

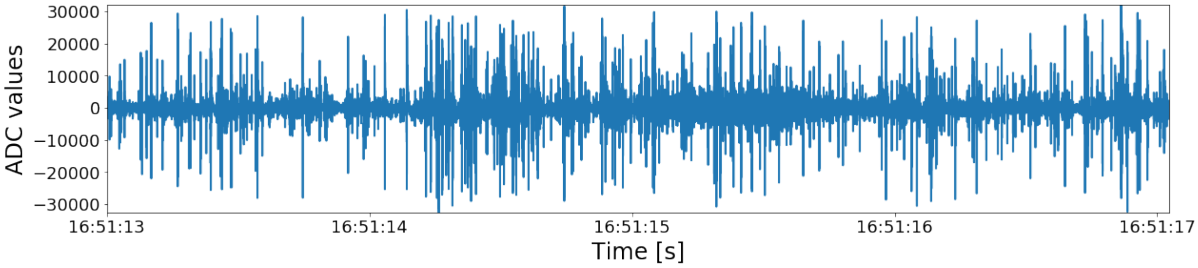


Figure 4.2: Raw Data collected by the AWD node at Marion island. The data shows the presence of strong spherics of magnitude above 20000.

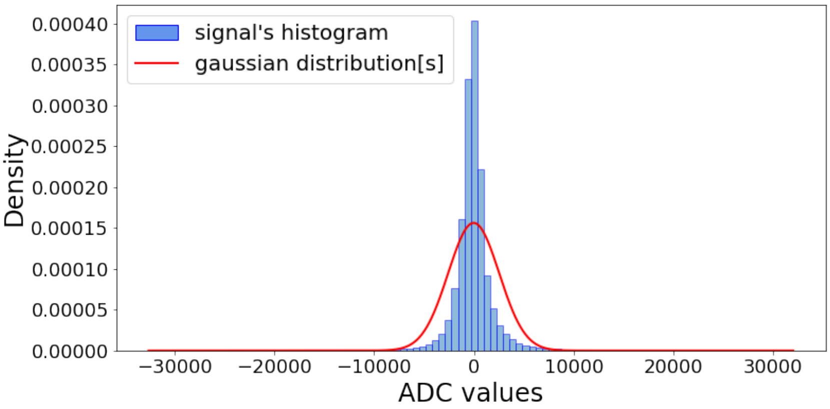


Figure 4.3: Histogram of the signal presented in figure 4.2.

provide a good indication of what is happening. While events like spherics are a burst of high amplitude signals at the VLF range and can thus be seen as a spike on a time plot, whistlers are signal which decays in frequency and can, therefore, not be easily seen on a time plot. A new representation of the data showing its frequency components is thus needed.

Time-Frequency Representation

The Short-Time Fourier Transform (STFT) is a succession of Fast Fourier Transform (FFT) on a windowed frame of a time series, where the window is slid forward on the frame over time [28]. The STFT thus provides a time-frequency representation of the signal also known as a spectrogram. As explained in Section A.1, the mathematical expression for the spectrogram is given by:

$$\hat{X}(f,t) = \sum_{t=0}^{Q-1}\sum_{n=0}^{N-1} x[n]w[n - t(M-K)]e^{-j\frac{2\pi f}{N}n}$$

A few parameters are available for tuning the spectrogram; the window size M, the number of samples K overlapping between two consecutive windows. The window size M affect the frequency and time resolution. The frequency resolution [1] of the spectrogram is more accurate as M increases while the time resolution is more accurate as M decreases. A trade-off between the frequency resolution and time resolution must be made. However, higher values of M results in a decrease in the number of sample in the time domain used to represent the signal. An example of spectrograms of the signal in Figure 4.2 is shown in Figure 4.4.

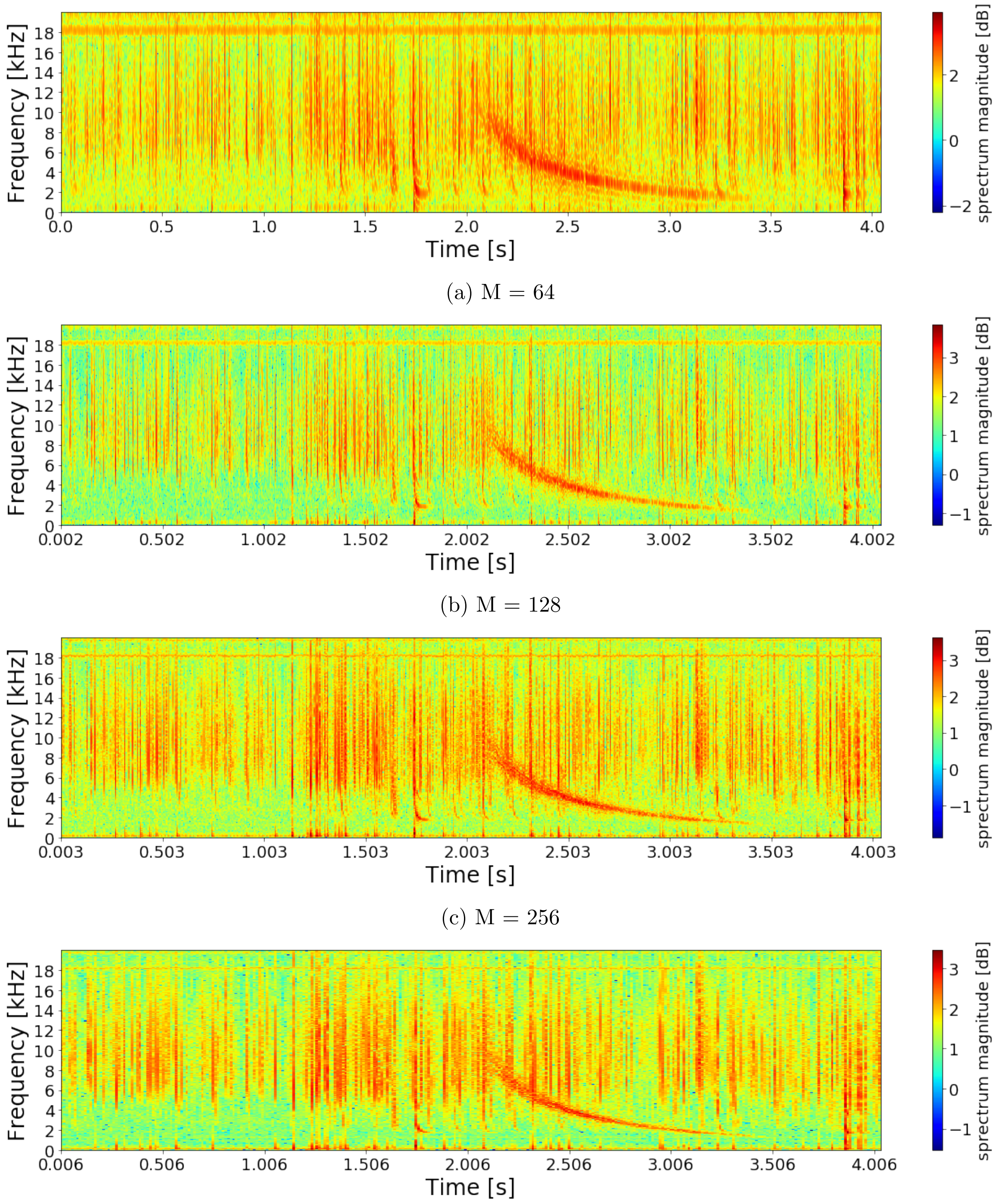
A window size of 512 produces an image with poor resolution and a one of 64, produces a blurry image. Visually, a window size of 256 is optimal for showing the details (whistlers) in the image. Moreover, the AWD uses this resolution for visualising the signal.

Another parameter to take into account is the overlap K. With this parameter, the number of time samples used to represent the signal is proportional to the value of K. Varying the value of K for $M = 256$ does not affect the quality of the image, therefore no overlap was used to generate the spectrograms.

The final frequency-time representation of the signal is thus computed with $M = 256$ and $K = 0$. Figure 4.4 shows the final representation of the signal in Figure 4.5. As opposed to the time representation of the signal, the time-frequency representation offers visual characteristics of the whistlers and the noise. A whistler lasting for 1.5s is observed at time $t = 2$. Some noise, such as spherics are seen in the form of vertical lines. Another noise coming are the man-made fixed frequency transmission at 18kHz.

The time-frequency representation of the data is more suitable when visualising whistlers and noise in the data and will be the representation of choice in this research. However, developing a technique using the time representation will be optimal as less processing time will be required.

[1] By frequency resolution, we do not refer to the number of sample in the frequency domain, we refer to the number of components represented in the frequency domain.



(a) M = 64

(b) M = 128

(c) M = 256

(d) M = 512

Figure 4.4: Spectrograms of the signal in Figure 4.2. (a) M=64 , the image is blurry and details of the whistlers are lost. (b) M=128, the image is clear and all noise and whistlers in the spectrogram are clearly seen. (c) M=256 , the image is clear and all noise and whistlers in the spectrogram are clearly seen. (d) M=512, the resolution of the image is poor and some details can be missed during processing.

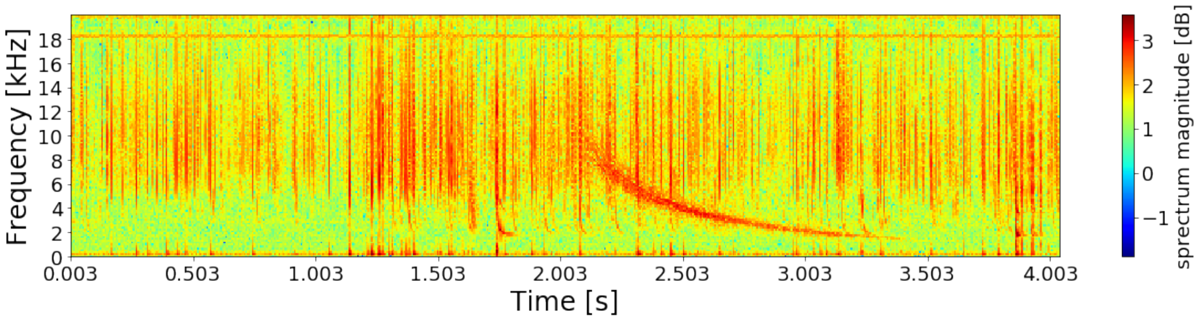
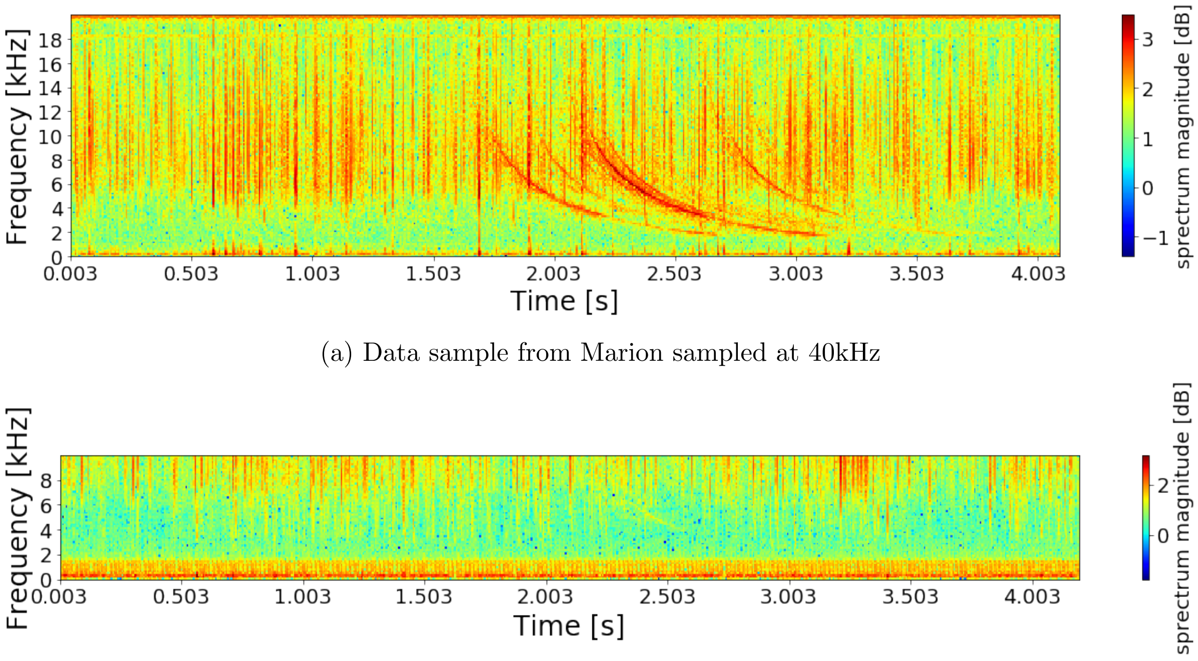


Figure 4.5: Spectrograms of the signal in Figure 4.2 generated a Tukey window with shape parameter of 0.25 and size 256 with no overlap, an FFT size of 256, showing the magnitude of the spectrum.

4.1.3 Data Partitioning

The data for this project comprises of 2196 samples from Marion Island and 375 from SANAE IV. Figure 4.6a and Figure 4.6b respectively display the spectrogram of one sample from Mario and SANAE IV. The data collected at Marion has frequency only up to 20 kHz while the one collected at SANAE IV has a maximum frequency of 10kHz. Unfortunately, the data collected at SANAE IV suffered from low SNR caused by a problem with the AWD amplifier. On the assumption that whistlers have the same characteristics and only the noise varies from one site to another, the SANAE IV data will not be used for the preliminary design of the whistler detector but for testing.



(a) Data sample from Marion sampled at 40kHz

(b) Data sample from SANAE IV sampled at 22kHz

Figure 4.6: Samples from data collected at Marion and SANAE IV

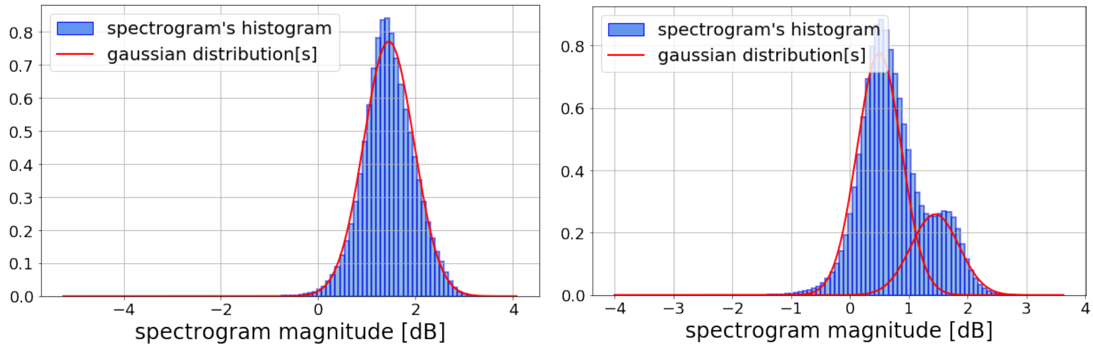
For the design of the whistler detector, the data at hand per site is split into two main portions, a training set and a testing set with a portion of 67% and 33% respectively. The numbers are broken down as follow:

- Marion: 1471 training samples, 725 testing samples
- SANAE IV: 251 training samples, 123 testing samples

This is done so that no mixing between training and testing set occurs to perform an out-of-sample testing. Note that splitting of the data was solely done at sample level and not at sample duration level to prevent having too few sample to work with.

4.1.4 Statistical Analysis of the Spectrogram

The time representation of the data has a Gaussian distribution with long tails as a result of the presence of strong spherics. The time-frequency representation of the data brings forth the presence of other noise in the signal such as transients and the continuous transmitted signal. This is observed in SANAE IV's training data as seen in Figure 4.6b by the high magnitude signal at frequencies between 0 and 2 kHz, and in Marion's data as seen in Figure 4.6a by the signal at 20kHz. Figure 4.7 presents the distribution of the spectrogram magnitude and their Gaussian Mixture Models for training data for Marion and SANAE IV. Marion's data can be fitted with one Gaussian distribution with a mean $\mu = 1.448$ and a standard deviation of $\sigma = 0.270$. As for SANAE IV's data, it can be fitted using two Gaussian distributions, one with a weight of $w = 0.264$, a mean of $\mu = 1.450$ and a standard deviation of $\sigma = 0.167$ and a second with a weight of $w = 0.736$, a mean of $\mu = 0.500$ and a standard deviation of $\sigma = 0.141$. SANAE IV's first Gaussian distribution with parameters $\mu = 1.450$ and $\sigma = 0.167$ differs from Marion Gaussian distribution by $\sigma = 0.002$ in their standard deviation. The main difference is the second distribution, which is the result of the strong presence of signals between 0 and 2 kHz in SANAE IV's data.



(a) Marion's data density distribution

(b) SANAE IV's data density distribution

Figure 4.7: Marion and SANAE IV data histograms fitted using a Gaussian Mixture Models. Marion's data distribution has a mean of $\mu = 1.448$ and a standard deviation of $\sigma = 0.270$. SANAE IV has a two Gaussian distributions, one with a weight of $w = 0.736$, a mean of $\mu = 0.500$ and a standard deviation of $\sigma = 0.141$ and a second with a weight of $w = 0.264$, a mean of $\mu = 1.450$ and a standard deviation of $\sigma = 0.167$

Generating the spectrogram distribution involves flattening it which removes the time-frequency characteristics of the whistlers. As a result, the distributions of the spectrograms do not provide enough information about the differences between the whistlers and other noise such as spherics which are present in the whistler frequency range. To tackle this challenge, pre-processing the data using the noise characteristics could be useful.

4.1.5 AWD Output

The data provided is pre-labelled using the Automatic Whistler Detector developed by Lichtenberger *et al.*. The detector provides results in the following format:

$$time : correlation\_result$$

where *time* is the 5kHz time of the detected whistler in UT time format, and *correlation_result* is a measure of the correlation between the simulated whistler and the portion of the spectrogram used for the correlation. This measure indicates how strong the correlation is, however, it does not provide a scale for identifying the best and worst correlation value.
An example of the AWD output for a sample from the Marion dataset is presented in Figure 4.8.

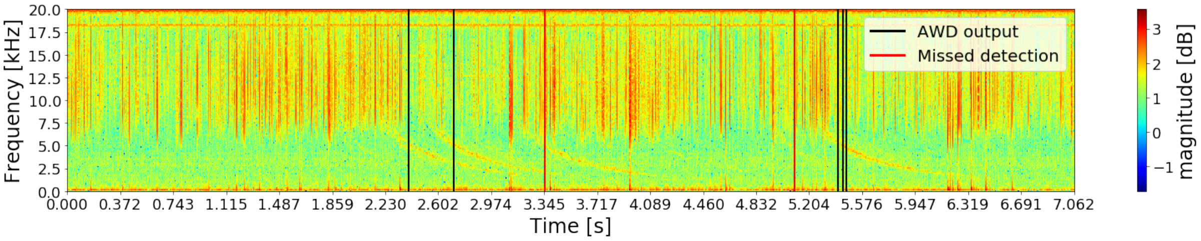


Figure 4.8: Spectrogram with output provided by Lichtenberger's Automatic Whistler Detector. The detector located five whistlers at t = [2.4, 2.71, 5.41, 5.44, 5.47] seconds.

The AWD detected five whistlers for the sample presented in Figure 4.8. Two whistlers are detected with correlations above 60. The last three detections have correlation ranging from 1 to 65 and correspond to whistlers not more than 0.6 seconds apart which are presumably the same whistler. Moreover, the AWD missed two whistlers located at 3.35 and 5.10 seconds.

4.2 Whistler Simulation

L. Bernard *et al.* [8] proposed a equation for approximating the whistler's dispersion. From this, an expression is derived relating the nose frequency f_n, the zero dispersion D_0, the normalised electron gyrofrequency λ_n, the time travel t and the frequency f of the whistlers in Equation 2.5. To simulate the whistlers, we make use of Equation 2.5 to generate a time-frequency representation of the whistler at the chosen parameters.
Equation 2.5 has four parameters:

- f: The frequency range of the whistler. Since most of the peak of the whistler are located between 3 to 6kHz, a frequency range is chosen such that 1kHz$\leq f \leq$ 10kHz.

- f_n: The nose frequency of the whistler. Lichtenberger *et al.* [7] selected a nose frequency of $f_n = 25$kHz, since our work is to be compared to theirs, this nose frequency is chosen.
- D_0: The zero dispersion of the whistler is chosen to match the one used by Lichtenberger *et al.* [7]. The dispersion is thus chosen such that $20s^{1/2} \leq D_0 \leq 80s^{1/2}$
- Λ_n: The normalised electron gyrofrequency is chosen such that $0.35 \leq \Lambda_n \leq 0.45$ as per Figure I in [8].

Figure 4.9 presents the whistler approximation at the boundaries of the variable parameters of Equation 2.5. For every frequency point, the time travel increases linearly as a function of the zero-dispersion with an increase in the zero-dispersion increasing in the time travel. The fastest dispersions occur at $D_0 = 20s^{1/2}$ and the slower ones at $D_0 = 80s^{1/2}$. The normalised electron gyrofrequency Λ_n slightly affects the time travel with an increase in Λ_n resulting in a decrease in the whistler's time travel.

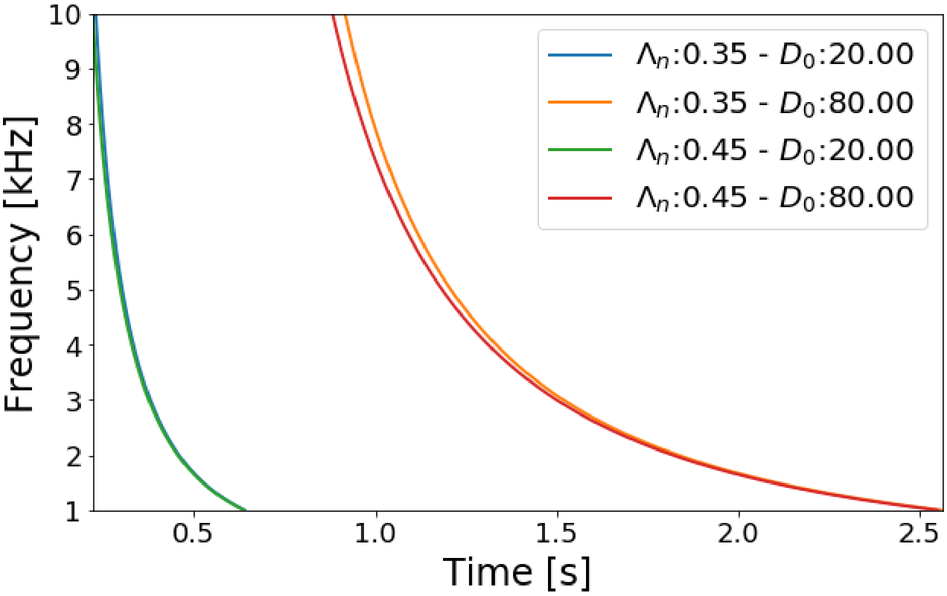


Figure 4.9: Whistlers' dispersion approximation with a nose frequency of $f_n = 25$kHz, and at varying zero dispersion and normalised electron gyrofrequency.

The approximations in Figure 4.9 are in two dimensions and must be converted to three-dimensional data so that they can be visualised as spectrograms. To achieve this, a three-dimensional array is created where the first and second axis represents the time and frequency obtained from the approximation. Since the approximation is based on time as a function of frequency, the time and frequency have the same number of sample. To fix this problem, the time and frequency are scaled based on the time and frequency resolution of the spectrogram of the whistlers data. The third dimension represents the magnitude of the whistler which follow the dispersion based on the scaled time and frequency. Figure 4.10 shows the spectrogram of the simulated whistler with $fn = 25$kHz, $D_0 = 20s^{1/2}$, $\Lambda_n = 0.35$, and 1kHz $\leq f \leq$ 10kHz.

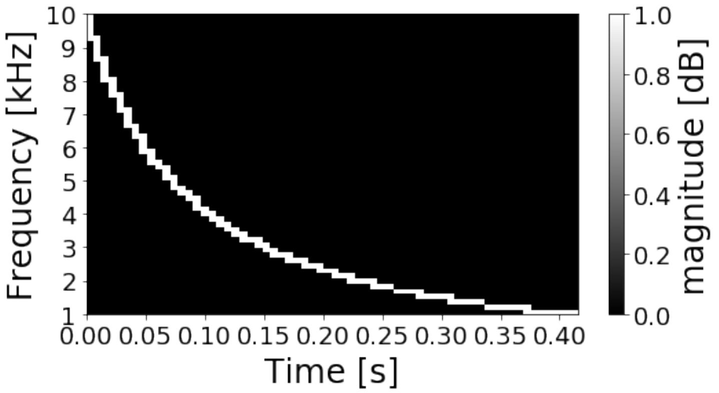


Figure 4.10: Simulated whistler with $fn = 25\text{kHz}$, $D_0 = 20s^{1/2}$, $\Lambda_n = 0.35$, and $1\text{kHz} \leq f \leq 10\text{kHz}$

4.3 Preprocessing

The primary purpose of preprocessing the data is to reduce the noise (whistler, VLF transmissions, transient signal, etc) in the spectrogram without removing any whistler information. Ideally, if the noise is removed, finding the shape of a signal (whistlers) in another (spectrogram) could be solved using simple methods such as matched filtering. To evaluate the efficiency of the pre-processing methods, a custom signal to noise ratio (SNR) is defined.

4.3.1 Signal to Noise Ratio

In this section, signals are defined as the portions of the spectrogram where whistlers are visible to the human eye [7], and noise, the entirety of the spectrogram with the portions of the signal zeroed to remove redundancies in the calculations. By definition, SNR is a measure of how much of the desired signal exists in background noise and is mathematically defined as the ratio between the signal power and noise power. Since it is difficult to quantify the signal and noise in the spectrogram, the signal power is defined as the sum of the square of all elements in the signals divided by the sum of the area covered by those signal, and is mathematically represented in Equation 4.1:

$$S = \frac{\sum_k^{signals} \sum_i^{W_k} \sum_j^{H_k} |x[i,j]|^2}{\sum_k^{signals} Area_k} \tag{4.1}$$

with

signals, the different regions bounding the whistlers in the spectrogram.

W_k and H_k, describing the maximum time and frequency index of elements of signal k in the spectrogram.

$Area_k$, the area of the region bounding signal k.

$x[i,j]$, the value of the spectrogram at position $(t = i, f = j)$.

As for the noise power, it is defined as the sum of the square of all elements in the spectrogram with no signal included, divided by the area of the spectrogram. It is mathematically represented by Equation 4.2.

$$N = \frac{\sum_i^W \sum_j^H |x[i,j]|^2}{HW} \tag{4.2}$$

with H and W, the length of the time and frequency of the spectrogram.
Thus the formula of the SNR is:

$$SNR = 10log_{10}(\frac{S}{N}) \tag{4.3}$$

Therefore to calculate the SNR of a spectrogram, the boundaries of each whistler in the spectrogram are first identified. Figure 4.11 presents the spectrogram of a sample collected at Marion in which the AWD detected three whistlers [2] at 2.5s, 2.9s and 3.0s. The boundaries of a whistler are based on three characteristics of the whistler. Firstly, the peak of the whistlers is located between 3 to 6 kHz. Secondly, whistler can last for 500ms to 1.7s. Lastly, whistlers decay in frequency over time. These three characteristics are used to create a boundary with frequency ranging from 2kHz to 10kHz, and with a time range of 0.8s for each whistler. Since the preprocessing techniques can scale the spectrogram's magnitude, each spectrogram is scaled using $\mu \pm 4\sigma$ dB with *mean* and σ obtained from the distribution of the spectrogram in Figure 4.7a before SNR calculation.[3]

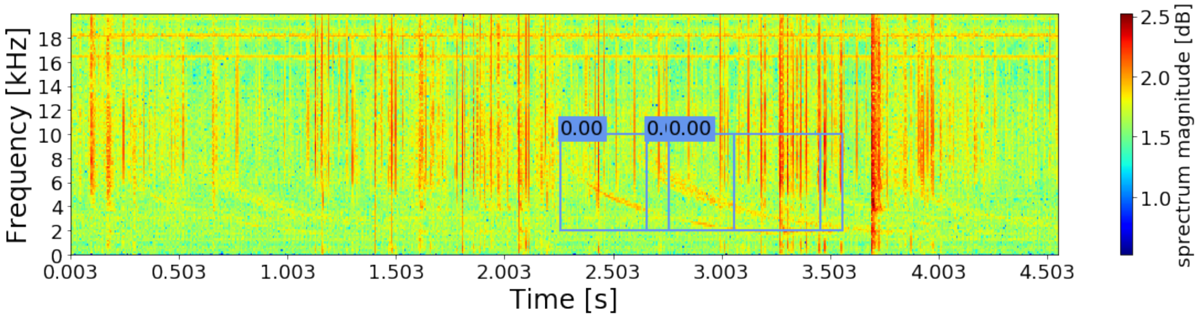


Figure 4.11: Boundaries chosen for calculating the SNR. The boundaries represented as $(t_{start}, t_{end}, f_{start}, f_{end})$ coordinates with t expressed in seconds and f in kHz are {(2.26,3.06,2,10), (2.66,3.46,2,10), (2.76,3.56,2,10)} and give a SNR of 1.166 dB.

The SNR for the spectrogram in Figure 4.11 is $SNR = +1.166dB$. This SNR, however, is not absolute. It firstly relies on an approximation of the location and boundaries of the whistlers which depend on the quality of the spectrogram and the intrinsic SNR of the signal. Secondly, the boundaries selected do not exclude the noise since spherics can overlap with the whistlers. This definition of the signal power is therefore not absolute. As a result, after pre-processing, the new signal power is also affected. However, this metric is good enough to show the effect of the different pre-processing method on the whistlers in the spectrogram.

[2] The AWD detected several whistlers between 2.4s and 2.5s and between 2.8s and 3s. We, however, group those whistler based on a 0.1 second interval, resulting in 3 whistlers located at 2.5s, 2.9s, and 3.0s.

[3] The spectrogram is converted back to linear scaling before SNR calculation.

4.3.2 Pre-processing visualisation

To visualise the effect of the pre-processing techniques on the spectrogram, we will work with frequency and time cuts of the spectrogram in Figure 4.11.

Figures 4.12 and 4.13 are respectively the frequency cuts and time cuts of the spectrogram in Figure 4.11.
The frequency cuts are taken at frequency $f = 5kHz$ and $f = 10kHz$ where a series of whistlers and spherics are observed. Whistlers have their peaks between 3 to 6 kHz, as a result, few regions of high magnitude can be observed on the 5kHz frequency cut around time $t = 2.50$ and $t = 2.90$. At 10kHz, the whistlers' magnitude significantly decreases, as a result, regions with high magnitudes on the 10 kHz frequency cut indicate the presence of strong spherics (1.4s, 1.8s, 2.4s, . . .). Before, pre-processing, the 10kHz cut has a mean of $\mu = 1.75$ while the 5kHz cut has a mean of $\mu = 1.67$ sitting below the 10kHz cut where the spheric is present.

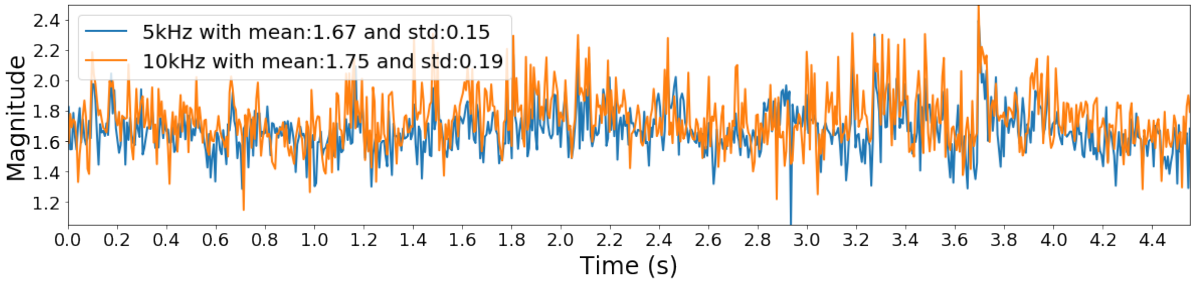


Figure 4.12: Frequency cuts at $f = 5kHz$ and $f = 10kHz$ of the spectrogram in Figure 4.11.

The time cuts in Figure 4.13 were taken at time $t = 2.5s$ where a whistler is observed and at time $t = 2.7s$ where a spheric is observed. On a time cut, since whistlers have their peak between 3-6kHz, a region of high magnitude is observed around 5kHz on the 2.5s time cut indicating the presence of a whistler. In these cuts too, the spherics have a higher mean that the whistlers.

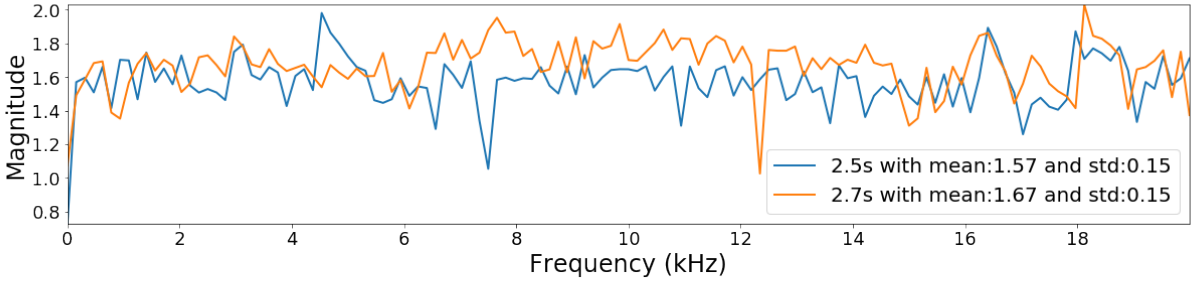


Figure 4.13: Time cuts at $t = 2.5s$ and $t = 2.7s$ of the spectrogram in Figure 4.11.

Ideally, what is expected after pre-processing is a decrease in the mean of the spherics compared to the mean of the whistlers. The time and frequency cuts will thus be used after each pre-processing method to visualise the effect of these methods on the spectrogram.

4.3.3 Z Score Transform

In the domain of statistics, the Z score or Standard score of a sample is the signed number quantifying how many standards deviation the sample is from the mean of the observed series [29]. If both the mean and standard deviation of a series are known, the z score is expressed as:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{4.4}$$

where

z_i and x_i respectively the z-score and the value of data point i

μ and σ, the mean and standard deviation of the data

and the resulting series has two interesting properties (see Appendix A.2 for derivations):

(1) a mean of 0

(2) a standard deviation of 1

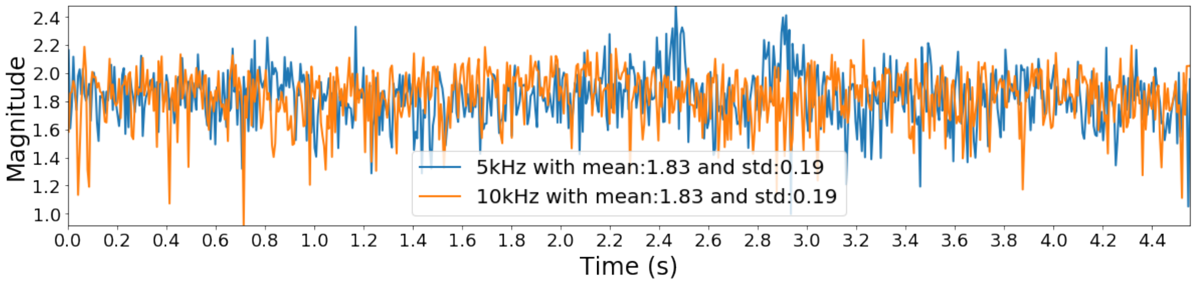
Using such a transform on the time-frequency representation of our data will only re-scale it since the transform itself is simply a linear transformation. However, using property (1) on each dimension on the data, any one-dimensional series (time series or frequency series) can be reduced to a series of mean 0 while preserving the shape of the series).
These are the observations we can make on the Z score transform:

(1) A point in any one-dimensional series which does not deviate much (within one standard deviation) from the mean of the observed series, is shifted to a region close to zero after application of the Z score transform.

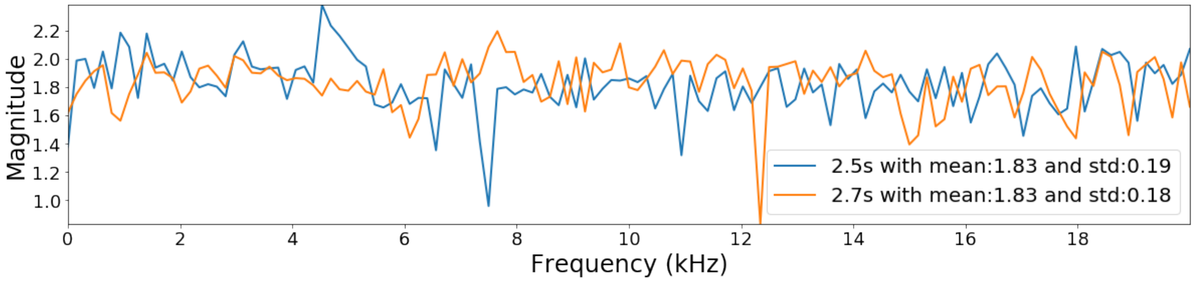
(2) All points in the series which do not deviate much from the mean of the series are shifted to a region close to zero after application of the Z score transform.

(3) A point which deviates much from the mean of the series deviates with the same way after application of the Z score transform.

Observation (2) implies that applying the transform on any time cuts containing spherics will remove their presence. Observation (3) implies that applying the transform on any time cuts containing whistlers will keep the whistlers. We, therefore, apply the transform on each frequency cut and then time cut of the spectrogram.
Figures 4.14a and 4.14b are respectively the scaled frequency cuts and scaled time cuts after applying the Z score on both axis of the spectrogram in Figure 4.11. The frequency cuts in Figure 4.14a have both been reduced to have the same mean. The region at 5kHz around 2.5s and 2.9s have the highest magnitude. This is an indication that the magnitude of the whistlers has increased compared to the noise/spherics. On the time cut, the 2.7s time cut has been reduced to have the same mean as the 2.5s time cut. As for the region around 5kHz, it has the



(a) Frequency cuts at $f = 5kHz$ and $f = 10kHz$.



(b) Time cuts at $t = 2.5s$ and $t = 2.7s$.

Figure 4.14: Frequency and Time cuts of spectrogram in Figure 4.11 at frequencies $f = 5kHz$, and $f = 10kHz$ and at times $t = 2.5s$ and $t = 2.7s$ after application of the Z score transform.

highest magnitude, confirming that the whistler magnitude has been increased as compared to the noise.

The spectrogram resulting from the application of the Z score transform on the time then frequency axis is shown in Figure 4.15. The spherics and transients present in the original spectrogram have been removed while the whistlers presented in Figure 4.11 are still visible. The application of the Z score transform results in an SNR of +1.219dB which is an improvement of +0.170dB.

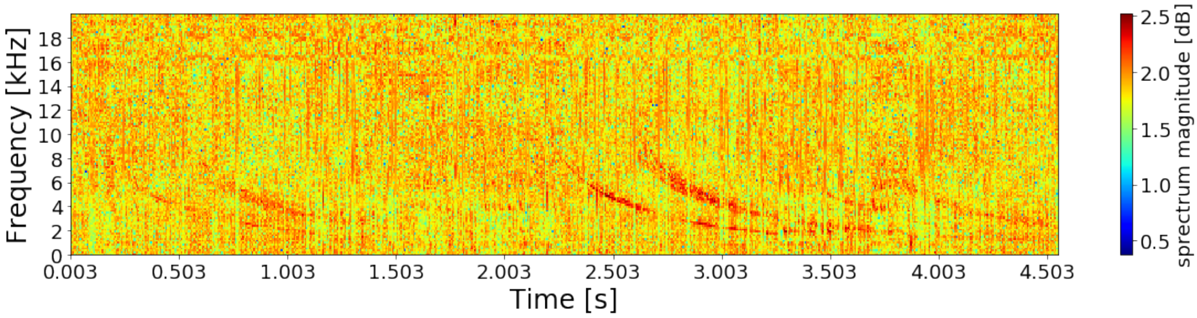


Figure 4.15: Result of the Z score transform on the spectrogram in figure 4.11 with an improvement of signal to noise ratio of +0.170dB.

4.3.4 Detrending

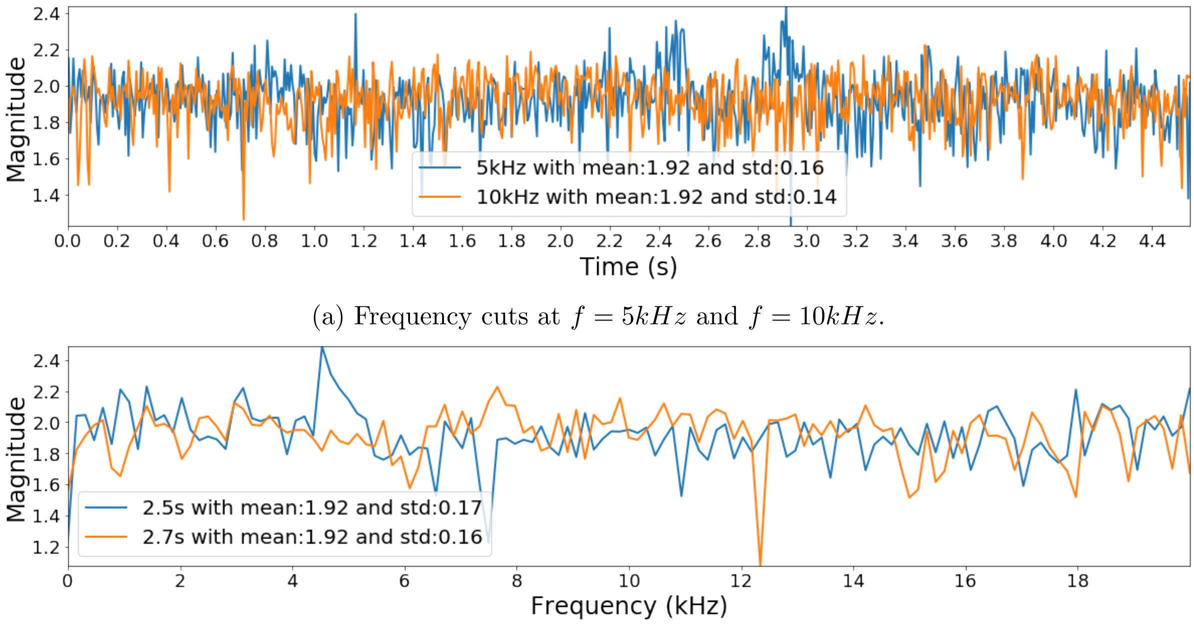
Detrending is the process of removing any known trend in a series. Similarly to the Z score transform, detrending can be used to reduce the presence of spherics and transient by either removing the mean or the linear least square of each frequency and time cut.

Reduction by the Mean

An approach to detrending is to remove the mean of the series from each element. Assuming that x is an element of the series X, the transform D_c which detrends a series by a constant c is given by Equation A.18.

$$x_c = D_c\{x\} = x - c \tag{4.5}$$

The series X_c such that $X_c = D_c\{X\}$ has a mean of $\mu_c = \mu - c$ and a standard deviation of $\sigma_c = \sigma$ (see Section A.3 for derivation). Detrending by the mean thus result in a series X_μ such that $X_\mu = D_\mu X$ with a mean of $\mu_\mu = 0$ and a standard deviation of $\sigma_\mu = \sigma$.
Figure 4.16a and 4.16b both shows the frequency and time cut after detrending the cuts in Figure 4.12 and 4.13 with detrending by the mean. As expected, the series have the same scaled mean. The standard deviation has changed, this is because each element of the series has been detrended twice, one from the x-axis and another from the y-axis. The spheric has been reduced as predicted and the peak observed is one of the whistlers located at around $5kHz$.



(a) Frequency cuts at $f = 5kHz$ and $f = 10kHz$.

(b) Time cuts at $t = 2.5s$ and $t = 2.7s$.

Figure 4.16: Frequency and Time cuts of spectrogram in Figure 4.11 at frequencies $f = 5kHz$, and $f = 10kHz$ and at times $t = 2.5s$ and $t = 2.7s$ after reduction by the mean.

Detrending the spectrogram by its mean results in an SNR of +1.166dB which is an improvement of +0.117dB.

Linear Least Square

Reducing each cut by its linear least-square fit as opposed to its mean does not necessarily centre the series at zero. This would be the case if the linear least-square fit has a gradient of

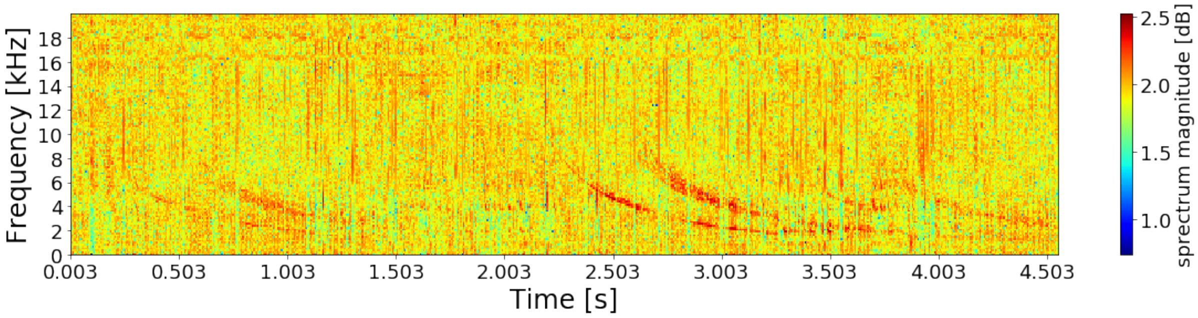
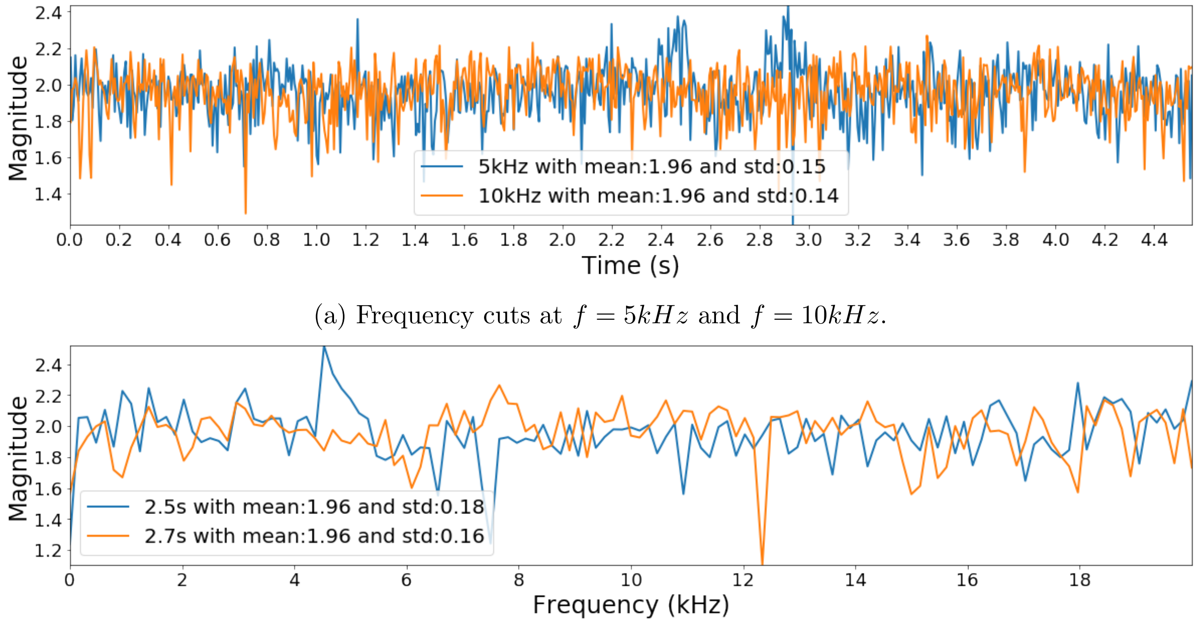


Figure 4.17: Spectrogram after detrending by its mean with +0.117dB increase in the SNR.

zero (constant through all increment). That could likely occur if the series contains spherics or transients. However, if the fit has a gradient, which is the case for series containing whistlers, the series is not centred to zero. It thus becomes inconvenient or difficult to find a simple relation between the result of the transform and the original spectrogram. Applying the transform on the cut in Figures 4.12 and 4.13 results in the cuts in Figures 4.18a and 4.18b.

Detrending the spectrogram using linear square fits results in an SNR of +0.903dB which is an improvement of -0.145dB.



(a) Frequency cuts at $f = 5kHz$ and $f = 10kHz$.

(b) Time cuts at $t = 2.5s$ and $t = 2.7s$.

Figure 4.18: Frequency and Time cuts of spectrogram in Figure 4.11 at frequencies $f = 5kHz$, and $f = 1 - kHz$ and at times $t = 2.5s$ and $t = 2.7s$ after reduction by a linear square fit.

4.3.5 Evaluation of the methods

To evaluate the performance of the pre-processing methods, all samples in the training set were pre-processed and the overall SNR per method was calculated. The results of these methods on the training set are shown in Figure 4.20. The overall SNR of the training set after applying

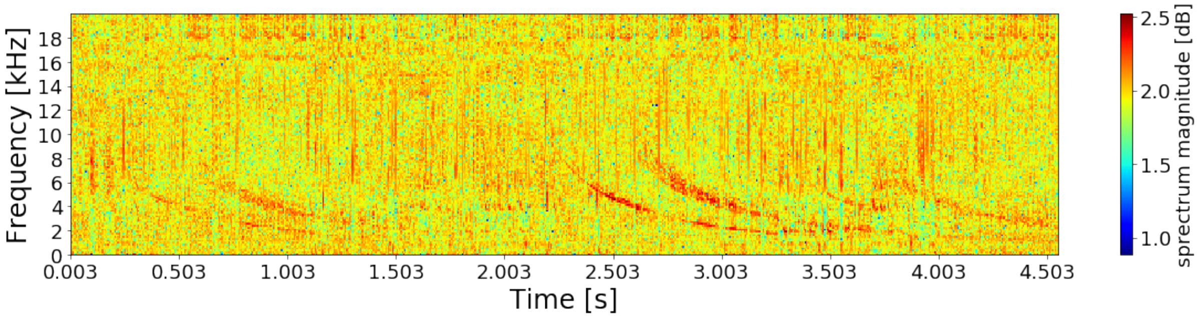


Figure 4.19: Spectrogram after detrending by its mean. An improvement of SNR by -0.145dB

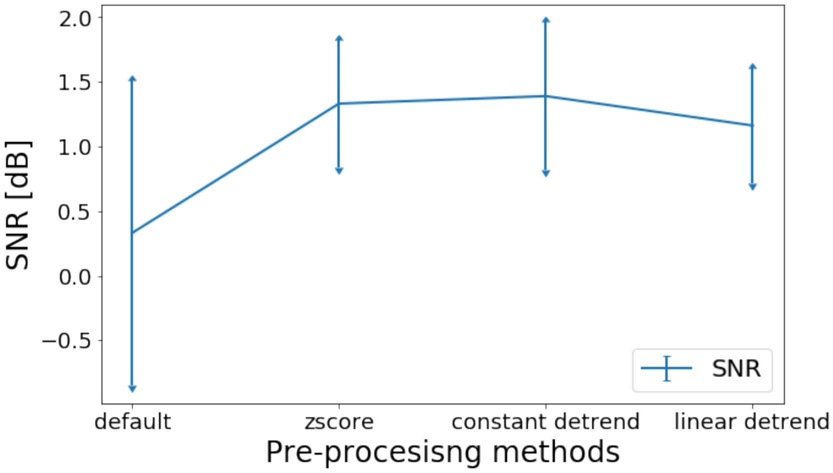


Figure 4.20: SNR of each pre-processing method on the training set.

any of the pre-processing transforms is higher than the default SNR. Among those, the constant detrending has the highest SNR, just 0.06dB above the second-highest, the Z score. However, the standard deviation of the Z score is 0.08db below the one of the constant detrending. The Z score will thus be the default transform used in the following chapter.

4.3.6 Time Complexity

The preprocessing techniques described in this section can be divided into two groups, namely the one and the two-dimensional preprocessing techniques. For both groups, the length of the frequency axis is constant, therefore, the complexity grows with the duration of the spectrogram. If N is the number of time sample in the spectrogram then the worst-case time complexity for the 1D preprocessing techniques is $O(N)$ and so it the time complexity for 2D techniques since the 2D kernel size is constant.

4.4 Summary

This chapter provided insight into the data at hand. The data were collected at two locations, Marion island and SANAE IV. Due to the poor quality of the data at SANAE IV, only the data collected at Marion IV was used. Out of the 2196 samples from Marion, 1471 (67%) samples were used for the training set and 375 (33%) samples for the testing set. The time-frequency representation (spectrogram) of the sample was selected as it provided a visual observation of the whistlers. Out of all the pre-processing method proposed, the one dimensional Z score transform has the highest SNR for the training set and was selected as the default pre-processing method for the spectrogram.

Chapter 5

Detection Using Cross-Correlation with a Simulated Whistler

This chapter focuses on the design of a method similar to the current state of the art in whistler waves detection developed by Lichtenberger *et. al* [7]. It first provides an overall understanding of the design, followed by a detailed explanation of all design decisions made for every input to the system. Lastly, the result of this design in a spectrogram is shown.

5.1 Design

From the literature, the more recent and well established method to detect whistlers is from Lichtenberger et al. [7].In his paper, Lichtenberger proposes a method which makes use of a simulated whistler using the Bernard approximation of whistlers [8] and a square-law detector. Our first approach in detecting whistler waves is to create a similar method to the one proposed by Lichtenberger et al.
The detector's design can be broken down in two pipelines as presented in Figure 5.1.

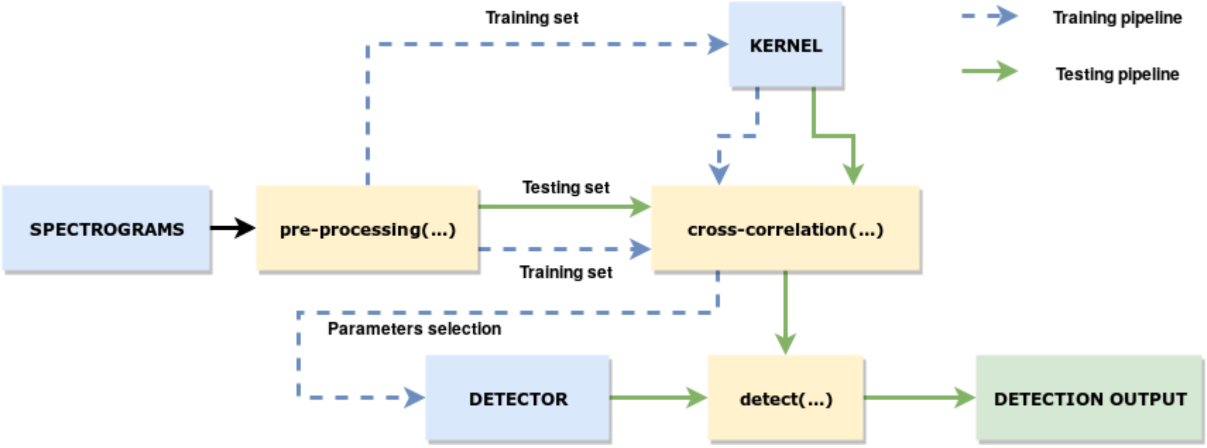


Figure 5.1: Overview of the whistler detection pipeline using cross correlation with a whistler kernel.

The method presented in Figure 5.1 has three processes and three inputs. The first process (*preprocessing(...)*) is described in Section 4.3. The second process (*corr-correlation(...)*) consists of finding the whistler's shape in the spectrogram of interest. The third process (*detect(...)*), consists of applying a suitable threshold to differentiate the whistlers from the noise present in the spectrogram. These processes are used in the training and testing pipeline of the design. During training, samples from the training set are preprocessed and analysed to select the kernel which later is used in conjunction with the same training samples to design the detector. During testing, the kernel is cross-correlated with pre-processed samples and the designed detector is applied to this result to generate the output of the detection.

5.2 Kernels

This section focuses on the identification of suitable kernels as inputs to the design. These kernels are two-dimensional representations of the whistlers. Two kernels are used, one extracted from the training data and another generated using the Bernard approximation of whistlers' dispersion [8].

5.2.1 Kernel Extracted From Data

Each sample in the dataset contains at least one whistler. The first approach is to use whistlers whose characteristics are intrinsic to the data at hand. Thus the labels provided by the AWD are used to extract whistlers from each sample of the training set. This extraction consists of generating whistler cuts from each spectrogram such that the entirety of the visible section of each whistler is present in its corresponding cut. We, therefore, choose these cuts to be 2.5 seconds long with the AWD output (the 5kHz time) at 30% of 2.5s and ranging from 0 to 12 kHz. Moreover, since only the whistlers are of interest, each sample is pre-processed using the Zscore before generating the cuts.
In total 8317 cuts are generated from the 1471 training samples. Figure 5.2 shows a few examples of the cuts with an indication of the time given by the AWD output.

The cuts presented in Figure 5.2 each contains at least one whistler, and in some cases, more. Using any one of these cuts will not provide a good representation of the whistler needed as a kernel. Therefore, the mean of the cuts is observed (Figure 5.3).

Figure 5.3 shows the intrinsic nature of the whistlers present in the training set, however, the boundaries initially chosen for the cuts contains portions of the spectrogram that are not of interest. We thus observed the magnitude of these cuts along the time and frequency axis as presented in Figure 5.4 to trim this cut.

The plots in Figure 5.4 present the whistler from Figure 5.3 alongside its normalised sum along its time and frequency axis. The plot of the normalised magnitude along the frequency axis shows the AWD output at 0.75s and the selected time boundaries at 0.55s and 1.55s. The normalised magnitude along the time axis shows the 5kHz reference line and the two selected

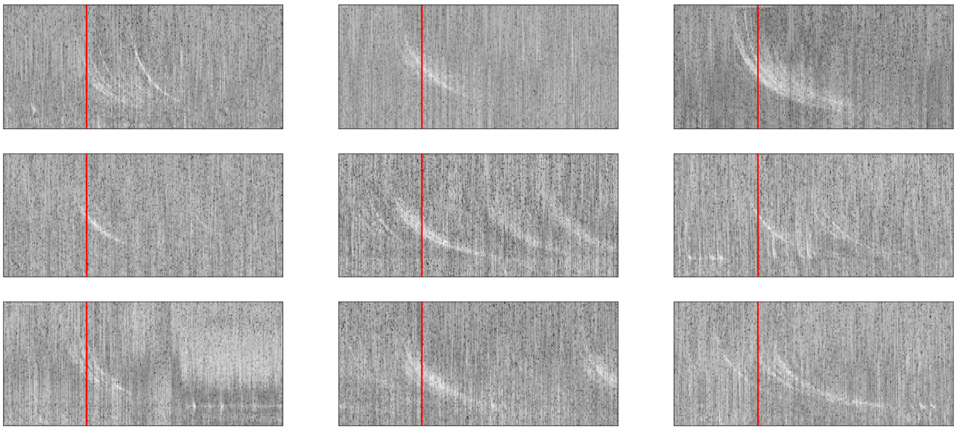


Figure 5.2: Whistler cuts generated from the training samples. The red line indicates the estimated location of the whistler using AWD.

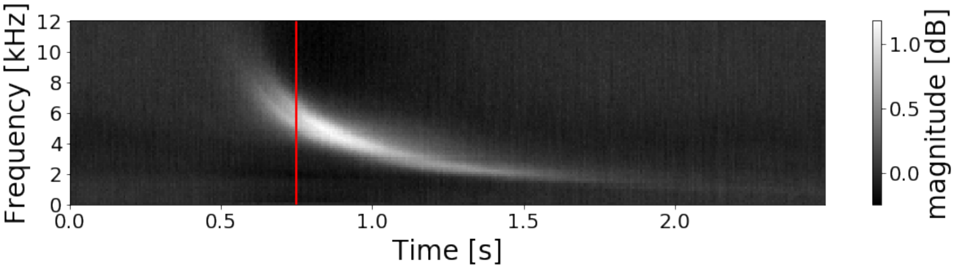


Figure 5.3: Mean of cuts generated from the sample in the training set.

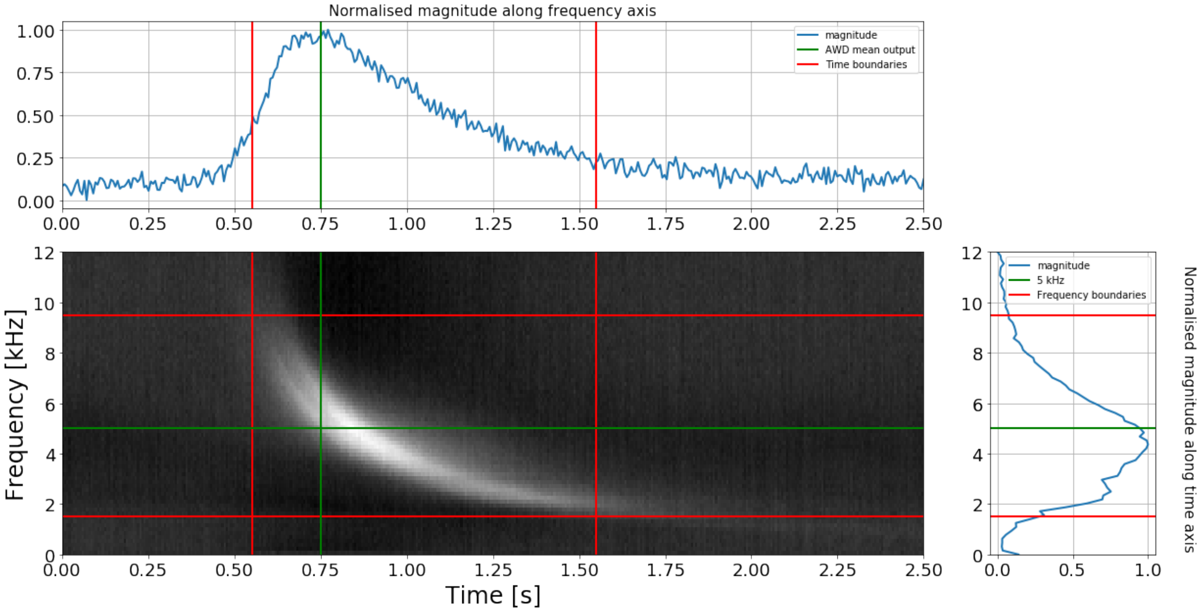


Figure 5.4: Mean of whistler cuts alongside the sum along the time and frequency axis. The boundaries chosen for the whistler are 0.55s, 1.55s, 1.5kHz and 9.5kHz

boundaries at 1.5kHz and 9.5kHz. This results in the final cut selection shown in Figure 5.5

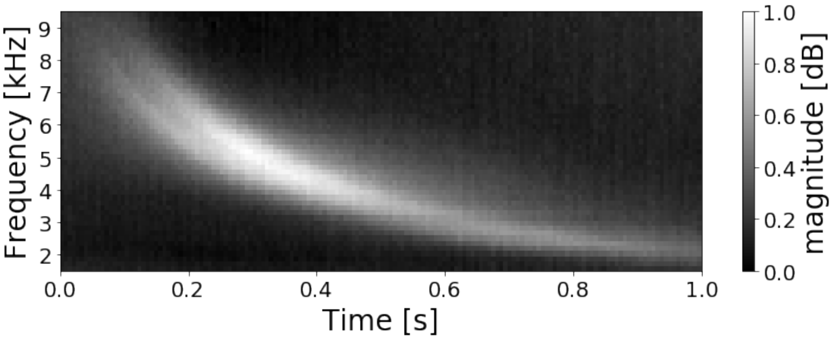


Figure 5.5: Kernel extracted from data. The final selected cut is 1s long and range from 1.5kHz to 9.5kHz.

5.2.2 Bernard approximation

The Bernard approximation of the Whistler dispersion provides two key parameters, the normalised nose frequency and the zero dispersion, both characterising the shape and duration of the whistlers. From Equation 2.5, t can be written as the linear function

$$t = f(f_n, f, \Lambda_n)D_0$$

with f_n, Λ_n, and f fixed parameters. Since $20 \leq D_0 \leq 80$ [7], an increase in D_0 results in an increase in the duration of the approximated whistler.

Since we have control over the shape and mostly, the duration of the whistlers, few kernels spanning over the range of D0 could be used. For example, Figure 5.3 shows a whistler lasting for more or less one second, this implies that the whistlers present in the data have more or less duration of one second. As a result, a simulated whistler with a duration of around one second should be optimal. To verify this, we correlate the whistler from Figure 5.3 with a series of simulated whistlers with D0 varying from 20 to 80 [7]. The results are presented in Figure 5.6.

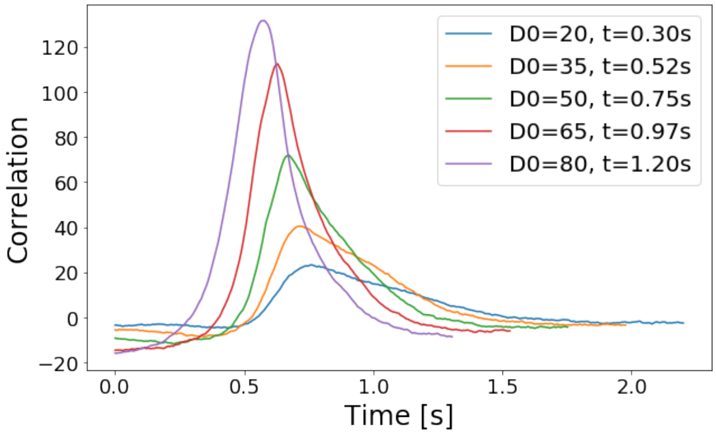


Figure 5.6: Result of correlation between whistlers generated using the Bernard approximation and the spectrogram in Figure 5.3.

All correlation have a peak in the 0.6s region which is expected since that is the start of the

whistler in Figure 5.3. As the duration of the whistler decreases, the area around the peak flattens, this is an indication the patterns present in the kernels are not representative of the whistler. In other words, whistlers of short duration are not dominant in the data. The highest result of the correlation is obtained for $D0 = 80$ which correspond to not only the longest generated whistler but one whose duration is closest to the whistler presents in Figure 5.3. The last parameter Λ_n is obtained from Figure 1 in [8] where Λ_n has a value of 0.35 for a diffusive equilibrium model.
Choosing the right simulated whistler for the correlation process will result in a sharp correlation curve in the area around the whistler of interest. We therefore propose to use a simulated whistler with the following parameter $f_n = 25kHz$, $\Lambda_n = 0.35$, $D_0 = 80$, and $1.5kHz \leq f \leq 9.5kHz$ Figure 5.7 as it has the closest similarities with the ones in the data.

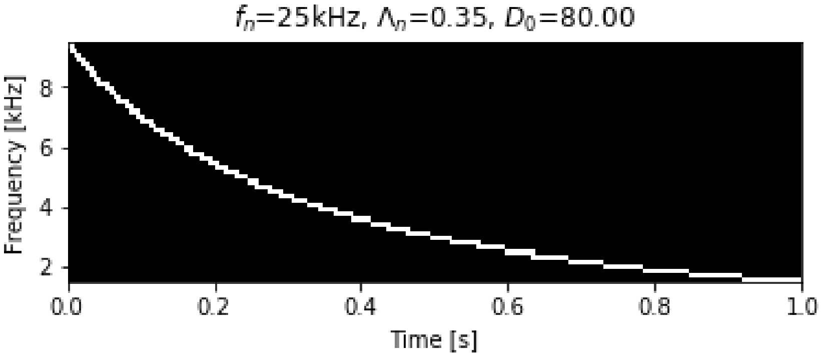


Figure 5.7: Kernel generated with $f_n = 25kHz$, $\Lambda_n = 0.35$, $D_0 = 80$, $1.5kHz \leq f \leq 9.5kHz$ with duration of respectively $t = 1.03$ seconds.

5.3 Spectrogram

As discussed in Section 4.3, the whistlers in the spectrogram can be emphasised by applying different preprocessing techniques. As a result, each spectrogram undergoes preprocessing before being correlated with the kernel of choice. Moreover, the correlation process aims to generate a one dimensional output, therefore, each spectrogram must have the same frequency range as each kernel. We thus crop each spectrogram from 1.5kHz to 9.5kHz as shown in the preprocessing pipeline illustrated in Figure 5.8. After cropping, the spectrogram can furthermore be processed using either a one or two-dimensional pre-processing methods. The last stage of preprocessing consists of scaling the spectrogram to withing the range of 0 and 1. Figure 5.9 shows the results

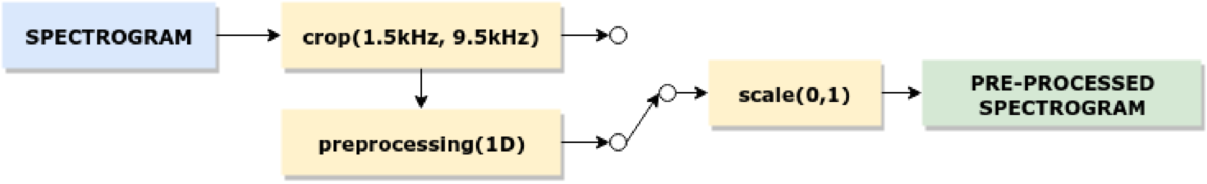
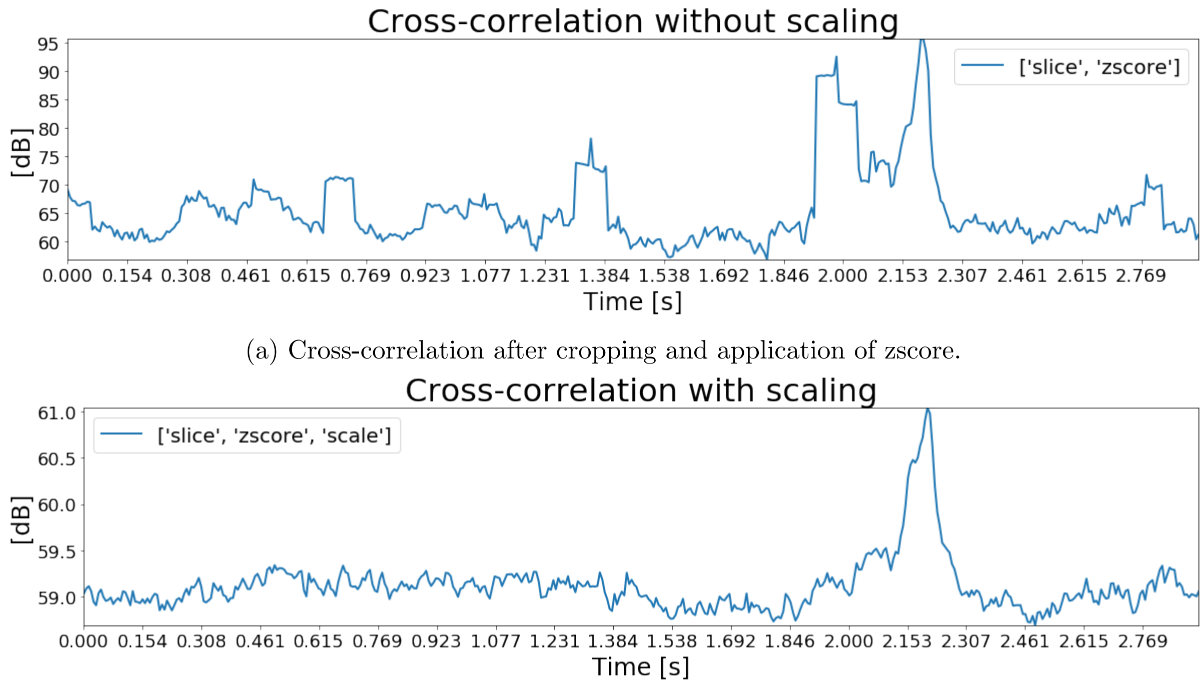


Figure 5.8: Pre-processing pipeling. Each spectrogram is firstly cropped, can be preprocessed using one of the 1D techniques proposed in Section 4.3, and lastly is scaled between 0 and 1.

of cross-correlating a simulated whistler with a spectrogram processed with and without a scale.



(a) Cross-correlation after cropping and application of zscore.

(b) Cross-correlation after cropping and application of zscore, and finally a scale from 0 to 1.

Figure 5.9: Results of Cross-correlation with pre-processed spectrogram with a whistler at 2.2s.

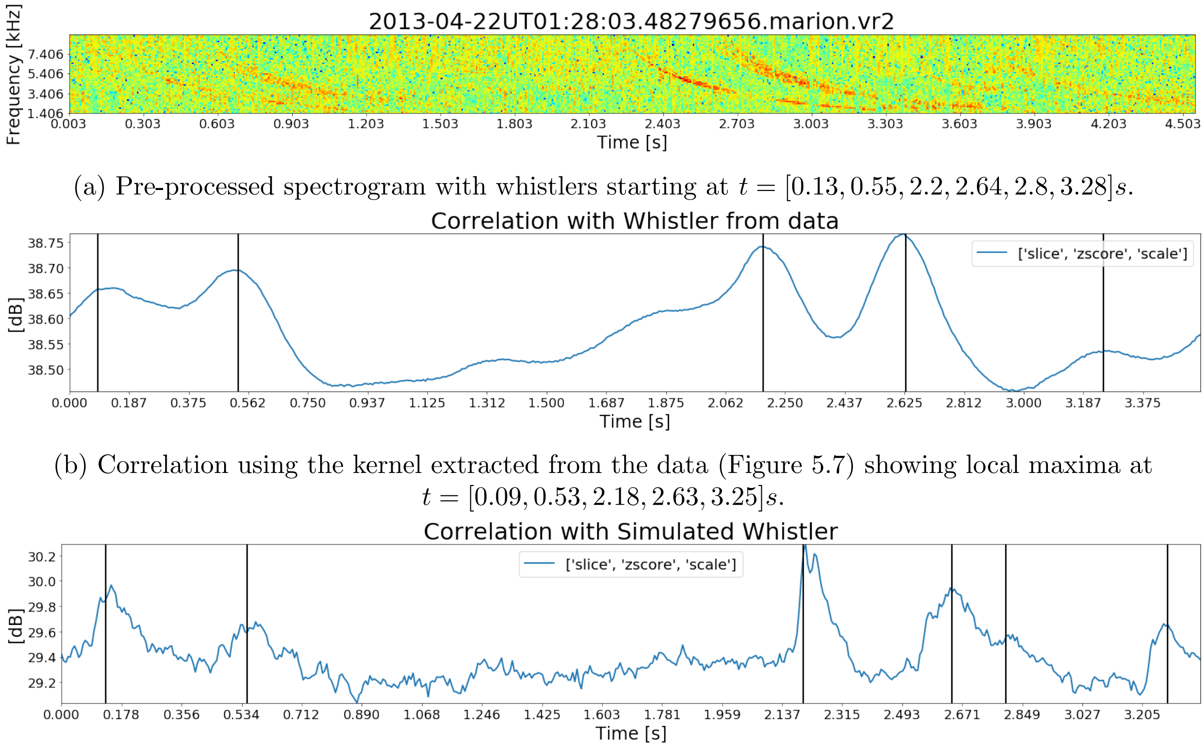
The noise level from the unscaled spectrogram (Figure 5.9a) has very heterogeneous noise levels while the noise level with the scaled spectrogram are homogeneous (Figure 5.9b). Scaling the pre-processed spectrogram before cross-correlation can thus reduce the presence of the noise in the result of the cross-correlation.

5.4 Cross-Correlation

The second process of the pipeline in Figure 5.1 is the cross-correlation between the preprocessed spectrogram and the kernel of choice. This process measures the similarities between the spectrogram and the kernel over time. It is easier to work with a 1D matrix, thus the kernel and the preprocessed spectrogram are chosen as input to Equation 2.10.

Figures 5.10b and 5.10c display the result of this process on the preprocessed spectrogram in Figures 5.10a with the kernels presented in Figure 5.5 and Figure 5.7. The result of the cross-correlation with the kernel extracted from the data shown in Figure 5.10b has four local maxima at $t = [0.09, 0.53, 2.18, 2.63]s$ and only captures the first 3.56 seconds of the spectrogram[1]. Figure 5.10c has five local maxima at $t = [0.13, 0.55, 2.2, 2.64, 2.8]s$ and only captures the first 3.38 seconds of the original spectrogram. Both results have local maxima at the location of the first four whistlers present in Figure 5.10a. The correlation in Figure 5.10b however, does not

[1] The cross-correlation being used in valid mode, the result is trimmed according to Equation 2.10.



(a) Pre-processed spectrogram with whistlers starting at $t = [0.13, 0.55, 2.2, 2.64, 2.8, 3.28]s$.

(b) Correlation using the kernel extracted from the data (Figure 5.7) showing local maxima at $t = [0.09, 0.53, 2.18, 2.63, 3.25]s$.

(c) Correlation using the simulated kernel (Figure 5.7) showing local maxima at $t = [0.13, 0.55, 2.2, 2.64, 2.8, 3.28]s$.

Figure 5.10: Results of correlation using the kernel extracted from the training data and the simulated kernel

have a local maximum at the location of the fifth whistler. This is due to the thickness [2] of the kernel extracted from the data. The local maxima in Figure 5.10b are thicker which is not ideal since consecutive whistlers might result thicker regions which could be detected as noise.

To maximise the detection of whistlers (local maxima after cross-correlation), the simulated whistler in Figure 5.7 will be used as the default kernel.

5.5 Detector

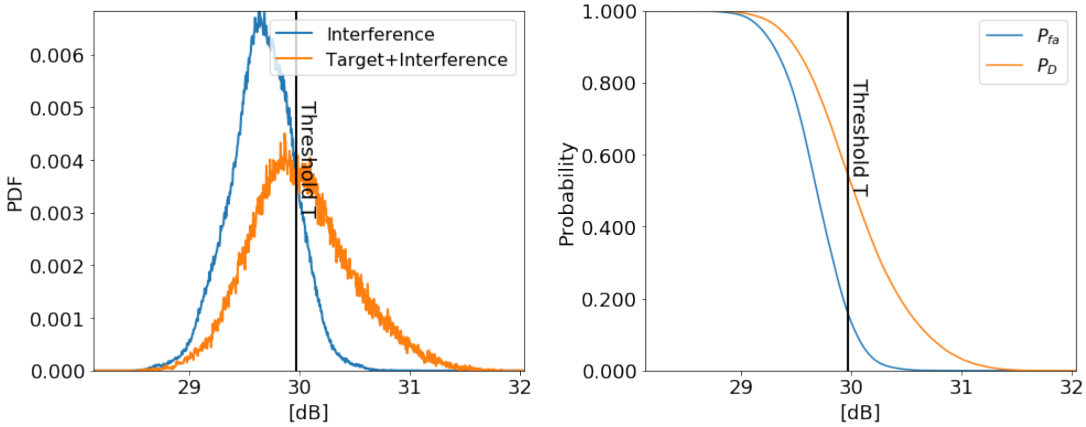
5.5.1 A need for an adaptive threshold

The third input to the pipeline in Figure 5.1 is the detector. The role of the detector is to generate a suitable threshold to discriminate targets from interference. To understand the behaviour of the targets and interference in the system, the interference (I) and target plus interference ($T + I$) from all cross-correlations resulted using the training set are extracted. The targets in any cross-correlation are obtained by using the labels provided by the AWD method. In Figure 5.3, the starting point of the whistlers was chosen to be 0.2s[3] before the AWD label (time at 5kHz of the whistler), therefore, this starting point of the AWD label is chosen as section of the cross-correlation containing targets plus interference. The interference

[2] Here, thickness refers to the spread of the magnitude of a whilster at a frequency, typically 5kHz.

[3] 5KHz point of the whistler where the magnitude is the highest.

is obtained by taking the section of the cross-correlation which does not contain the portion of targets plus interference.
Figure 5.11 presents the probability distribution of both interference and target plus interference as well as the probability of false alarm (P_{fa}) and probability of detection (P_D) per threshold.[4]



(a) Probability Density Function (PDF) of the interference and target plus interference.

(b) Probability of False Alarm P_{fa} and Probability of Detection P_D per threshold.

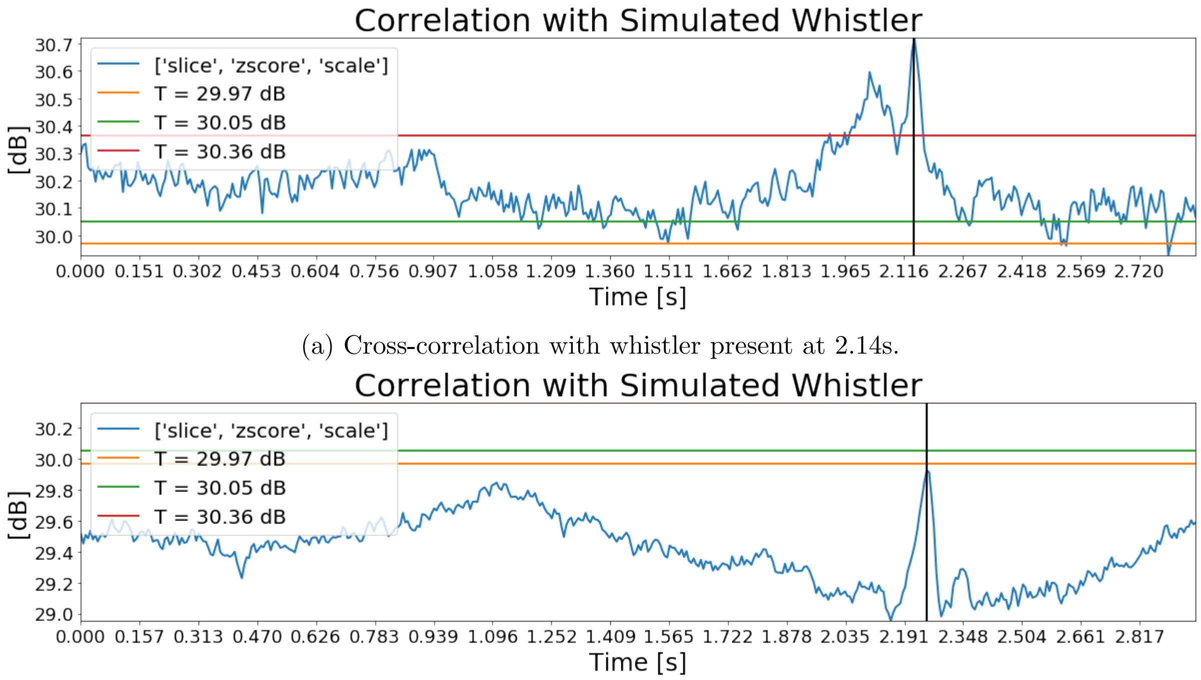
Figure 5.11: Statistics of interference and target plus interference of the cross-correlation on the training set with threshold $T = 29.97dB$.

Considering all training samples, the interference has a mean of $\mu_I = 29.68dB$ and a standard deviation of $\sigma_I = 0.30dB$ and the targets plus interference has a mean of $\mu_{T+I} = 30.06dB$ and a standard deviation of $\sigma_{T+I} = 0.47dB$. The $T + I$ and I PDF curves are both Rayleigh distributed [19] with the TI PDF curve flatter and skewer to the right. Selecting a static threshold for target detection can thus be easily done by selecting a desired P_d or P_{fa}. A simple approach is to select T where $P_D - P_{fa}$ is the highest. In this case, it corresponds to $T = 29.97dB$ where $P_D = 5.44\text{E}{-1}$and $P_{fa} = 1.58\text{E}{-1}$ as shown in Figure 5.11.

Figure 5.12 shows a few selected thresholds alongside the result of the cross-correlation for two training samples.

In a system with homogeneous interference, a static threshold can be used to detect targets from interference. However, the interference after pre-processing of the spectrograms and cross-correlation are unpredictable. For example, the interference presents in Figure 5.12b changes magnitude at different section of the cross-correlation. Moreover, the whistler's maximum cross-correlation value is below the threshold level guarantying the maximum difference between P_D and P_{fa}. In Figure 5.12a, however, the interference, as well as the target plus interference, are well above this threshold. This problem cannot be efficiently tackled if using a static threshold detector. We thus propose the use of adaptive threshold-based detectors.

[4] P_{fa} is the area under the interference PDF curve to the right of the threshold while P_D is the area under the target+interference PDF curve to the right of the threshold



(a) Cross-correlation with whistler present at 2.14s.

(b) Cross-correlation with whistler present at 2.25s.

Figure 5.12: Cross-correlation result with static thresholds selected such that $(T, P_{fa}) \in \{(29.9, 71.58\text{E}{-}1), (30.05, 1\text{E}{-}1), (30.36, 1\text{E}{-}2)\}$.

5.5.2 CFAR Detectors

Constant False Alarm Rate detectors require at least three parameters, the window size which includes the cell under test (CUT), the noise cells and guard cells, and a scaling factor given as a function of the probability of false alarm.

CFAR Window

CFAR detectors use a window and a statistical scaling factor to generate an adaptive threshold. One challenge with CFAR is the selection of the CFAR window size. The window size W is dependent of the number of guard cells G and the number of noise cells N on each side of the cell under test, precisely, $W = 2(N + G) + 1$. Ideally, the width of the local maxima should be less than twice the number of guard cell ($whistler \leq 2G$) while the next consecutive whistler should be located at $\pm(N + G)$ cells away from the CUT. However, whistlers do not have a constant or minimum duration from one to the next. Moreover, the presence of whistlers might be faint in the spectrograms resulting in some cases where the interference levels are above the whistlers.

Figure 5.13 shows the window selected by the CFAR detector. With each cell being 6.4ms long, the window selected in Figure 5.13 has 7 guard cells (44.77ms) around each CUT followed by 10 noise cells (63.95ms) accounting for a total of 35 cells (223.83ms) in each window. This window was chosen on the assumption that whistlers will be separated 0.2s (time between the start of the whistler and its peak).

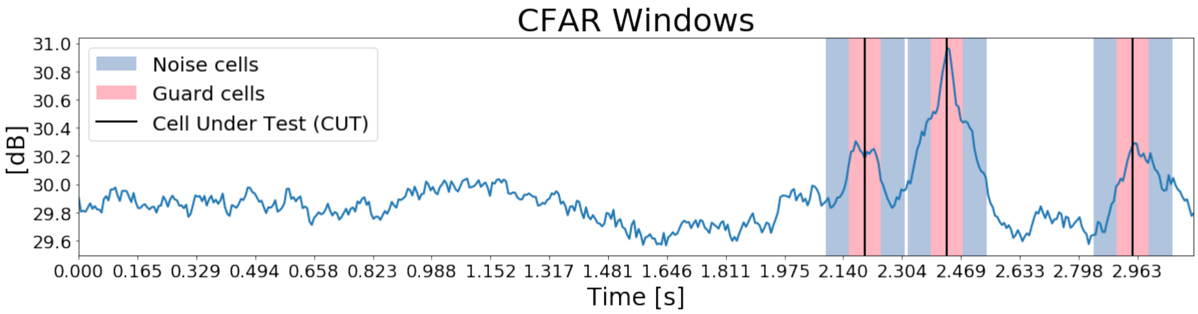


Figure 5.13: CFAR windows.

CFAR Probability of False Alarm

The most basic type of CFAR is the Cell Averaging CFAR where the adaptive threshold at the CUT is given by

$$T_{CUT} = T\hat{\sigma}^2_{CUT} \tag{5.1}$$

with T, the threshold defined by Equation 2.11 and $\hat{\sigma}^2_{CUT}$ the average noise power in the corresponding noise cells. In decibels, this is equivalent to

$$[T_{CUT}]_{dB} = [T]_{dB} + [\hat{\sigma}^2_{CUT}]_{dB}$$

$$[T_{CUT}] - Noise_{power} = [T]_{dB}$$

Let X_{dB} be the difference between the threshold and the noise power, then

$$X_{dB} = 10log_{10}(2N(P_{fa}^{-1/2N} - 1))$$

with N the number of noise cell on each side of the CUT. From this, P_{fa} can be expressed in term N and X_{dB} as

$$P_{fa} = \left(\frac{1}{1 + \frac{10^{X_{dB}/10}}{2N}} \right)^{2N} \tag{5.2}$$

We previously mentioned how P_{fa} can be obtained by selecting a threshold using Figure 5.11b, however, since a single whistler in the cross-correlation can cover many target cells, this method is thus difficult to visually understand the adaptive threshold for CFAR with our type of data. From Equation 5.2, P_{fa} can be obtained if the number of noise cells N is known as well as the desired decibel level X_{dB} between the power of the CUT and its corresponding noise power. Note that this approach uses the interference power instead of the interference magnitude.

The difference in decibel between the power in the target plus interference and the interference in the training set is $X_dB = 0.75dB$. The target+interference (whistler) is 0.2s long, however, only the peak of these cells are of interest. Also, since an adaptive threshold detector is used, X_dB can be decreased in an attempt to increase the probability of detection while maintaining

a low probability of false alarm. Therefore, a value of $X_dB = 0.5dB$ [5] is chosen as for the CFAR threshold. This corresponds to a threshold of $T = 29.81dB$ and is associated with a probability of false alarm $P_{fa} = 3.36\text{E}{-1}$ and a probability of detection of $P_D = 6.82\text{E}{-1}$ as shown in Figure 5.11.
Figure 5.14 shows the cross-correlation result of the spectrogram in Figure 5.10a with lines below each maximum CUT, marking the desired decibel level between the CUT and the noise power for $X_{dB} = 0.5dB$.

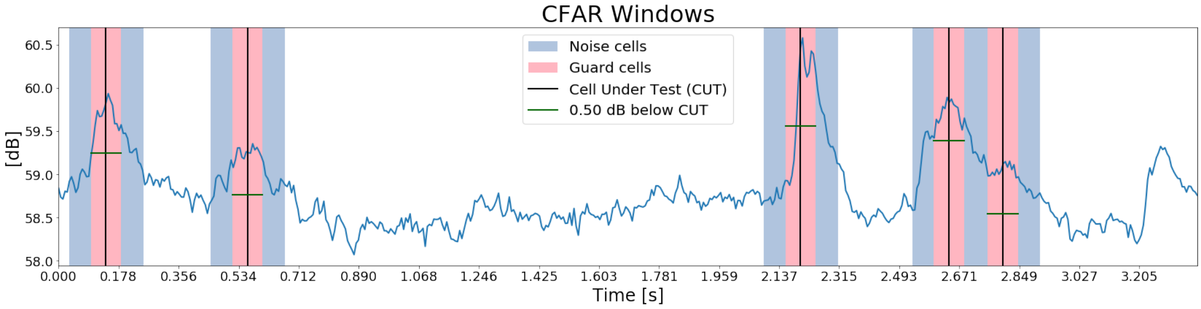


Figure 5.14: Cross-correlation with CFAR window and the 0.5dB line below the maxima for the spectrogram in Figure 5.10a

For most of the peaks in Figure 5.14, the noise power levels are below the $-0.5dB$ line, except for the noise cells at $t = 0.54s$, $t = 0.14s$ and $t = 2.82s$, consequently, the CUT related to these noise cells might not be detected by this CA CFAR detector.

Cell Averaging CFAR (CA CFAR)

To emphasise on the difference between the CFAR detectors described below, the pre-processed spectrogram in Figure 5.15 will be used as a reference.

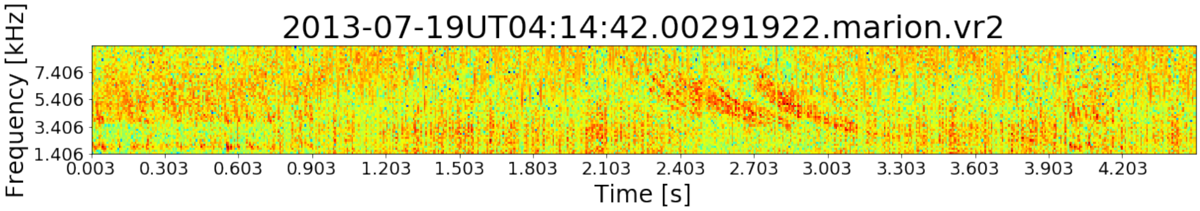
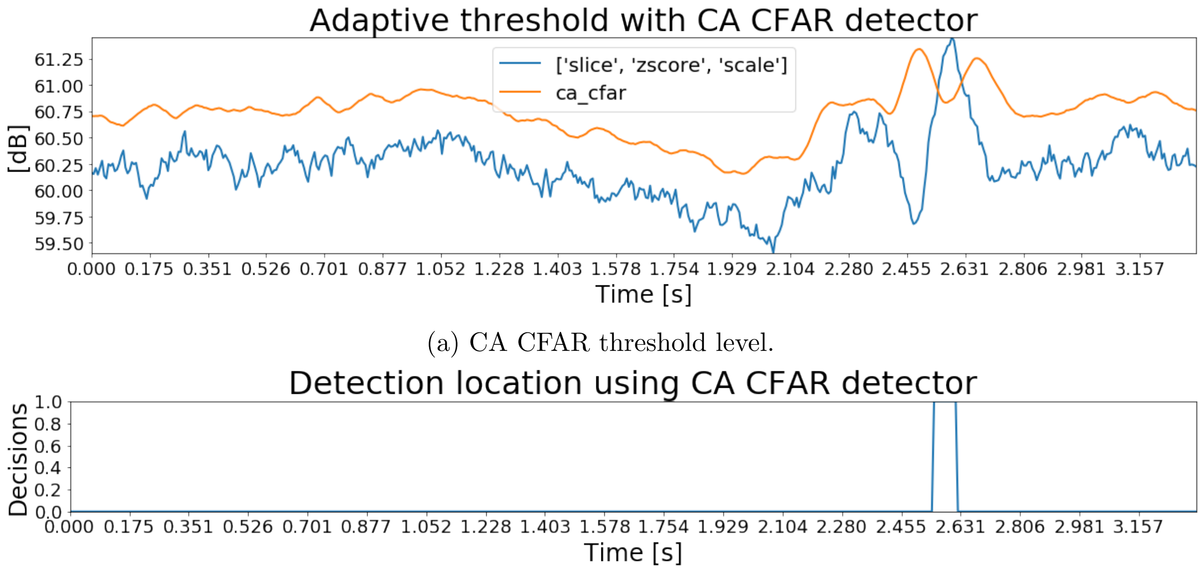


Figure 5.15: Pre-processed spectrogram used as example for CFAR detector with three whistlers at $t = [2.38, 2.37, 2.59]$.

The first CFAR detector is the Cell Averaging CFAR (CA CFAR) which is applied using the parameters chosen above. The result of this CA CFAR detector with $N = 10$, $G = 7$, and $P_{fa} = 3.36\text{E}{-3}$ is shown in Figure 5.16.[6]

[5] This was chosen experimentally and can be used as a parameter when tuning the detector

[6] All cross-correlation plot are the square of the cross-correlation. This is simply because the detector use the power in the signal for detection decisions.



(a) CA CFAR threshold level.

(b) CA CFAR detection decisions.

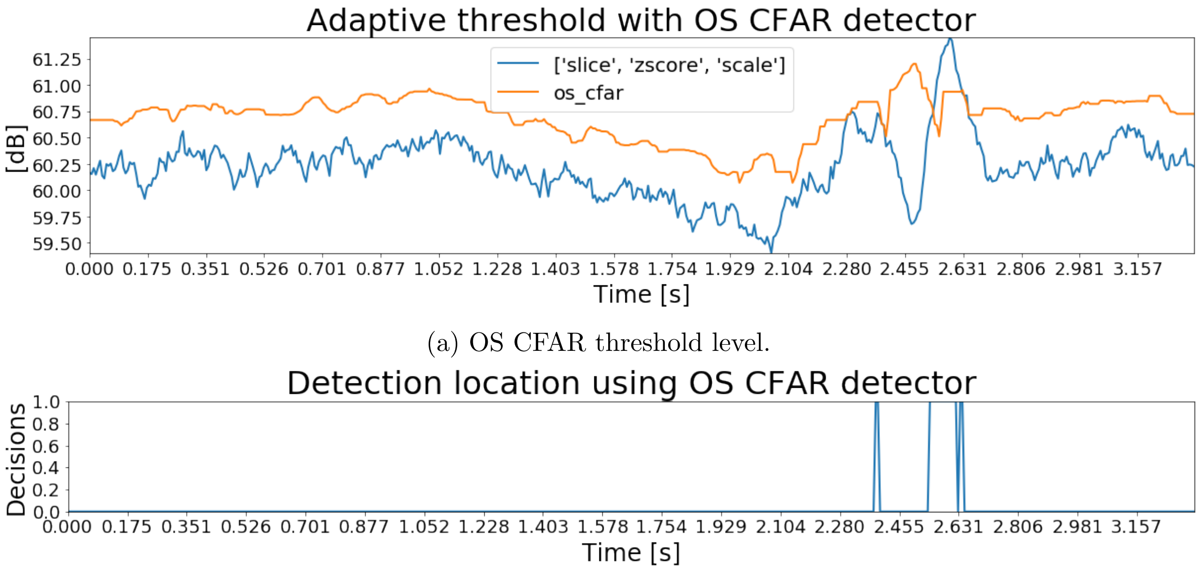
Figure 5.16: Results of CA CFAR with $N = 10$, $G = 7$, and $P_{fa} = 3.36\text{E}{-3}$. One whistler is detected at $t = 2.59s$.

The CA CFAR detector was able to detect only one out of the three whistlers present in the spectrogram. The first two whistlers are not detected as they interfere with one another. The detection of these whistlers might be achieved by reducing the window size such that there are no interfering targets, however, this might also increase the probability of false alarm since the noise is not homogeneous. We thus propose more robust CFAR techniques for detecting such targets.

Order Statistic CFAR

Order Statistic CFAR (OS CFAR) selects the noise power in the k^{th} ordered noise cell as average noise power to be considered by the CFAR detector.
Let $\mathcal{N} = \{n_1^2, n_2^2, \ldots, n_N^2, \ldots, n_{2N}^2\}$ be the set of ordered noise power of the noise cells, x_{CUT}^2 the power of the CUT, and $\hat{\mathcal{N}}$ the average noise power. Assuming that the noise around the targets have a Gaussian distribution with $x_{target}^2 \geq \hat{\mathcal{N}} + X_{dB}$, since $\hat{\mathcal{N}}$ is ordered, n_k^2 for $k = N$ is likely to be equal to $\hat{\mathcal{N}}$. If either side of the CUT has at least one interfering target [7], it is safer to assume that the interfering target[s] covers the half side of the noise cells. Consequently, $k \leq N$ is a preferred option. If the CUT is surrounded by interfering targets on both sides, changing the value of k will not help, instead, a different window size should be chosen to ensure that such case do not occur.
For the OS CFAR, the number of noise cells and guard cells, as well as the probability of false alarm, are unchanged. k is chosen such that $k \leq N$, precisely, $k = N - 1 = 9$. The result of the OS CFAR detector with $N = 10$, $G = 7$, $k = 9$, and $P_{fa} = 3.36\text{E}{-3}$ is shown in Figure 5.17.

[7] Interfering target, in this case, refers a cell belonging to another whistler.



(a) OS CFAR threshold level.

(b) OS CFAR detection decisions.

Figure 5.17: Results of OS CFAR with $N = 10$, $G = 7$, $k = 9$, and $P_{fa} = 3.36\text{E}{-3}$. Three whistlers are detected at $t = [2.386, 2.590, 2.641]s$.

The OS CFAR detector detected two out of the three whistlers present in the spectrogram. Moreover, it found two instead of a single whistler around time $t = 2.6s$. This is the result of the discontinuity of the whistlers in the spectrogram.

Trimmed Mean CFAR

Trimmed Mean CFAR (TM CFAR) can be though of as a generalisation of OS CFAR. Instead of a single noise cell used to calculate the noise power, a section of the ordered noise power in each noise cell is used. TM CFAR requires two additional parameters T_S and T_L on top of N, G and P_{fa}. T_S and T_L are used to select the subset $\hat{\mathcal{N}}_{(L,S)}$ of $\hat{\mathcal{N}}$ such that $\hat{\mathcal{N}}_{(L,S)} = \{n^2_{T_S}, \ldots, n^2_{T_L}\}$ with $0 \leq T_S < T_L \leq 2N$. T_L is chosen assuming that the CUT is surrounded by interfering targets, while T_S is chosen to reduce clutter edge false alarm.

For the TM CFAR, we will assume that we want to remove a maximum of 80% of the interfering target cells and 30% [8] of the clutter cells leading to an clutter edge. The result of the TM CFAR with $N = 10$, $G = 7$, $T_S = 3$, $T_L = 8$ and $P_{fa} = 3.36\text{E}{-3}$ is shown in Figure 5.18.

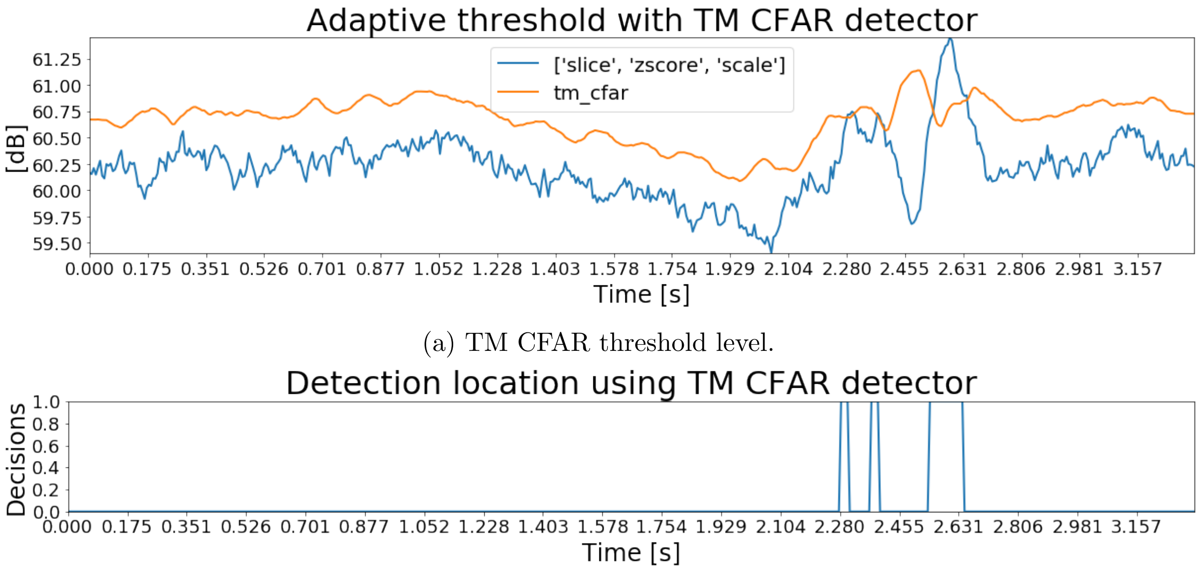
The TM CFAR detector detected all three whistlers present in the spectrogram.

Each of these CFAR work more or less well individually, however, a more robust CFAR, combining these CFAR can be used.

Linear Fusion CFAR

The Linear Fusion CFAR (LF CFAR) shown in Figure 2.9 is used to combine the decision of the CA CFAR, OS CFAR and TM CFAR. Figure 5.19d shows the output of the LF CFAR with

[8] Value chosen experimentally and can be used as parameter for improving the detector



(a) TM CFAR threshold level.

(b) TM CFAR detection decisions.

Figure 5.18: Results of TM CFAR with $N = 10$, $G = 7$, $T_S = 3$, $T_L = 8$, and $P_{fa} = 3.36\text{E}{-3}$. Three whistlers are detected at $t = [2.296, 2.373, 2.590]s$.

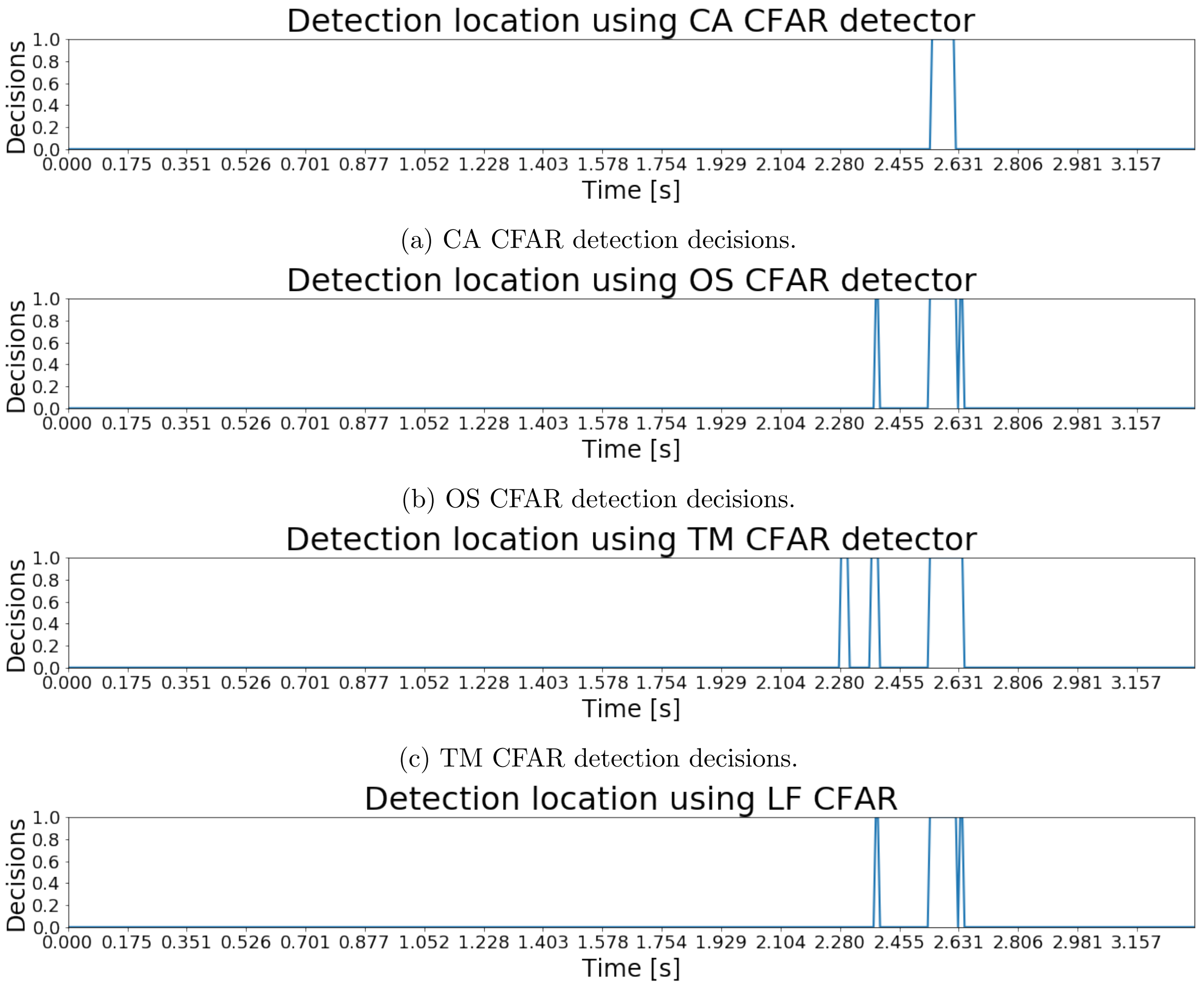
$N = 10$, $G = 7$, $k = 9$, $T_S = 3$, $T_L = 8$, and $P_{fa} = 3.36\text{E}{-3}$.

The Linear Fusion CFAR detector found three whistlers with their starting times at $t = [2.386, 2.590, 2.641]s$. The first detected whistler is an example of the LF CFAR rules where a target is detected if detected by both the OS CFAR and TM CFAR despite misdetection by CA CFAR due to either interfering targets or change in clutter features.

5.6 Detector Output

The output of the detector is the starting time and its associated cross-correlation value for each detected whistler in the spectrogram. The maximum number of whistlers that can be detected in an interval of time is given by the time resolution of the spectrogram; for Marion, the time resolution is $6.4ms$. This leads to results that are sometimes unreadable, thus a time resolution of $0.1s$ is selected. Therefore, whistlers detected within $0.1s$ interval are considered as one single detection with the cross-correlation value being the highest of them all. To satisfy requirement R_{12} from Table 3.1, the ending, therefore, the duration of the whistlers should also be found. This requirement can easily be met by using the duration (1s) of the simulated whistler use as kernel as the duration of each detected whistler. However, a more suitable approach can be used by taking the duration of the simulated whistler associated with the maximum cross-correlation between a 1s cut of the detected whistler and simulated whistlers of all possible duration.

The final result of the detection using the method described in this chapter for the sample shown in Figure 5.15 is shown in Figure 5.20.



(a) CA CFAR detection decisions.

(b) OS CFAR detection decisions.

(c) TM CFAR detection decisions.

(d) LF CFAR detection decisions.

Figure 5.19: Results of LF CFAR with $N = 10$, $G = 7$, $k = 9$, $T_S = 3$, $T_L = 10$, and $P_{fa} = 3.36\text{E}{-3}$. Three whistlers are detected at $t = [2.386, 2.590, 2.641]s$.

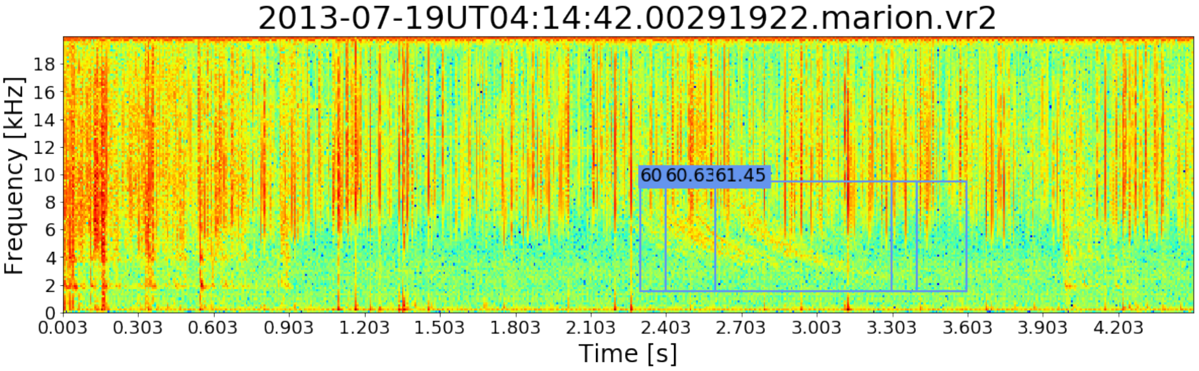


Figure 5.20: Result of detection using Cross-correlation with a simulated whistler. Three whistlers are detected starting at t=2.3s, t=2.4s, and t=2.6, all with a duration of 1s.

5.7 Time Complexity

The time complexity of detecting and locating whistlers using this method can be found by analysing the three processes in the testing pipeline. The first process (preprocessing) has a cropping stage ($O(1)$), a 1D preprocessing stage ($2O(N)$) and a scaling stage ($O(1)$). The worst-case time complexity for this process is thus $O(N)$. The second process in the pipeline is the cross-correlation which has a time complexity of $O(N)$. The detector process also has a complexity that varies with the duration of the spectrogram so its time complexity is also $O(N)$. Therefore, the time complexity of the design is $O(N)$ with N the number of sample in a spectrogram.

5.8 Summary

This chapter provided an in-depth elaboration of a replica of the current state of the art in whistler wave detection. The method proposed uses a cross-correlation between a simulated whistler and a portion of a spectrogram. The simulated kernel's parameters were selected such that the simulated whistler matches best the general shape of the whistler from the training data. As opposed to the SOTA, the spectrograms were pre-processed using a one-dimensional transform. This lead to the reduction of spherics and transient signal and the emphasis of the whistlers present in each spectrogram. Furthermore, to reduce the heterogeneity of the interference level in the spectrograms, each spectrogram were scaled to a value between 0 and 1. The selected simulated whistler and pre-processed spectrogram were cross-correlated to generate a one-dimensional curve measuring the presence of whistler in the spectrogram. A fusion detector combining three adaptive threshold-based detectors (CA CFAR, OS CFAR, TM CFAR) was used to detect the starting time of the whistlers present in the curve generated by the cross-correlation. Lastly, the results were adjusted as to be readable and the duration of each detected whistler was found using the result of the cross-correlation between a 1 second cut of the detected whistler and a set of simulated whistler with different duration 5.20.

Chapter 6

Object Detection Using Machine Learning Techniques

In this chapter, two machine learning-based designs for whistler detection are presented. The first design uses an approach similar to the whistler wave detector presented in Chapter 5 by using a neural network classifier instead of a simulated whistler. The second approach uses a pre-trained model from the current state of the art in object detection known as You Only Look Once (YOLO).

6.1 Detection using image classification

In the domain of computer vision, image classification is the action of identifying objects, people, actions, etc, from an image. In this section, the design of a whistler detector that uses a whistler classifier to detect and localise whistlers in a spectrogram is presented.

6.1.1 Design

The overview of the design of the whistler wave detector using a neural network classifier is presented in Figure 6.1.

The whistler wave detector has two pipelines, a training, and a testing pipeline. During training, a neural network classifier is trained using preprocessed samples from the training set and their AWD labels. During testing/evaluation, the trained classifier is slid over a preprocessed spectrogram resulting in a probability of detection over time curve. This curve is then fed to a detector which detects and locates the whistlers.

6.1.2 Neural Network Classifier

The first part of designing this detector is training the classifier. This neural network classifier requires sufficient labelled samples for training and a choice of network architecture. The design

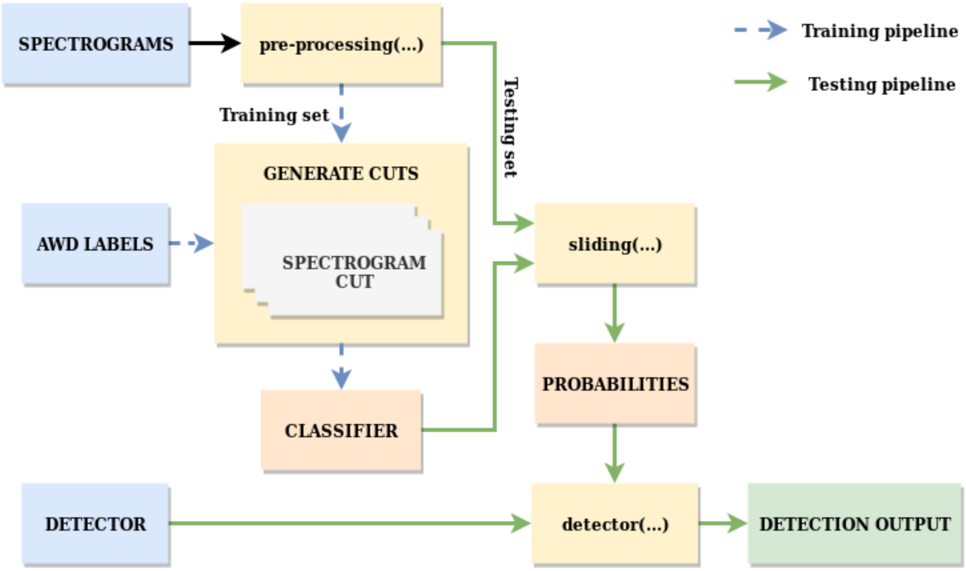


Figure 6.1: Overview of the whistler detector design using a neural network classifier.

decisions concerning these two requirements are thoroughly elaborated in this section.

Dataset

From the Marion dataset, 1471 samples were selected to be from the training set. Using the AWD method, we generate whistler and noise cuts of 1s between 1.5kHz, and 9.5kHz as in Section 5.2. The noise cuts are the portion of the spectrogram which do not contains whistlers while the whistler cut are the portions containing whistlers. From this training set, a total of 11148 cuts are generated with 8317 (74.61%) whistler cuts and 2831 (25.39%) noise cuts. Figure 6.2 shows nine of the cuts generated using the training set.

The dataset generated is unbalanced with three times more whistler cuts than noise cuts. If we were interested in detecting noise, this would be a problem, however, this unbalance work

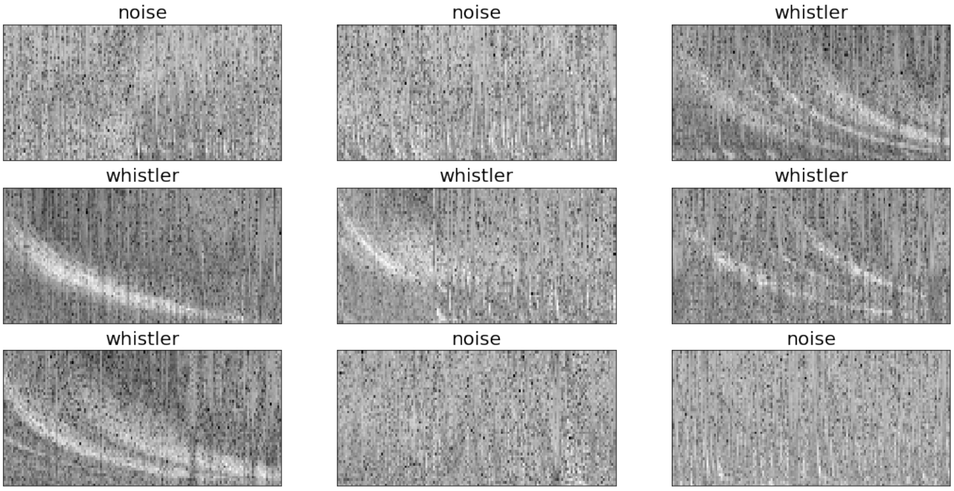


Figure 6.2: Cuts generated from the training set and preprocessed with a zscore and scaled between 0 and 1.

in our favour since we are only interested in detecting the whistler and not what is not.

Neural Network Architecture

Numerous neural network architectures have been developed since the popularisation of artificial intelligence [3]. Among those, Deep Convolutional Neural Networks are known to be adequate for image classification mostly because of their convolutional layers. These networks have two main components, a convolutional base and a classifier (Figure 6.3). The convolutional base typically features convolution cells or pooling layers and few kernels. This base learns hierarchical feature representations of the data. According to Yosinski[30], there is a general to specific learning process occurring in the network. In such cases, adding more layers might result in more complex features being learnt at the deeper layers. As for the classifier, it is generally a set of fully connected layers that classify an image according to its features.

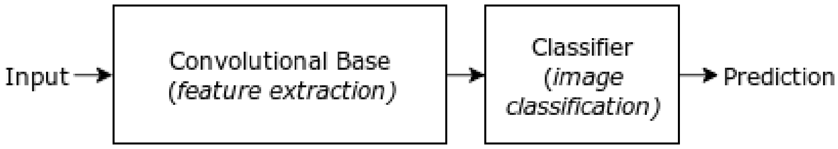


Figure 6.3: Topology of a Deep Convolutional Neural Network [3].

The object of interest for this project are whistlers. As explained in the literature, whistlers have a simple shape, consequently, a convolutional neural network with at least three convolutional layers should be sufficient for whistlers wave classification.

The architecture for the classifier is presented in Figure 6.4. This network takes each cut as input images with shape $(1 \times 52 \times 156)$ and an output with shape (2×1). The convolutional base is composed of two pairs of convolutional layer - max-pooling layer followed by a convolutional. The output of the convolutional base is flattened and fed to the classifier with has one dense layer of 128 neurons and a final dense layer with only two neurons as the output of the classifier using a *softmax* activation. This architecture is common for detection of simple shape such as whistlers. The idea behind the number of convolution layer is to get more refined details of the filters.

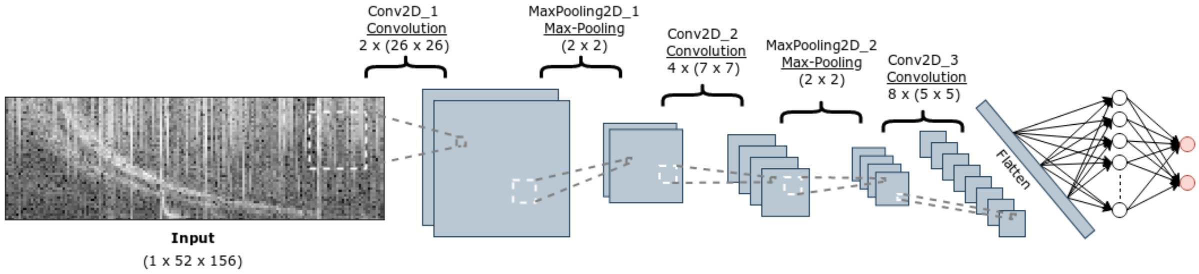


Figure 6.4: Architecture of the classifier of the whistler wave detector

Cross-validation

When training a neural network, the aim is to minimise the error rate ϵ defined as the portion of errors made during training. This error is defined as the number of time $h(x) \neq y$ for each (x, y) pairs, where y is the label of sample x[31]. If the hypothesis h has a lower rate on the training set, it by no means implies that this will hold for the test set. One approach to solve this is to split the training set into a training and a validation set to evaluate ϵ, this is known as cross-validation. One drawback with this is that the training data is now smaller resulting in the data not being fully used. We tackle this by using a technique known as k-fold cross-validation. In k-fold cross-validation, the entire training set is split into k-folds. For each of the k iterations on the training set, the model is evaluated on 1 of the k^{th} folds and trained on the $k - 1$ other folds. With these techniques, the average error of ϵ_{mean} is a better estimate of the performance of the model. Another version of the k-fold cross-validation is the stratified shuffle split (Figure 6.5). This technique account for unbalance in the dataset by preserving the same percentage for each class and maintaining the ratio of training data used for validation [4].
Our model will be evaluated using five iterations of the stratified shuffle split with a validation percentage of 33%.

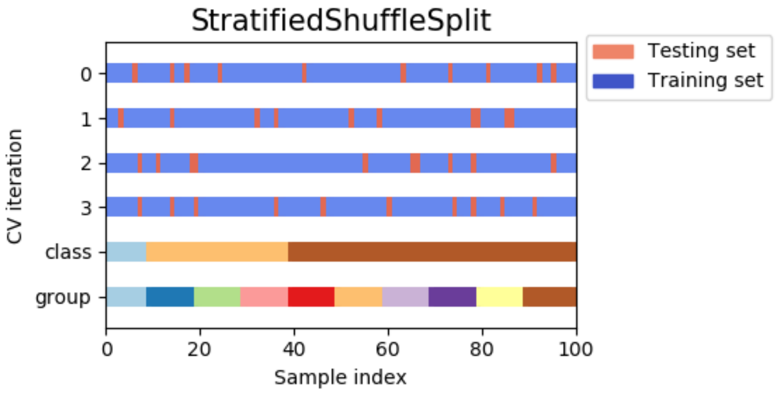
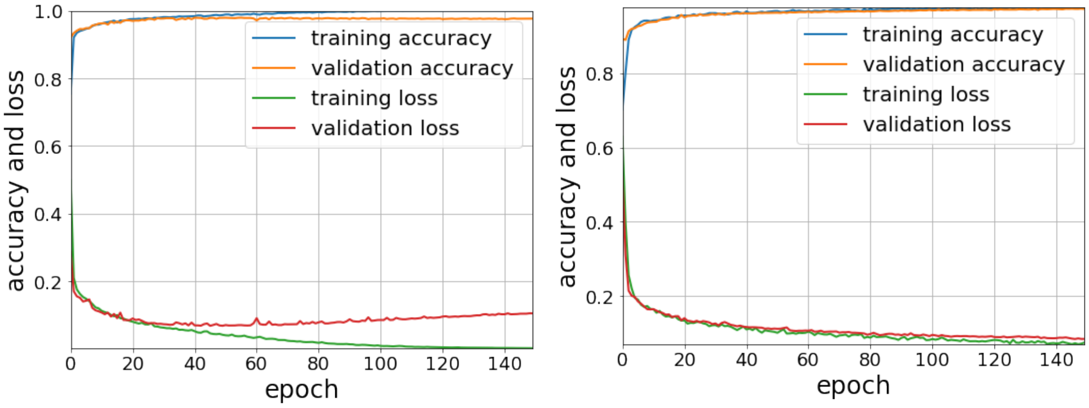


Figure 6.5: Cross-validation behaviour of the Stratified Shuffle Split [4].

Reducing Overfitting with Dropout Layers

Overfitting refers to a model that fits the training data too well while not fitting the validation as well. A common method to avoid overfitting is through regularisation. Aurelien Geron in his book "Hands-On Machine Learning with Scikit-Learn and Tensorflow" [21] proposes four methods for regularisation. Among those, the most popular, Dropout is used to drop certain neurons during training. Each neuron in the Dropout layer has a probability p of being dropped. Adding such layers to a network forces the neurons being used during training to learn general features during training. It is common to set a higher probability of dropout for the dense layers at the beginning of the network.
The model is updated by adding a few dropout layers after each MaxPooling layer, Flatten layer and the first Dense layer and compare the accuracy and loss of these model in Figure 6.6. Figure

6.6a shows the loss of the model presented in Figure 6.4. This model quickly overfits after the 30^{th} epoch. The loss is shown in Figure 6.6b is one of the new models with added dropout layers. This model fits perfectly the data even after 150 epochs. This shows that adding a few Dropout layers can regularise this model.



(a) With no Dropout layers

(b) With Dropout layers

Figure 6.6: Model loss with and without Dropout layers.

Classifier performance

The final model for the classifier is a deep convolutional neural network with three Convolutional layers, two MaxPooling layers, two Dense layers, and four Dropout layers are presented in Figure 6.7 accounting for a total of 522,112 trainable parameters.

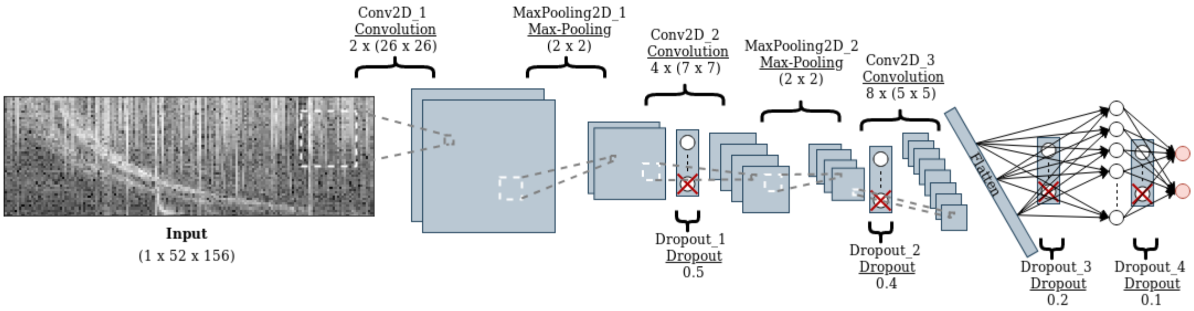


Figure 6.7: Final architecture of the deep convolutional neural network classifier with additional Dropout layers.

The model is trained in batches of 250 samples from a dataset of 11148 samples comprising 8317 whistler cuts and 2831 noise cuts. The dataset was divided into 5 folds of 150 epochs each with a validation set percentage of 33% (7469 samples for training, 3679 samples for validation) following the stratified shuffle split cross-validation techniques. We use a Stochastic Gradient Descent (SGD) with a learning rate of 0.001 and a momentum of 0.9 as the optimiser. The loss and accuracy of the training are shown in Figure 6.8. The parameters above are the most commonly used as a starting point for training this type of model.

The plots in Figure 6.8 are the curves with error bars of the accuracy and loss of both the

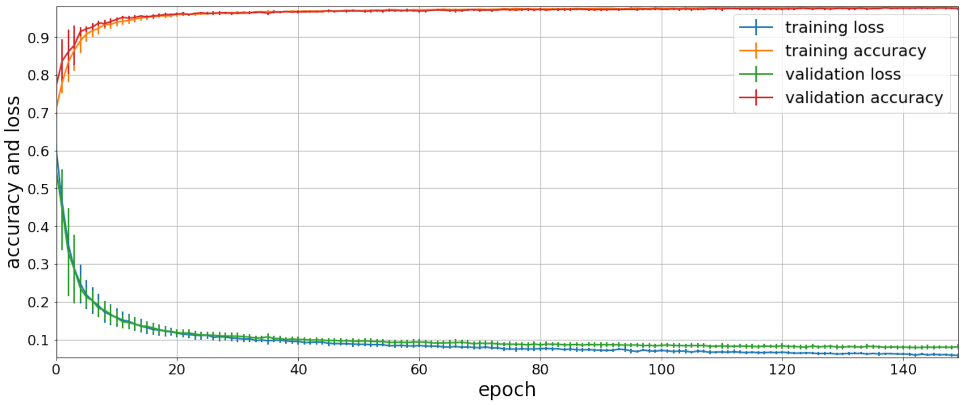


Figure 6.8: Accuracy and loss of the training using 5 fold stratified shuffle split cross-validation on 7439 training samples and 3679 validation samples.

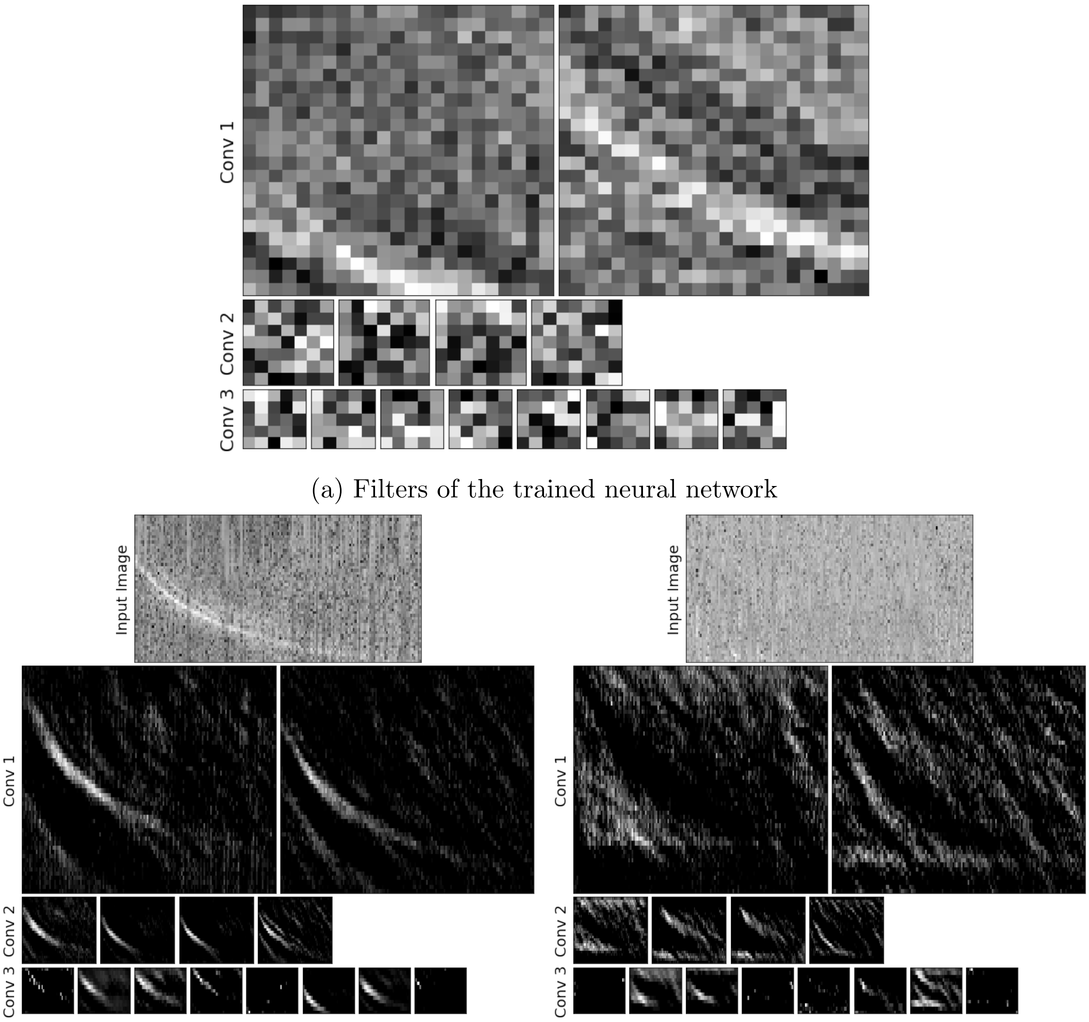
training and validation set of the model. The error in the curves decreases as training continues meaning that the accuracy and loss of both training and validation set converge to the same constant value at each fold. Moreover, both training and validation accuracy converge to 1 after the 80^{th} epoch, while the training and validation loss do not converge to the same value but diverge after the 50^{th} epoch. To reduce the effect of this divergence, the model should be trained for 100 epochs.

The convolutional layers provide insight into the decisions taken by the network. Figure 6.9 shows the convolutional filters learnt by the model as well as their associated feature maps for a whistler cut and a noise cut.

The two filters of the first convolutional layer (Figure 6.9a) show the most basic shapes that are looked for by the model in an image. These filters both have the same feature (line with decreasing gradient) but at a different location in the filters. The similarities in the features can be because only whistlers are present in the whistler cuts and no distinct shapes are present in the noise cuts. As we go deeper into the network, the filters have a more complex shape that is not easily understood because of the low resolution of these filters. The feature maps, however, present more information on the model's prediction. Figure 6.9b and 6.9c show the feature maps of the model when it is respectively presented with a whistler cut and noise cut. As expected, both feature maps of the first convolutional layer present the feature corresponding to lines of decreasing gradient in the image. In Figure 6.9b, the shape of the original whistler can be clearly observed while in Figure 6.9c, there are no distinct shapes being observed. These features are more and more refined into different new features as we move to the deeper layers of the network.

6.1.3 Detection with sliding DCNN

The second part of the design is using the trained classifier to perform object detection. We do so by generating the probability of classification of each portion of the preprocessed spectrogram and using a detector to make the final decision on the location of the whistlers.



(a) Filters of the trained neural network

(b) Feature maps when detecting a whistler cut

(c) Feature maps when detecting noise cut

Figure 6.9: Filters and Feature Map of the trained model.

Sliding classification

A challenge of object detection is to locate an unknown number of object in an image. This problem cannot be solved directly as an object classification problem since a classifier has a fixed input shape. Let assume that our spectrogram after preprocessing is a $M \times P$ image, the classifier size was initially chosen such that it classifies images of size $M \times N$ with $N < P$. The spectrogram image cannot then be classified directly, instead, we generate a region of interests (RoI) with sizes that match the classifier. Since both spectrogram and classifier have the same width, the RoI are extracted by sliding the classifier along the length of the spectrogram. The resulting number of RoI N_{RoI} is then given by:

$$N_{RoI} = \lfloor \frac{P-N}{S} \rfloor \tag{6.1}$$

with the stride S, the distance between two consecutive RoI. The finest resolution can be obtained with $S = 1$ when $N_{RoI} = P - N$ and the ROI are distant by the spectrogram time resolution. However, the finer the resolution, the more computationally expensive the detection become. To reduce the computation time, S can be chosen such that the ROI are distant by 0.1s. A stride of 1 will be used in the following section to explain the algorithm of the detector.

The result of the classifier on the spectrogram by generating regions of interest with a stride of 1 is shown in Figure 6.10. The result is a curve of the whistler's probability of detection in the preprocessed spectrogram. The probability of detection is below 0.5 in most of the spectrogram except at time between 2 and 2.7 seconds where a series of whistlers are observed (Figure 5.15). Moreover, Figure 6.10 shows the presence of two whistlers with the lowest probabilities at 0.8 and different thickness.

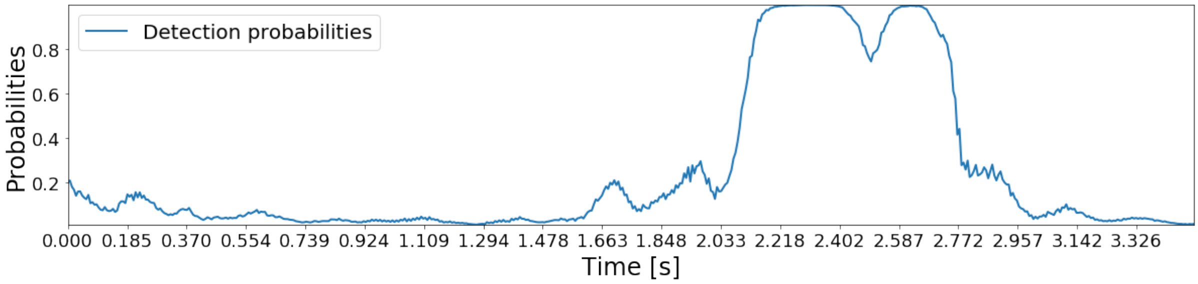


Figure 6.10: Probabilities of detection using the preprocessed spectrogram in Figure 5.15

Detector

The last phase of the detection is the decision of the whistler location. The classifier performs quite well on the dataset with an accuracy of 0.976 and a loss of 0.08. As a result, finding the peak of a detected whistler is quite hard since a group of whistlers have a detection probability of 1. Consequently, using a static threshold is not the perfect option, instead we find the peak of the whistlers by finding the peaks of the concave down portions of the spectrogram above a probability threshold. For this detector, we select an arbitrary threshold of 0.9 (90% probability). The result of this detection is shown in Figure 6.11. The two whistlers present in the spectrogram are detected at time t=2.334s and t=2.622s.

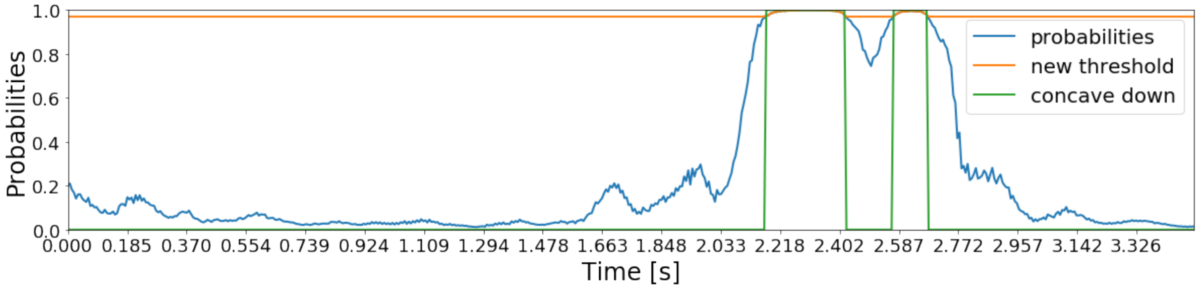


Figure 6.11: Detector decisions with a new threshold at 0.9 with two whistlers detected at $t = [2.334, 2.622]s$.

6.1.4 Detector Output

After finding the starting location of whistlers, the ending location is found using the length of the whistler cut use for training the data. The final result of the detection for the sample in Figure 5.15 showing the bounding box prediction with the probability of detection is shown in Figure 6.12. Two whistlers are found at t=2.3 and t=2.6 seconds. All detected whistlers have similar bounding boxes since the frequency range and the end time are static (1.5kHz to 9.5kHz and a duration of one second).

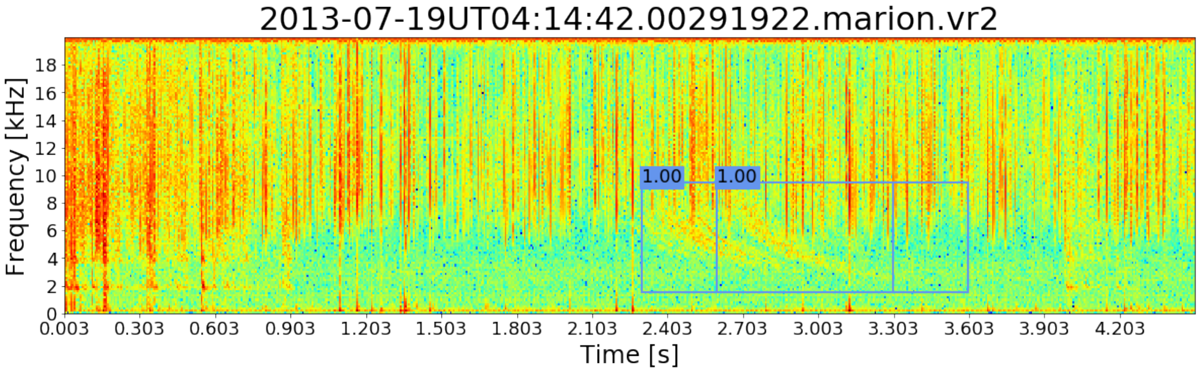


Figure 6.12: Bounding box prediction using the sliding deep convolutional neural network classifier.

6.1.5 Time Complexity

This design has three processes in its testing pipeline. It was already found that preprocessing has a time complexity of $O(N)$. The second process has a complexity that varies with the duration of the spectrogram, thus its time complexity is also $O(N)$. The detector's time complexity also depends on the duration of the spectrogram and consequently has a time complexity of $O(N)$. The time complexity of the entire process is thus $O(N)$. It is important to note that the sliding process involves doing a deep convolutional neural network classification which is costly. Therefore, this design might be significantly slower than the one proposed in Chapter 5.

6.2 You Only Look Once

As per 2018, You Only Look Once is the current state of the art in object detection. In this section, we adapt the implementation of YOLO to create a whistler detector and localiser.

6.2.1 Design

The design proposed in this chapter make uses of the state of the art in object detection as a method of detecting whistlers in a spectrogram. An overview of the design is presented in Figure 6.13. This design has two pipelines, a training stage, and testing pipeline. The training pipeline has three parts during which preprocessed samples from the training set are used to generate YOLO formatted labelled using the AWD labels and are later used to train a pre-trained YOLO model. The testing pipeline has two parts in which the preprocessed samples from the testing set are evaluated on the new YOLO model.

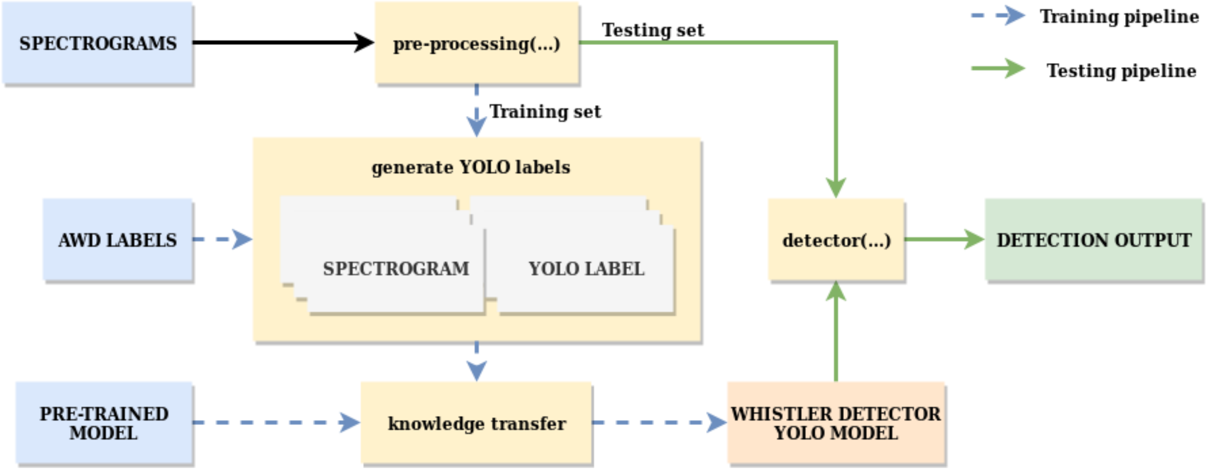


Figure 6.13: Overview of the whistler detector design using a YOLO model.

In the following sections, the choice of the inputs to the pipeline as well as the design decision for the processes are elaborated.

6.2.2 YOLO dataset generation

YOLO is a supervised machine learning algorithm which uses bounding boxes and class label as label data. YOLO requires five parameters from every labelled object in the form of (c, x, y, w, h), with x and y, the normalised coordinate of the centre of the object, w and h, the normalised width and height of the object, and c the object class.

The labelled dataset generated for training our model using YOLO is obtained from our training set. Each vr2 sample is converted to a spectrogram, pre-processed, converted into an RGB image and then labelled using the AWD by Lichtenberger [7] according to the cut shown in Figure 5.2. Figure 6.14 shows a sample from the training set and its YOLO label.

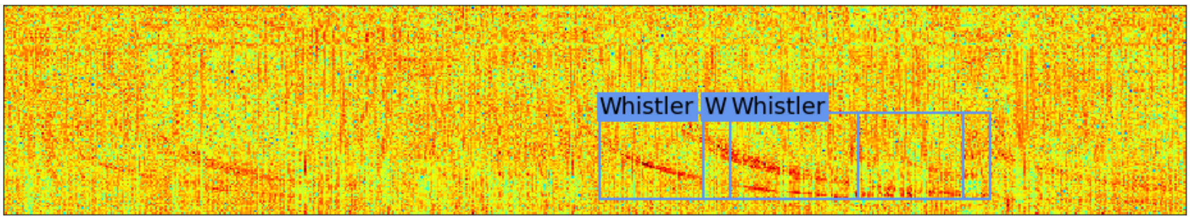


Figure 6.14: Sample from the labelled dataset used for training the YOLO model. Three whistlers are detected by AWD and the YOLO labels are $[0, 0.58, 0.73, 0.17, 0.45]$, $[0, 0.67, 0.73, 0.17, 0.45]$, and $[0, 0.69, 0.73, 0.17, 0.45]$

6.2.3 Knowledge Transfer

Knowledge transfer in the domain of machine learning is used to transfer the knowledge acquired by a trained network to a new network[6] and is based on the assumption that similar tasks require the same underlying knowledge. The common Deep Neural Network (DNN) has two main components, a convolutional base and a classifier. The convolutional base typically learns hierarchical feature representation of the data (according to Yosinski[30], in such cases, there is a general to specific learning process occurring in the network) while the classifier is generally a set of fully connected layers that classify an image according to its features. The first layers of the network thus contain features that are domain-independent while the last layers contain domain-dependent features. There are two main strategies in knowledge transfer known as fine-tuning and transfer learning. Fine-tuning consists of training a pre-trained network. This strategy is computationally expensive especially for very deep neural networks. The second strategy is to freeze the first few layers of the pre-trained network before training. This results in faster training time at the expense of the network performance. To reduce training time during experimentation, we use transfer learning to train the pre-trained YOLO model.

YOLO Model selection

Few pre-trained YOLO models are available and among those trained by Redmon *et al.*[32], YOLOv3-spp and YOLOv3-tiny are the one of interest. The dataset used contain images such as animals, vehicle types, and much more common image used in object detection.The performance of these models is presented in Table 6.1 Table 6.1 presents the performance of YOLOv3-spp and YOLOv3-tiny on the COCO dataset. The performance metric for object detection are:

Table 6.1: Performance on the COCO Dataset.

Model	**Train**	**Test**	**mAP**	**FLOPS**	**FPS**	**Size**
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	34M
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	241M

mAP : mean precision average [33]

FLOPS : number of floating point operations per minute

FPS : number of frame per second

Size : size of the pre-trained model

YOLOv3-spp is the latest YOLO pre-trained model and is more accurate (60.6 mAP) than YOLOv3-tiny but at the expense of more computations (26 times more than YOLOv3-tiny). YOLOv3-tiny will be the model of choice during experimentation.

Training using Darknet

Darknet is an open-source framework for neural network written in C and CUDA and supporting both CPU and GPU[34]. It moreover, provides easy commands for interacting with YOLO models.
Using Darknet, we train both YOLOv3-tiny and YOLOv3-ssp using and transfer learning (freezing the convolutional layers) until the average loss is almost constant or below 0.06. The following Darknet commands were used to train the models on a Linux operating system:
Freezing layers

```
    ./darknet partial <cfg> <model> <name>  <layer_nbr>
```

Training network

```
    ./darknet detector train <data> <cfg> <model> -clear 1
```

Predict

```
    ./darknet detect <cfg> <model> <image>
```

with <data>, the path to the .data file, <cfg> the path to the cfg file, <model>, the path to the YOLO model, <name>, the name of the new model, <image>, the image for prediction, and <layer_nbr>, the number of the last frozen layer.

6.2.4 Detector Output

Once the whistler detector is trained, the result of the detection is a list of bounding boxes in YOLO format along with the class label and class probabilities. The result of a detector trained using the training set with non-processed sample for the sample in Figure 5.20 is shown in Figure 6.15. The YOLO network detected two whistlers with a probability of 0.97 and 1, starting time of 2.24 and 2.64 seconds and duration of 1s.

The object of interest for this project are whistlers. As explained in the literature, whistlers have a simple shape, consequently, a convolutional neural network with at a few layers should be enough to classify the whistlers from the noise.

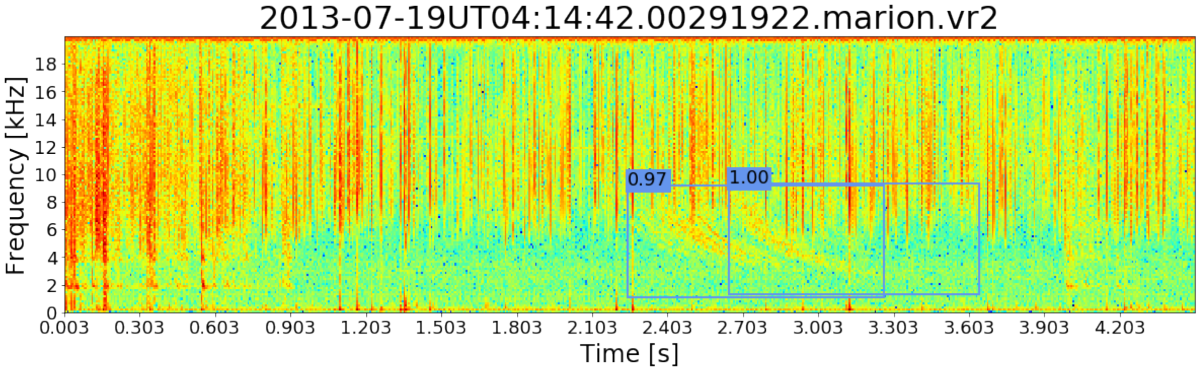


Figure 6.15: Result of detection using YOLO with a simulated whistler. Two whistlers are detected at t=2.24s and t=2.64s.

6.2.5 Time Complexity

This design has two processes in its testing pipeline. The preprocessing of the spectrograms has a time complexity of $O(N)$ while the detection using the YOLO model has a time complexity of $O(1)$. The time complexity of this design is thus $O(N)$.

6.3 Summary

This chapter presented the design and implementation of two whistler waves detector using a machine learning approach. The first design solves the object detection problem by using a deep convolutional neural network to classify region of interest generated from preprocessed spectrograms. The second design uses the current state of the art to solve this object detection problem by looking at the spectrogram once instead of generating regions of interest. While the time complexity of the first method grows linearly, the one using YOLO is constant if pre-processing is not taking into account.

Chapter 7

Model performance and validation

This chapter presents the performance of the detectors designed in Chapters 5 and 6. A definition of the benchmark is presented as well as the different evaluation metrics. This chapter outtlines the selection process for the suitable detector for each design is done using parameters tunning. Finally the performance of all three detectors are evaluated on Marion's testing set and SANAE IV's dataset.

7.1 Defining the benchmark

The Automatic Whistler Detector developed by Lichtenberger *et al.* [7] has been used to generate the labels of the data used for this research. His method outputs the 5kHz time of the detected whistler. In this research, the starting and ending time of each whistler are of interest. Therefore, we extract the starting point of the whistler located at 5kHz using the statistic obtained from whistlers in the training set.
Figure 5.3 shows the starting point of the mean of all whistlers in the training set and their corresponding AWD output. The starting point is located 0.2s before the 5kHz. We, therefore, use this result to approximate the starting location of all whistler detected by the AWD. It is important to note that this approximation itself is not exact and can lead to error in the evaluation. Figure 7.1 shows the starting location of whistlers detected using the method proposed in Chapter 5 and the approximated starting location of the whistlers detected by the AWD.

Three out the five whistlers detected by AWD are detected using this method, however, these results point to the presence of two series of whistlers starting at 2.2 s and 2.6 s. Therefore, using the starting location can be misleading. Instead, the detector is evaluated based on the number of detected whistlers (Figure 7.2). Therefore, the following metrics are defined:

- **True Positive** TP: The number of AWD starting point that are within the bounding boxes of our detection.
- **False Positive** FP: The number of bounding boxes which do not contain any whistlers

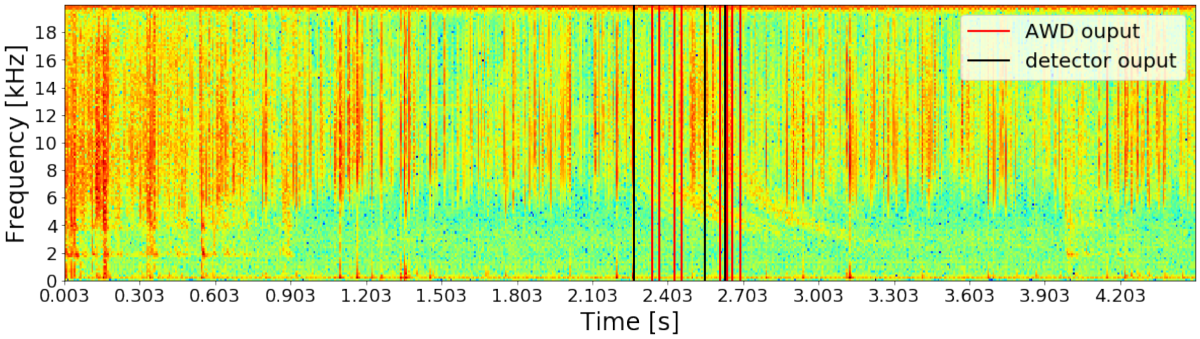


Figure 7.1: Starting location of the AWD method and a detector output.

detected by AWD.

- **False Negative** FN: The number of whistlers found by the AWD that are not within any of our bounding boxes.
- **True Negative** TN: The noise detected by both AWD and the new methods. This is not applicable for this research since we are not interested in detecting noise.

Figure 7.2 shows the output of both AWD and the predicted bounding boxes. The whistlers at 2.4 and 2.5s located by AWD are within the bounding box starting at 2.1s and therefore are true positives. The whistler at 0.1 was not detected by this method, this is a false negative. The whistlers in the bounding box starting at 2.8s were not detected by AWD and therefore are false positive.

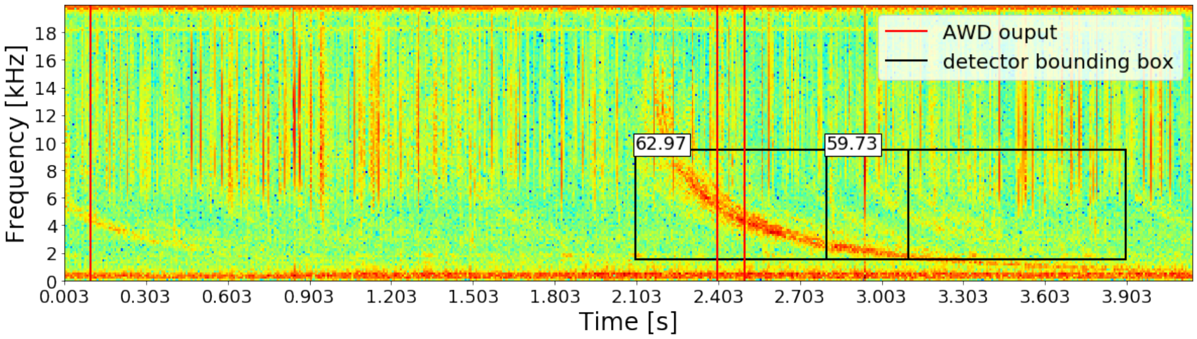


Figure 7.2: Metrics of evaluation of whistlers.

AWD provides a figure of merit for each detected whistler. However, no information related to its evaluation was found in the literature, therefore, it cannot be used as an appropriate benchmark for the evaluation of figure of merit associated with each detected whistler.

7.2 Metrics

The evaluation of the designs is performed based on a few metrics. The confusion matrix is composed of the number of occurrences of TP,FP,FN for the training set. This matrix is used to define other related measures such as:

- **Precision or Positive Predictive Value** PPV: The portion of the positive values that are true positives and is expressed as:

$$PPV = \frac{TP}{TP + FP}$$

- **False Alarm Rate**: The portion of the positive value that are false positive and is expressed as:

$$FAR = \frac{FP}{TP + FP} = 1 - PPV$$

- **Recall or True Positive Rate** TPR: The portion of the true positive that are successfully retrieved and is expressed as:

$$TRP = \frac{TP}{TP + FN}$$

- **Misdetection** The portion of the predicted positives that are false positive and is expressed as:

$$M = \frac{FP}{TP + FN} = 1 - TNR$$

- **F1-score**: Measures the test's accuracy by calculating the harmonic mean of the precision and recall and reaches its best at 1 and lowest at 0. It is expressed as:

$$F1_{score} = 2 \times \frac{PPV \times TRP}{PPV + TRP}$$

7.3 Best Model Performance

The design and implementation of a few whistler detector and localiser was presented in Chapters 5 and 6. These designs, however, are not the ones achieving the best performance of both the training set of the data collected at Marion. In this section, the parameters of each design is altered to obtain the best performance on the training data.

7.3.1 Cross-Correlation using a Simulated Whistler (CCSW)

This CCSW design has three inputs whose parameters can be altered to find the best model. For the spectrogram, the 1D preprocessing method can be changed. The kernel's frequency range f and zero-dispersion D_0 can be adjusted. And finally, the parameters of the detector (N,G,X_{dB},k,T_S,T_L) can be adjusted.

Firstly, the detector is observed. The fusion detector is a combination of three CFAR detectors. The parameters of the CA CFAR (N,G,X_{dB}) are adjusted by setting k,T_s, and T_l to zero[1]. Figure 7.3 presents the performance of the CCSW detector on the 200 randomly selected sample from the training set when varying X_{dB},G, and N. The precision increases as X_{dB} increases, while the recall and f1 score decrease. For each X_{dB}, the precision decreases as G and N decrease while the recall and the f1 score increases for these behaviours of G and N. The precision and

[1] TM CFAR with $T_s = T_l = 0$ is just a CA CFAR, therefore, the fusion detector behaves just like a CA CFAR.

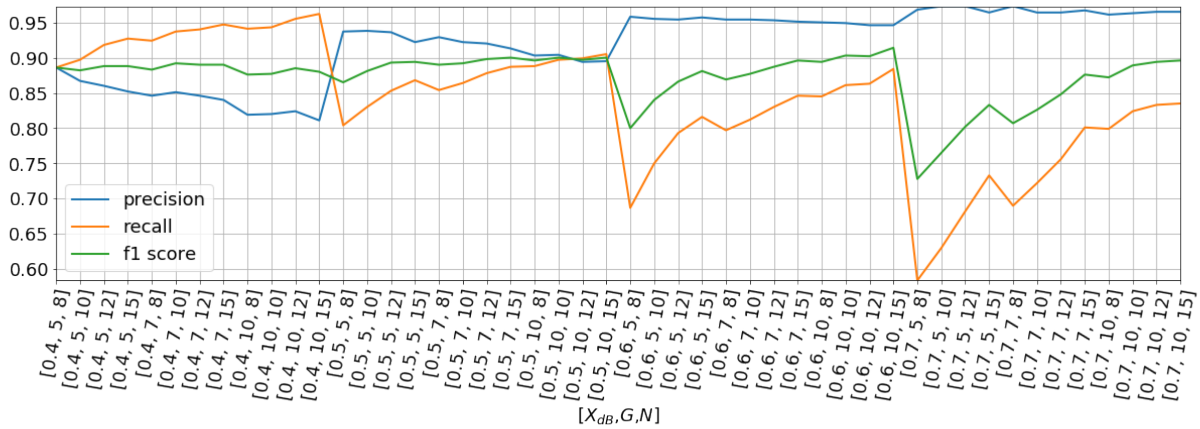


Figure 7.3: Performance of the CCSW with zscore preprocessing with $1.5kHz \leq f \leq 9.5kHz$, $k = T_s = T_l = 0$.

recall have opposite behaviour per parameter change. Since the ground truth for this project is not absolute, the best fit at this level is the performance at which produces one of the highest f1 scores while having similar precision and recall value. Thus the following values: $N = 12$, $G = 10$, and $X_{dB} = 0.5$ are selected as best parameters for the CA CFAR.

The second set of parameters to be evaluated are the ones for the TM and OS CFAR. Figure 7.4 shows the performance of the detector when varying k,T_s, and T_l. The precision increases for increasing values of T_l and decreases for increasing value of T_l above 3. $T_l = 3$ is a pivot point. Below this point, the precision is higher than the recall and this changes after $T_l = 3$. This implies that removing more than the highest three cells used for the noise power calculation results in a decrease in false negatives, and therefore in an increase in true positives. The values of k and T_s increase the precision below $T_l = 3$ and do the opposite above this value.

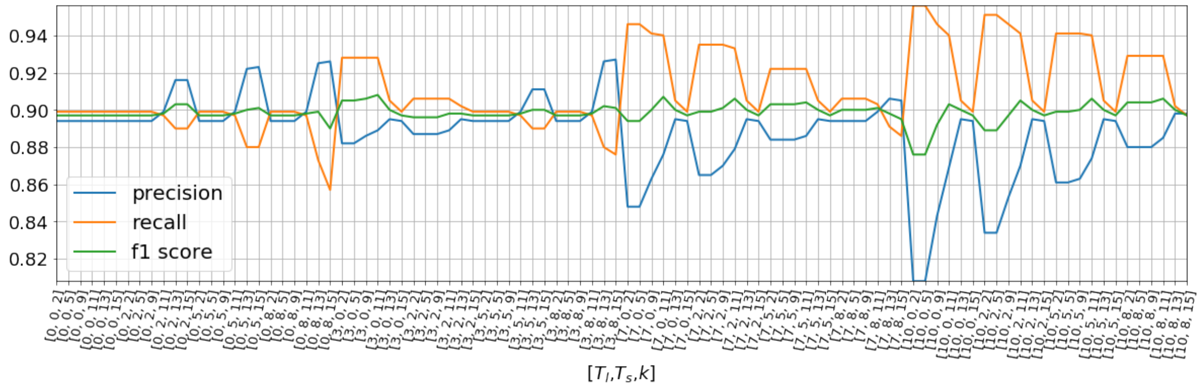


Figure 7.4: Performance of the CCSW with zscore preprocessing with $1.5kHz \leq f \leq 9.5kHz$, $N = 12$,$G = 10$,$X_{dB} = 0.5$.

The best performance can be selected for values of $T_l \leq 3$. We select $T_l = 3$, $T_s = 5$, and $k = 13$. This detector achieve a precision of 0.892, a recall of 0.894, an f1 score of 0.893, a misdetection of 0.106 and a false alarm rate of 0.108.

After tuning the detector's parameters for the Z score preprocessing and a frequency range of 1.5kHz to 9.5kHz, the same is done on different preprocessing and the frequency range used by

AWD. Table 7.1 presents the performance of the best models on Marion's training set.

Table 7.1: Performance of CCSW detector with different tuning parameters.

Parameters			**Performance**		
Preprocessing	Kernel $[f_{min}, f_{max}]$	Detector (N,G,X_{dB},k,T_S,T_L)	Misdetection	False Alarm	F1 score
None	[1.5,9.5]	(10,5,0.3,13,5,7)	0.092	0.199	0.851
Zscore	[1.5,9.5]	(12,10,0.5,13,5,3)	0.106	0.108	0.893
Zscore	[4.5,11.5]	(15,7,0.7,15,2,7)	0.045	0.174	0.886
C. Detrend	[1.5,9.5]	(8,5,0.4,5,0,7)	0.062	0.159	0.887
L. Detrend	[1.5,9.5]	(12,10,0.4,13,8,5)	0.088	0.124	0.894

All detectors achieve a good f1 score of 0.882 on average on Marion's training set. Without any preprocessing, the detector achieved the worst overall performance with an f1 score of 0.851. A kernel with the frequency range between 4.5 and 11.5 kHz as selected in the AWD results in a good misdetection rate but a high false alarm. Preprocessing using either the Z score, a constant or linear detrend with $1.5 \leq f \leq 9.5$ results in similar performances on both training and testing set. However, the Z score performs quite well with both misdetection and false alarm of about 10% for the training set.

7.3.2 Sliding Deep Convolutional Neural Network (SDCNN)

The SDCNN requires the generation of regions of interest and a trained classifier. In this section, the performance of this detector is evaluated for different preprocessing techniques with RoI generated such that two consecutive RoI from a spectrogram is distant by 0.1s. For each preprocessing technique, the classifier in Figure 6.7 was trained for 100 epochs with the remaining parameters being the same as before. The results of the evaluation on Marion's training set is tabulated in Table 7.2.

Table 7.2: Performance of the sliding deep convolutional neural network on Marion's training set.

Parameters	**Classifier**		**Performance**		
Preproessing	Accuracy	Loss	Misdetection	False Alarm	F1 score
None	0.974	0.087	0.089	0.040	0.935
Zscore	0.977	0.077	0.092	0.036	0.935
C.Detrend	0.979	0.076	0.102	0.030	0.932
L. Detrend	0.970	0.096	0.109	0.025	0.931

Table 7.2 shows the accuracy and loss of the classifier during training as well as the performance of the detector on Marion's training set. For each preprocessing technique, the classifiers reached an accuracy and a loss of respectively about 0.975 and 0.084 on average. All detectors achieved a high f1 score around 0.933 with a misdetection rate between 0.089 and 0.109 and a false alarm

rate below 0.04 on the training set.

The best detectors are obtained when spectrograms are not preprocessed or when preprocessed with the Z score transform. Both detectors have the same f1 score but the Z score has slightly higher misdetection and a lower false alarm rate.

7.3.3 You Only Look Once (YOLO)

After training the YOLO model on Z score preprocessed spectrograms, the same model is trained on data preprocessed with different other techniques. The results of the evaluation of these new detectors on Marion's training set are tabulated in Table 7.3.

Table 7.3: Performance of the sliding deep convolutional neural network on Marion's training set.

Parameters	**YOLO**	**Performance**		
Preprocessing	Loss	Misdetection	False Alarm	F1 score
None	0.351	0.085	0.002	0.955
Zscore	0.352	0.075	0.001	0.961
C. Detrend	0.361	0.109	0.002	0.942
L. Detrend	0.364	0.057	0.001	0.97

After 280000 iterations, the models' loss was 0.357 on average with the linear detrend techniques having the highest loss. All detectors perform very well on Marion's training set with an average of 0.082, 0.002, and 0.957 respectively for the misdetection, false alarm, and f1 score. Among these detectors, the ones with the Z score and the linear detrend have the highest performance, however, the latter achieves a lower misdetection and false alarm rate on the training set. The model trained using preprocessed samples with linear detrend is, therefore, the best model.

7.4 Models comparison

All detectors each output the results of the whistlers' detection and localisation given an vr2 file with each detection having a figure of merit. These detectors thus meet requirement R_1 (Table 3.1).

The best detectors from the three designs developed in this study are evaluated on Marion's testing set. The performance of these detectors are tabulated in Table 7.4. All detectors perform very well on Marion's testing set with the poorest performance with an f1 score of 0.942 from the CCSW. The best performance is achieved by the SDCNN with an f1 score of 0.939, a misdetection rate of 0.093 and a false alarm rate of 0.026. All detector satisfy specification S_2 from Table 3.3 with all misdetection and false alarm below 20%.

The processing time of the detectors was also evaluated by computing the processing ratio r (see requirement 2 in Table 3.2), defined as the ratio between the processing time of a detector on a sample and the duration of this sample. This evaluation was performed on 5 batches of

Table 7.4: Performance of each detector on Marion's testing set.

Detector	**Performance**		
	Misdetection	False Alarm	F1 score
CCSW	0.096	0.094	0.906
SDCNN	0.093	0.026	0.939
YOLO	0.141	0.021	0.915

50 samples from Marion's training set. The results of the evaluation is tabulated in Table 7.5. All detectors have a ratio below 0.5, this indicates each detector takes less than half of the data collection time to process the data. This satisfies requirement R_2 and specification S_3.

Table 7.5: Ratio of processing time over sample duration for each detector.

	CCSW	**SDCNN**	**YOLO**
r	0.179	0.376	0.010

The detectors were also evaluated on SANAE IV's dataset despite the poor quality of this dataset. Table 7.6 presents the performance of the detector on SANAE IV's dataset. The performance of the detectors on SANAE IV's dataset is worst compared to the one on Marion's testing set. The false alarm rate is still below 20%, so more than 80% of the detections were correct. However, the misdetections are considerably high with the worst performance reaching a misdetection of 0.815 and 0.91. Yet, the best performance is achieved by the SDCNN with a misdetection of 0.532, a false alarm rate of 0.055 and an f1 score of 0.626.

Table 7.6: Performance of each detector on SANAE IV's dataset.

Detector	**Performance**		
	Misdetection	False Alarm	F1 score
CCSW	0.91	0.157	0.04
SDCNN	0.532	0.055	0.626
YOLO	0.815	0.043	0.31

The corruption of SANAE IV's dataset participated in degrading the detector's performance, however, the whistlers present in this data differ from the ones in Marion. Figure 7.5 shows the mean of the whistlers present in the data collected at SANAE IV. As opposed to the ones present in Marion's dataset (see Figure 5.3) which correspond to a simulated whistler of zero-dispersion D_0 of 80 (Figure 5.6), these whistlers correspond to simulated whistlers with a zero-dispersion of approximately 35 using the Bernard approximation. In fact simply by solely changing D_0 in CCSW from 80 to 35, the detector achieves an f1 score of 0.597 which is an improvement of 0.593. This emphasises the difference in whistler parameters between the ones collected at Marion and SANAE IV.

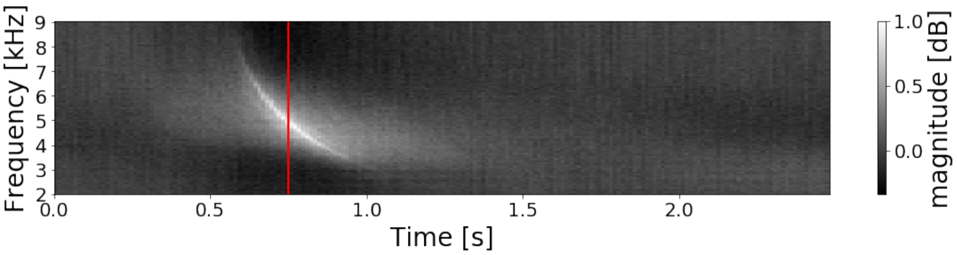


Figure 7.5: Mean of the whistlers present in SANAE IV's dataset.

7.5 Summary

This chapter compiled the evaluation of the detectors designed in Chapter 5 and Chapter 6.

The best model using the Cross-Correlation with a Simulated Whistler (CCSW) is one that uses a Z score transform for its preprocessing, a frequency range of 1.5 to 9.5 kHz for its kernel, and with a Linear Fusion CFAR with $N = 12$, $G = 10$, $k = 13$, $T_S = 5$, $T_L = 3$, and $X_{dB} = 0.5$. This model achieved a 90.61% on the f1 score with 9.6% and 9.4% for misdetection and false alarm on Marion's testing set.

The best model using the Sliding Deep Convolutional Neural Network (SDCNN) is achieved by selecting the Z score for its preprocessing. This model achieved a 93.9% on the f1 score with 9.3% and 2.6% for misdetection and false alarm.

The best model using You Only Look Once (YOLO) is achieved by selecting the Linear Detrend as its preprocessing. This model achieved a 91.5% on the f1 score with 14.1% and 2.1% for misdetection and false alarm.

Moreover, all detectors process each sample in less than half its collection time.

Chapter 8

Conclusion & Recommendations

8.1 Conclusion

The goal of this research was to design and implement new methods to detect and localise whistler radio waves that would meet the project requirements given in Tables 3.1 and 3.2. This research results in the design and implementation of three different detectors.

The first detector (CCSW) gets its inspiration from the current state of the art in whistler wave detection developed by Lichtenberger *et al.*[7]. This method makes use of the whistler's dispersion approximation developed by Bernard [8] to simulate a whistler matching best the mean of a few selected sample from the dataset. The simulated whistler is cross-correlated with preprocessed spectrograms of the sample outputting a curve which is a measure over times of the similarities between the simulated whistler and the section of the spectrogram being cross-correlated. This curve is fed to an adaptive threshold detector to detect the starting point of the whistler. Finally, portions of the spectrogram around the starting locations are cross-correlated with a few simulated whistlers to find the duration of the whistlers present in the spectrogram and therefore their bounding boxes. This method presents one challenge, the selection of the simulated whistler. This selection requires knowledge of the characteristics of the whistlers present in the data which is not always available just as in this research.

The second detector (SDCNN) uses a deep convolutional neural network as a classifier to detect the presence of whistler radio waves on generated Region of Interest (RoI). The classifier was trained on Marion's training set using cross-validation to find the option balance between specialisation and generalisation of the classifier. RoI are generated by cropping the spectrograms in regions of 1.5kHz to 9.5kHz and with 100ms interval between two consecutive RoI. The result of this detector is the result of the trained classifier on the RoI. A major challenge in this design is the lack of diversity in the data. With less than 1500 samples in the training set and all having approximately the same whistler's characteristics, the classifier does not learn the detect all type of whistler by those in the data.

The last detector (YOLO) uses the state of the art in object detection developed by Redmon *et al.*[2]. This detector is an adaptation of YOLO on the data available for this research. The main challenge with this detector is its failure to distinguish very close objects. Moreover,

it requires the training data to be labelled, with the labels being the bounding boxes of the whistlers in each spectrogram image. However, the labels provided by AWD do not provide such information, and therefore, an approximation was made. This approximation which generalised the bounding boxes introduced errors in the label which also affected the performance of this detector.
A few detection results of these three detectors are shown in Appendix C.

The traceability table of this research is presented in Table 8.1.

Table 8.1: Traceability table

Requirements	**Specifications**	**Acceptance Tests**
R_1	S_1	AT_1
	S_2	AT_2
R_2	S_3	AT_3

The first requirement goes as follow:
R_1: "The system should be able to count the number and time occurrence of whistlers in a data file." with its sub-requirements tabulated in Table 3.1. All sub-requirements (R_{11},R_{12},R_{13}) presented in Table 3.1 are met by allowing each detector to approximate a bounding box for each detected whistler. In addition, each whistler is given a figure of merit for its detection. In CCSW this figure of merit is the cross-correlation value of the simulated whistler with the preprocessed spectrogram. In SDCNN and YOLO, it is the probability of detection associated with the whistler. The resolution of the detection was set to be in 100ms per detection (S_1). This requirement was achieved by both CCSW and SDCNN. YOLO, however, is defaulted to use the highest possible resolution. The second specification requires the false alarm and misdetection of the detectors to be less than 20%. As per AT_2, all detectors were tested on the testing set. These detectors were able to achieve a misdetection and false alarm of less than 15% on Marion's testing set meeting specification (S_2) and thus closing off with requirement R_1.
The second requirement states that:
R_2: "The system should process any data faster than the generating time of the data, *i.e.* a data of length t should have a processing time of $t_{processing} \leq t$." The acceptance test for this requirement introduces the processing-to-data-collection ratio which was computed on 5 batches of 50 samples from the training set. All detectors achieved a processing-to-data-collection ratio of less than 0.5 (7.5). This implies that all detectors can process each samples twice as fast as its collection time. Therefore, requirement (R_2) is met.
The performance of the detectors on all available dataset are presented in Appendix B.

8.2 Limitations

A major limitation in this research is the availability of the data. The current state of the art in whistler radio waves detection was tested on 100 hours of raw data and was labelled manually by whistler specialists [7]. The data available for this research however, accumulates to only 3 hours of raw whistler data which are not labelled by any specialists but solely by AWD.

Not only the size of the dataset is small, but the data itself is not diverse. The available data was collected at Marion island and SANAE IV. The data from Marion contains whistlers with an average zero-dispersion of $D_0 = 80$ - using the Bernard approximation - while the one from SANAE IV contains whistlers with $D_0 = 35$. Moreover, the data from SANAE IV has a poor quality and was not used during training. This cause the detector to adapt well to whistlers with D_0 around 80 and perform poorly for whistlers with a different zero-dispersion.

8.3 Recommendations for Further Research

The recommendations below are made from the design, the outputs and the performance of the detectors developed in this study.

An analysis of the whistlers present in Marion's training set led to the conclusion the magnitude of these whistlers are highest between 1.5kHz and 9.5kHz of the spectrograms. The cross-correlation kernel used in CCSW uses a simulated whistler over the entire range. Reducing this range to one maximising the magnitude of the whistlers could lead to better performances. Therefore, the effect of the frequency range on the performance of the detectors should be investigated.

The characteristics of the whistler extracted from Marion's dataset matches only a certain type of whistlers with zero-dispersion of around 80. However, it was later found that SANAE IV's dataset has whistler with a different characteristic. The lack of diversity in the available data for this research constrained the generalisation of the detectors. It is recommended that for further investigations on whistler waves detection, a diverse dataset containing a few thousand samples from different sites should be available.

SDCNN and YOLO use supervised machine learning algorithm. While SDCNN requires at least two classes of label data, YOLO requires accurate bounding boxes around each whistler present in the data. The labels provided by AWD was not adequate for image classification and object detection. For further research, the data should be properly labelled. For accurate labelling, each sample should be manually labelled.

To alleviate problems associated with a lack of diversity in the data or poor labelling, the Bernard approximation of the dispersion of whistler waves could be used to simulate a dataset for the research. Machines learning models such are SDCNN and YOLO could be trained on the diverse simulated dataset and the learnt knowledge transferred on the real dataset training set.

Bibliography

[1] J. Lichtenberger, "A new whistler inversion method," *Journal of Geophysical Research: Space Physics*, vol. 114, no. 7, 2009.

[2] J. S. D. R. G. A. F. Redmon, "(YOLO) You Only Look Once," in *CVPR proceedings*, 2016, pp. 1–10.

[3] A. Tch, "The mostly complete chart of neural networks, explained," Aug 2017. [Online]. Available: https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

[4] "3.1. cross-validation: evaluating estimator performance¶." [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

[5] H. J. Christian, "Global frequency and distribution of lightning as observed from space by the Optical Transient Detector," *Journal of Geophysical Research*, vol. 108, no. D1, may 2003.

[6] Y. B. Goodfellow, I. and A. Courville, "Deep learning," 2016. [Online]. Available: https://www.deeplearningbook.org

[7] J. Lichtenberger, C. Ferencz, L. Bodnár, D. Hamar, and P. Steinbach, "Automatic Whistler Detector and Analyzer system: Automatic Whistler Detector," *Journal of Geophysical Research: Space Physics*, 2008.

[8] L. Bernard, "A new nose extension method for whistlers," *Journal of Atmospheric and Terrestrial Physics*, vol. 35, no. 5, pp. 871–880, 1973.

[9] R. A. Helliwell, "Whistlers and Related Ionospheric Phenomena," *Geophysical Journal of the Royal Astronomical Society*, vol. 11, no. 5, pp. 563–564, 1966.

[10] S. Kumar, D. Anil, A. Kishore, and V. Ramachandran, "Whistlers observed at low-latitude ground-based VLF facility in Fiji," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 69, pp. 1366–1376, Aug. 2007.

[11] M. A. U. (Eds.), *The Lightning Discharge*, ser. International Geophysics 39. Elsevier, Academic Press, 1987.

[12] "Ieee standard definitions of terms for radio wave propagation," pp. i–, 1998.

[13] W. C. Burgess, "Lighting-induced coupling of the radiation belts to geomagnetically conjugate ionspheric regions lighting-induced coupling of the radiation belts to geomatically conjugate ionosphere regions," no. May, 1993.

[14] R. L. Stenzel, "Whistler waves in space and laboratory plasmas," *Journal of Geophysical Research*, vol. 104, pp. 14 379–14 395, 1999.

[15] L. R. O. Storey, "An Investigation of Whistling Atmospherics," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 246, no. 908, pp. 113–141, 1953.

[16] R. A. Helliwell, J. H. Crary, J. H. Pope, and R. L. Smith, "The "nose" whistler—a new high-latitude phenomenon," *Journal of Geophysical Research (1896-1977)*, vol. 61, no. 1, pp. 139–142, 1956. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JZ061i001p00139

[17] E. Weisstein, "Cross-correlation." [Online]. Available: http://mathworld.wolfram.com/Cross-Correlation.html

[18] "xcorr2." [Online]. Available: https://www.mathworks.com/help/signal/ref/xcorr2.html

[19] M. Radar, *Principles of Modern Radar: Volume 3: Radar Applications*, 2014.

[20] D. Ivkovic, M. Andric, and B. Zrnic, "Detection of very close targets by fusion CFAR detectors," *Scientific Technical Review*, vol. 66, no. 3, pp. 50–57, 2016.

[21] A. Geron, *Hands-on ML with Sckit-Learn Keras & Tensorflow 2.0*, 2019.

[22] T. M. Mitchell, *Machine Learning*, 2011, vol. 17.

[23] N. S. Chauhan, "Introduction to artificial neural networks(ann)," Oct 2019. [Online]. Available: https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[25] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 1440–1448, 2015.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016.

[28] Y.-w. Liu, “The Short-Time Fourier Transform The Discrete Fourier transform (DFT) DFT as DTFT sampled in frequency,” Tech. Rep. 1, 2015.

[29] E. Kreyszig, H. Kreyszig, and E. J. Norminton, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ: Wiley, 2011.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” *Advances in Neural Information Processing Systems*, vol. 4, no. January, pp. 3320–3328, 2014.

[31] S. J. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*, 2010, vol. 3.

[32] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv*, 2018.

[33] J. Hui, “map (mean average precision) for object detection,” Apr 2019. [Online]. Available: https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

[34] J. Redmon, “Darknet: Open source neural networks in c,” http://pjreddie.com/darknet/, 2013–2016.

[35] E. Sejdi and I. Djurovi, “Time – frequency feature representation using energy concentration : An overview of recent advances,” vol. 19, pp. 153–183, 2009.

Appendix A

Preprocessing Techniques Theory

A.1 Spectrogram, A Frequency-Time Representation of Signals

This section explores the basic mathematical principles of generating a spectrogram from a sequence of data.

A.1.1 Finite Duration Signals Fourier transform

Let's assume that we have a discrete time signal x with N samples, the Discrete Time Fourier Transform (DTFT) is used to analyse the frequency content of the x and is obtained from the equation A.1. And the original signal can be obtained by the frequency components using equation A.2.

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) e^{-i\frac{2\pi k}{N} n}, k = 0, ..., N-1. \tag{A.1}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}(k) e^{i\frac{2\pi k}{N} n}, n = 0, ..., N-1. \tag{A.2}$$

Combining all sample values, equation A.1 can be represented as:

$$x = \frac{1}{N} F \hat{x} \tag{A.3}$$

$$\hat{x} = \bar{F} x \tag{A.4}$$

With F, a NxN Fourier matrix:

$$F = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{i2\pi\frac{1}{N}} & e^{i2\pi\frac{2}{N}} & \dots & e^{i2\pi\frac{N-1}{N}} \\ 1 & e^{i2\pi\frac{2}{N}} & e^{i2\pi\frac{4}{N}} & \dots & e^{i2\pi\frac{2(N-1)}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{i2\pi\frac{N-1}{N}} & e^{i2\pi\frac{2(N-1)}{N}} & \dots & e^{i2\pi\frac{(N-1)^2}{N}} \end{bmatrix} \tag{A.5}$$

And

$$\bar{F} = NF^{-1} \tag{A.6}$$

A.1.2 Short-Time Fourier Transform

The Short-Time Fourier transform is used to observe the change in frequency of a portion of the signal over time [35].

Consider the same signal as mentioned in section A.1.1. Now let's create a new signal X made of portions of length M of the signal x where portions are overlapped by K samples as depicted in the following sequence A.7.

$$\overbrace{x[0],\dots,\rlap{$\underbrace{}_{X_2:frame2}$}x[M-K],\dots,x[M-1]}^{X_1:frame1},\dots,x[2M-K-1],\dots,\overbrace{x[Q-M],\dots,x[Q-1]}^{X_{N-M}:frameN-M} \tag{A.7}$$

The new signal MxQ ($Q = \left\lfloor \frac{N}{M-K} \right\rfloor$) matrix X can be represented as:

$$X = \begin{bmatrix} X_0 & X_1 & X_2 & \dots & X_{Q-1} \end{bmatrix} \tag{A.8}$$

$$X = \begin{bmatrix} x[0] & x[M-K] & x[2(M-K)] & \dots & x[Q-M] \\ x[1] & x[M-K+1] & x[2(M-K)+1] & \dots & x[Q-M+1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x[M-1] & x[2M-K-1] & x[3M-2K-1] & \dots & x[Q-1] \end{bmatrix} \tag{A.9}$$

Typically, the selection is made using a window (*e.g.* Hamming, Turkey, ...). The mathematical expression of the DTFT of the matrix X also knwon as the Short-Time Fourier Transform (STFT) is given by:

$$\hat{X}(f,t) = \sum_{t=0}^{Q-1} \sum_{n=0}^{N-1} x[n]w[n-t(M-K)]e^{-j\frac{2\pi f}{N}n} \tag{A.10}$$

where

x is the discrete time signal

w is a window of size M

K the overlap between consecutive windows, with $0 \leq K < M$

$f = 0, 1, 2, \dots, N-1$

This time-frequency representation of the signal is known as a spectrogram and can be represented as

$$\hat{X}(f,t) = \bar{F}(f)X(t) \tag{A.11}$$

and its inverse by

$$X = \frac{1}{M} F \hat{X} \tag{A.12}$$

With F a MxM Fourier matrix:

$$F = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{i2\pi\frac{1}{M}} & e^{i2\pi\frac{2}{M}} & \dots & e^{i2\pi\frac{M-1}{M}} \\ 1 & e^{i2\pi\frac{2}{M}} & e^{i2\pi\frac{4}{M}} & \dots & e^{i2\pi\frac{2(M-1)}{M}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{i2\pi\frac{M-1}{M}} & e^{i2\pi\frac{2(M-1)}{M}} & \dots & e^{i2\pi\frac{(M-1)^2}{M}} \end{bmatrix} \tag{A.13}$$

And

$$\bar{F} = MF^{-1} \tag{A.14}$$

In practical, F can be any square matrix of at least MxM size, since the Fast Fourier Transform is used to each row of the matrix X.

A.2 Z score

The z-score of a data point is the number of standard deviations from the mean that the data point is and is mathematically expressed by:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{A.15}$$

with z_i and x_i respectively the z-score and the value of data point i, μ and σ, the mean and standard deviation of the data.

Let's denote by T the set of N points on either a time or frequency cut of the spectrogram, x_i and z_i the density of any point prior and before the z-score transformation of T. After application of the Z-score transformation on T, the mean μ_z of $T_z = Z_{score}(T)$ becomes

$$\begin{aligned} \mu_z &= \frac{\sum_{i=0}^{N-1} z_i}{N} \\ \mu_z &= \frac{\sum_{i=0}^{N-1} x_i - \mu}{N\sigma} \\ \mu_z &= \frac{\sum_{i=0}^{N-1} x_i}{N\sigma} - \frac{\sum_{i=0}^{N-1} N\mu}{N\sigma} \\ \mu_z &= \frac{\mu}{\sigma} - \frac{N\mu}{N\sigma} \\ \mu_z &= 0 \end{aligned} \tag{A.16}$$

As for the standard deviation σ_z of $T_z = Z_{score}(T)$, it becomes

$$\sigma_z = \sqrt{\frac{\sum_{i=0}^{N-1}(z_i - \mu_z)^2}{N}} = \sqrt{\frac{\sum_{i=0}^{N-1} z_i^2}{N}}$$

$$\sigma_z = \sqrt{\frac{\sum_{i=0}^{N-1} \frac{(x_i-\mu)^2}{\sigma^2}}{N}}$$

$$\sigma_z = \sqrt{\frac{1}{\sigma^2} \frac{\sum_{i=0}^{N-1}(x_i-\mu)^2}{N}}$$

$$\sigma_z = \frac{1}{\sigma}\sqrt{\frac{(x_i-\mu)^2}{N}}$$

$$\sigma_z = \frac{1}{\sigma}\sigma$$

$$\sigma_z = 1 \tag{A.17}$$

The Z-score transform thus transform a one dimensional dataset to one with mean of zero and a standard deviation of one.

A.3 Detrending with constant

Let x be a element of a series X and D be the transform defined by equation A.18.

$$x_c = D_c\{x\} = x - c \tag{A.18}$$

D detrends the series X by a constant c. The mean μ_c of the transformed series X_c is then:

$$\mu_c = \frac{\sum_{i=1}^{N} x_d}{N}$$

$$\mu_c = \frac{\sum_{i=1}^{N} x - c}{N}$$

$$\mu_c = \frac{\sum_{i=1}^{N} x}{N} - \frac{\sum_{i=1}^{N} c}{N}$$

$$\mu_c = \bar{x} - c = \mu - c \tag{A.19}$$

And the standard deviation σ_c of the series X_c is given by:

$$\sigma_c = \sqrt{\frac{\sum_{i=1}^{N}(x_c-\mu_c)^2}{N}}$$

$$\sigma_c = \sqrt{\frac{\sum_{i=1}^{N}(x-c-(\mu-c))^2}{N}}$$

$$\sigma_c = \sqrt{\frac{\sum_{i=1}^{N}(x-\mu)^2}{N}}$$

$$\sigma_c = \sigma \tag{A.20}$$

Appendix B

Detectors’ Performance

Table B.1: Performance of CCWS detector with different tunning parameters.

Parameters			**Performance**		
Preprocessing	Kernel $[f_{min}, f_{max}]$	Detector (N,G,X_{dB},k,T_S,T_L)	Misdetection	False Alarm	F1 score
Marion Island (Training Set)					
None	[1.5,9.5]	(10,5,0.3,13,5,7)	0.092	0.199	0.851
Zscore	[1.5,9.5]	(12,10,0.5,13,5,3)	0.106	0.108	0.893
Zscore	[4.5,11.5]	(15,7,0.7,15,2,7)	0.045	0.174	0.886
C. Detrend	[1.5,9.5]	(8,5,0.4,5,0,7)	0.062	0.159	0.887
L. Detrend	[1.5,9.5]	(12,10,0.4,13,8,5)	0.088	0.124	0.894
Marion Island (Testing Set)					
None	[1.5,9.5]	(10,5,0.3,13,5,7)	0.096	0.199	0.851
Zscore	[1.5,9.5]	(12,10,0.5,13,5,3)	0.096	0.094	0.906
Zscore	[4.5,11.5]	(15,7,0.7,15,2,7)	0.047	0.157	0.886
C. Detrend	[1.5,9.5]	(8,5,0.4,5,0,7)	0.066	0.146	0.892
L. Detrend	[1.5,9.5]	(12,10,0.4,13,8,5)	0.086	0.116	0.899
SANAE IV (Entire dataset)					
None	[1.5,9.5]	(10,5,0.3,13,5,7)	0.91	0.157	0.162
Zscore	[1.5,9.5]	(12,10,0.5,13,5,3)	0.98	0.1	0.04
C. Detrend	[1.5,9.5]	(8,5,0.4,5,0,7)	0.901	0.078	0.178
L. Detrend	[1.5,9.5]	(12,10,0.4,13,8,5)	0.954	0.189	0.086

Table B.2: Performance of the sliding deep convolutional neural network on Marion data.

Parameters	**Classifier**		**Performance**		
Preproessing	Accuracy	Loss	Misdetection	False Alarm	F1 score
Marion Island (Training Set)					
None	0.974	0.087	0.089	0.04	0.935
Zscore	0.977	0.077	0.092	0.036	0.935
C.Detrend	0.979	0.076	0.102	0.03	0.932
L. Detrend	0.97	0.096	0.109	0.025	0.931
Marion Island (Testing Set)					
None	N/A	N/A	0.102	0.036	0.93
Zscore	N/A	N/A	0.093	0.026	0.939
C.Detrend	N/A	N/A	0.112	0.026	0.929
L. Detrend	N/A	N/A	0.113	0.018	0.932
SANAE IV (Entire dataset)					
None	N/A	N/A	0.568	0.07	0.59
Zscore	N/A	N/A	0.532	0.055	0.626
C.Detrend	N/A	N/A	0.494	0.089	0.651
L. Detrend	N/A	N/A	0.574	0.066	0.585

Table B.3: Performance of YOLO on Marion data.

Parameters	**YOLO**	**Performance**		
Preprocessing	Loss	Misdetection	False Alarm	F1 score
Marion Island (Training Set)				
None	0.351	0.085	0.002	0.955
Zscore	0.352	0.075	0.001	0.961
C. Detrend	0.361	0.109	0.002	0.942
L. Detrend	0.364	0.057	0.001	0.97
Marion Island (Testing Set)				
None	N/A	0.159	0.023	0.904
Zscore	N/A	0.147	0.019	0.912
C. Detrend	N/A	0.192	0.013	0.889
L. Detrend	N/A	0.141	0.021	0.915
SANAE IV (Entire dataset)				
None	N/A	0.704	0.082	0.448
Zscore	N/A	0.8	0.047	0.331
C.Detrend	N/A	0.766	0.061	0.375
L. Detrend	N/A	0.815	0.043	0.31

Appendix C

Detectors' Output

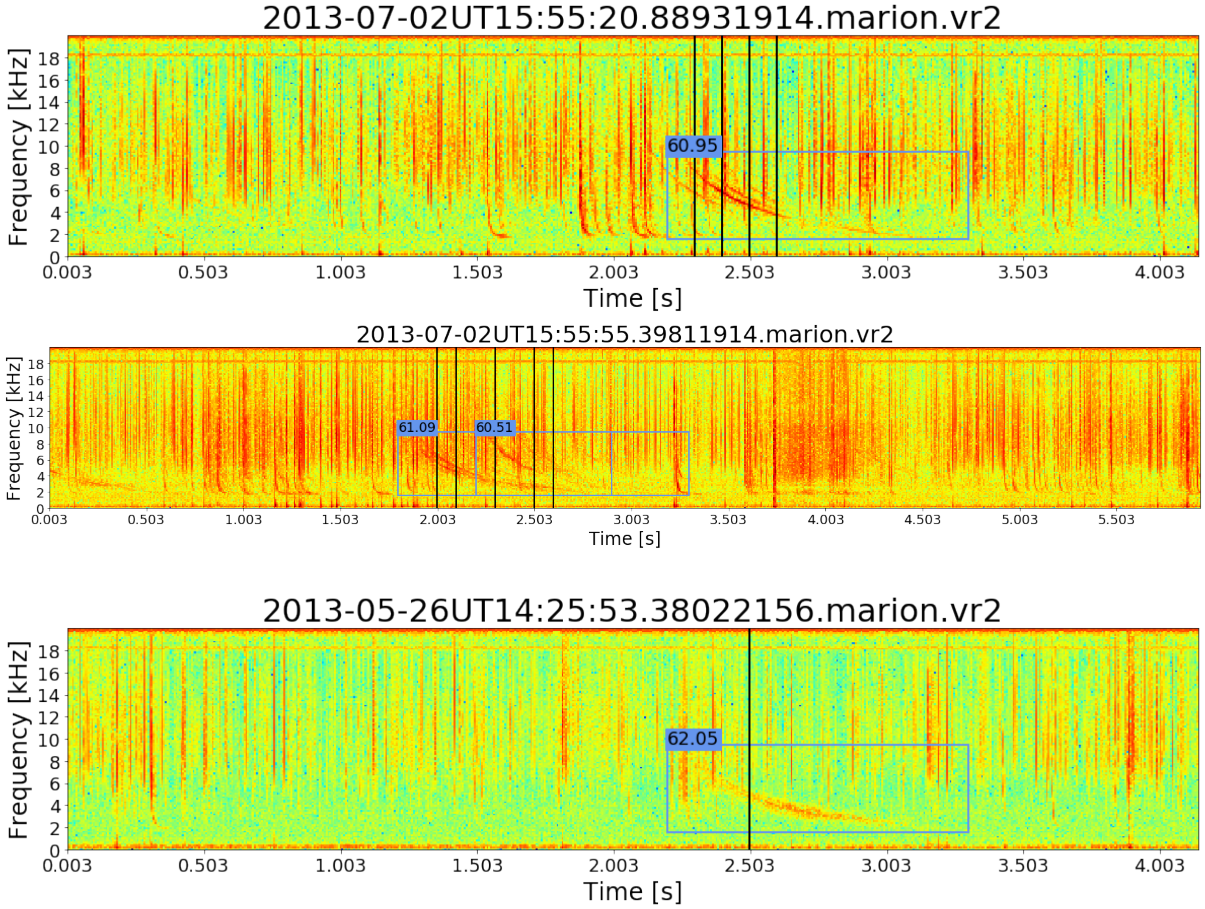


Figure C.1: Detection on a few training samples using CCSW.

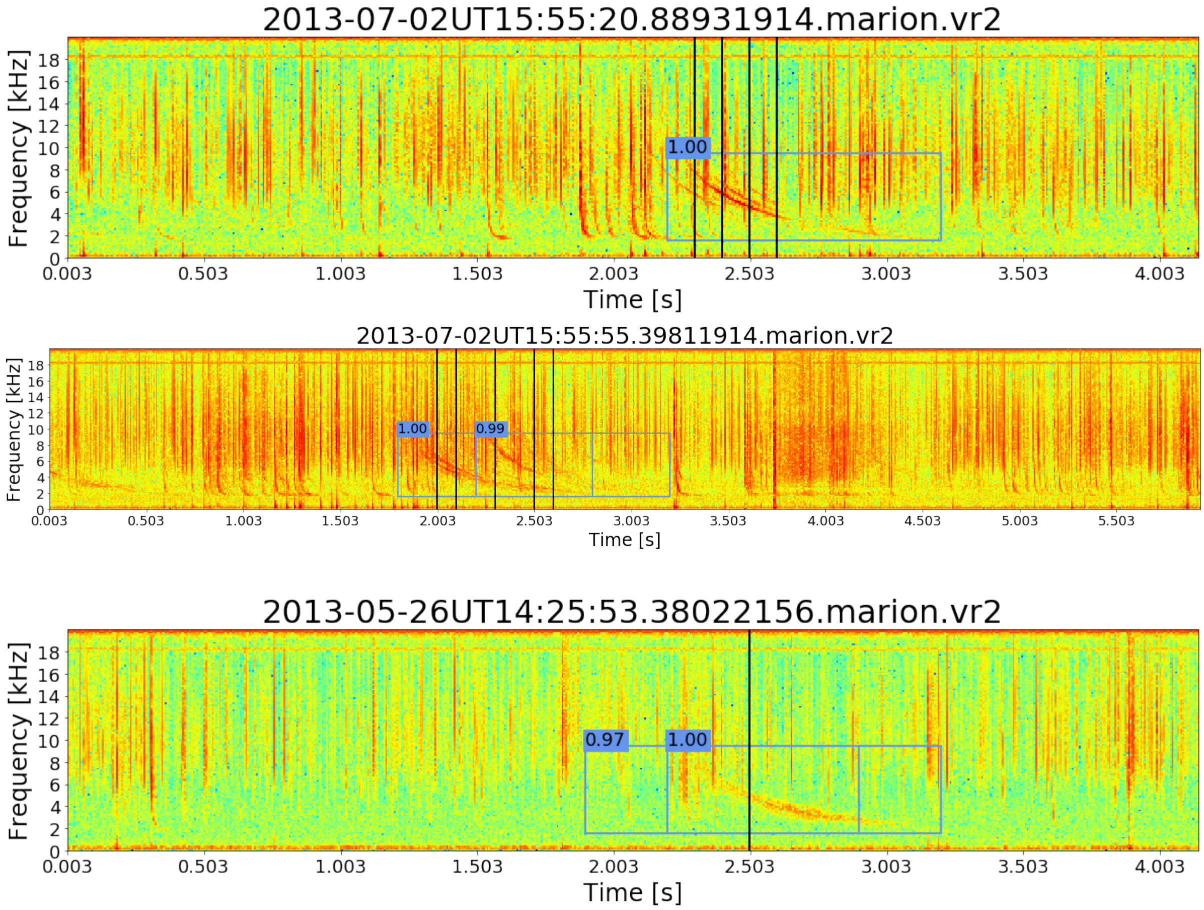


Figure C.2: Detection on a few training samples using SDCNN.

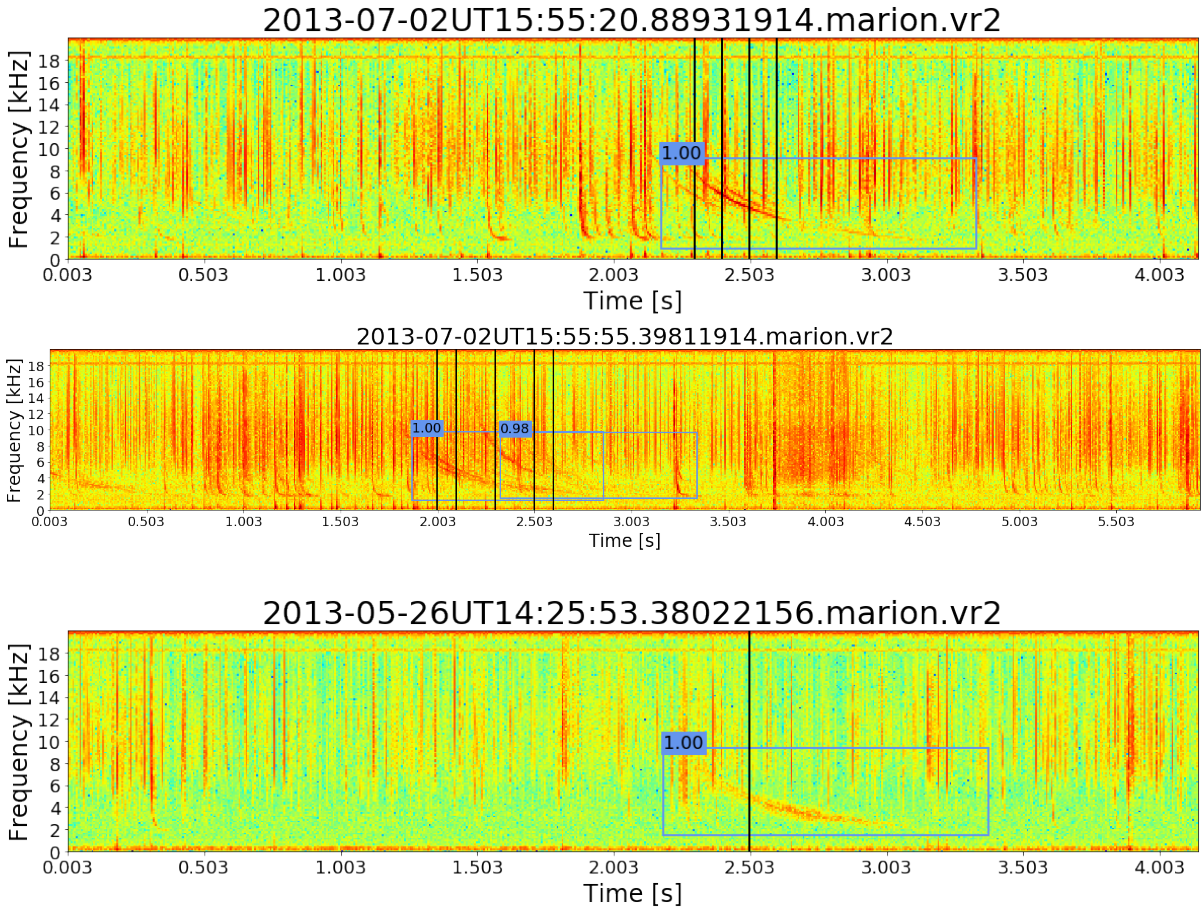
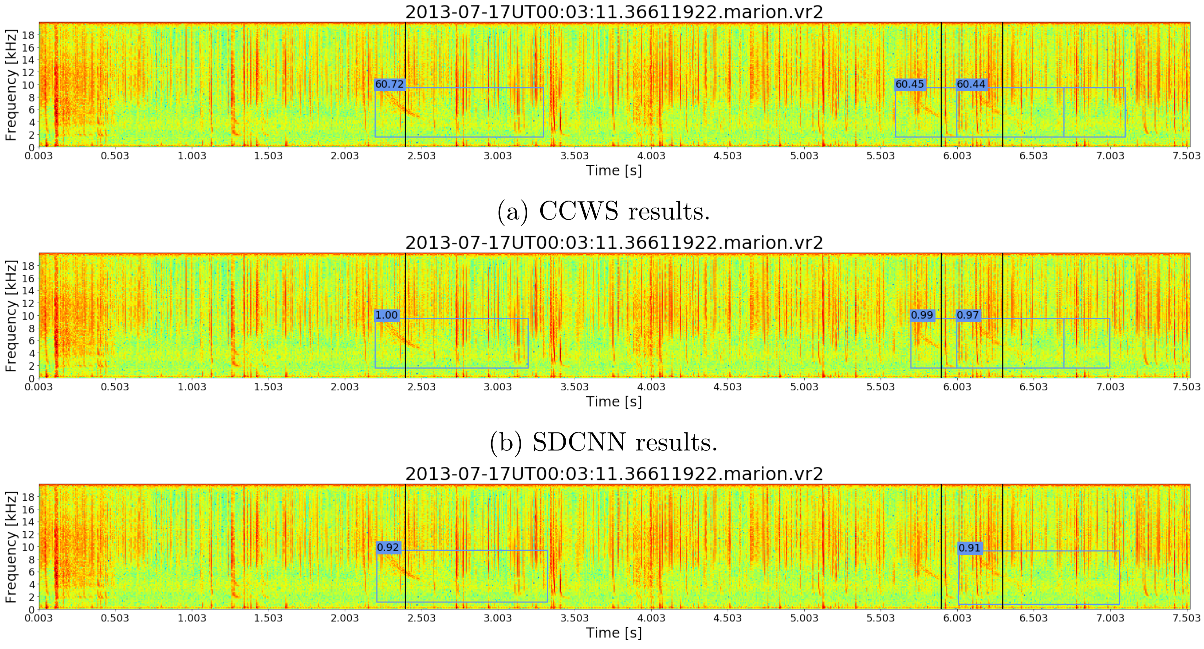


Figure C.3: Detection on a few training samples using YOLO.

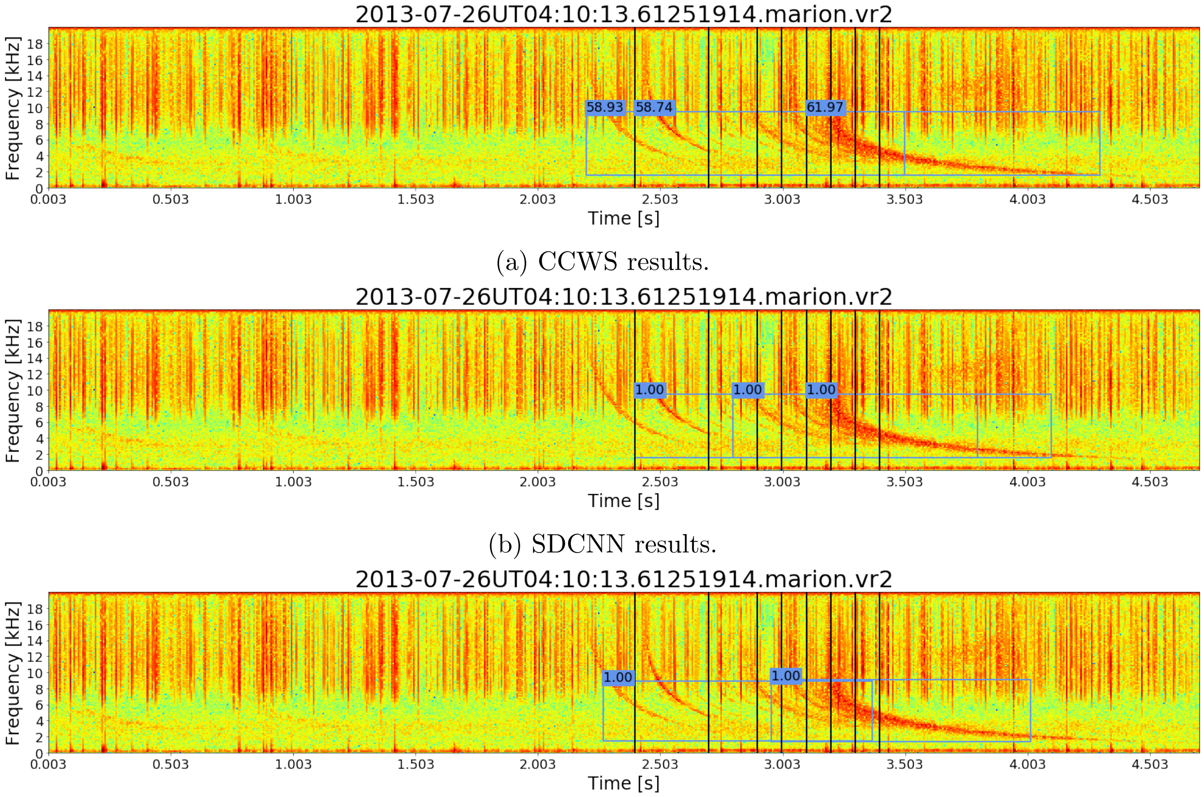


(a) CCWS results.

(b) SDCNN results.

(c) YOLO results.

Figure C.4: Detection on testing sample ”2013-07-17UT00:03:11.36611922.marion.vr2” using all three detectors.

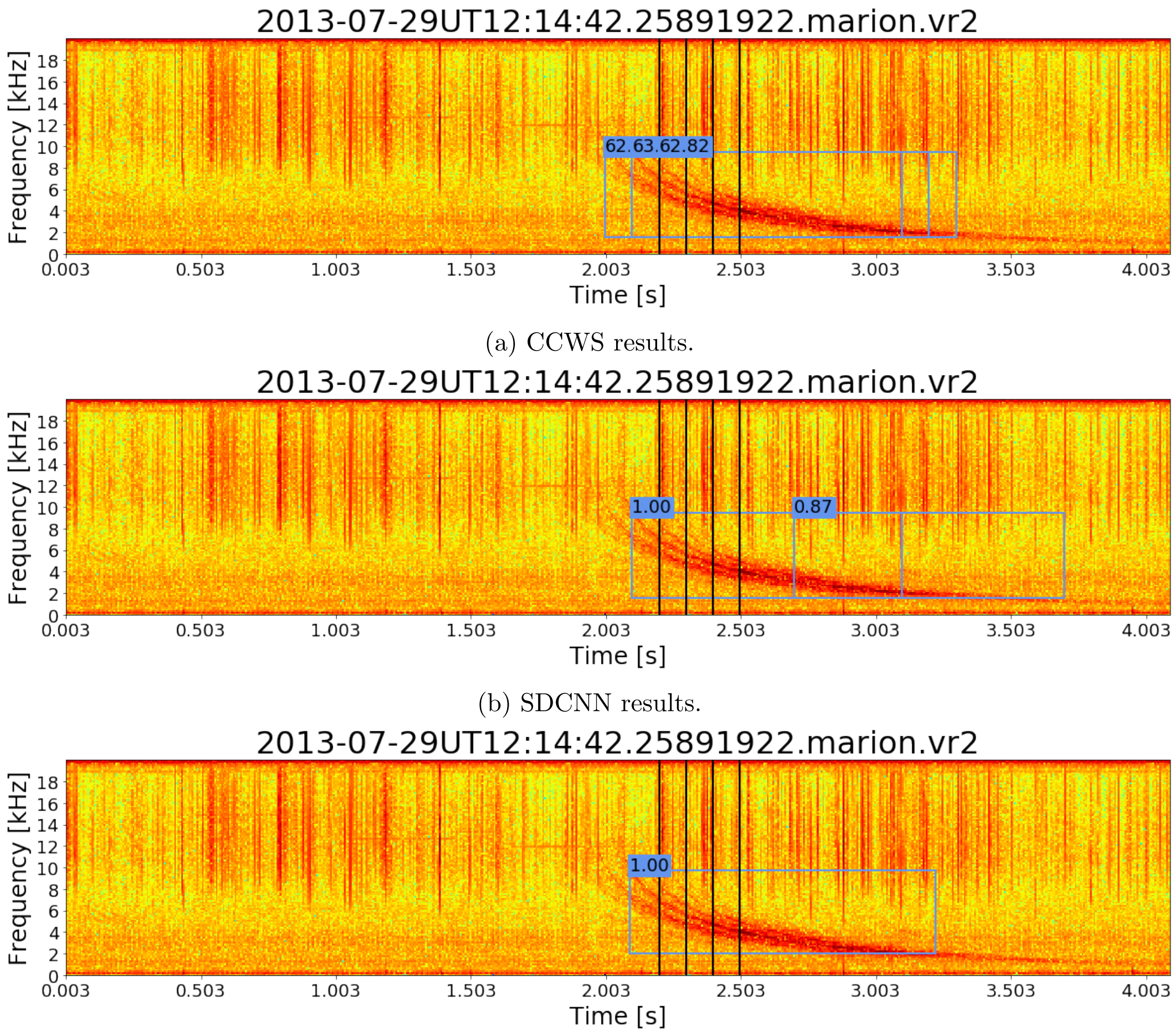


(a) CCWS results.

(b) SDCNN results.

(c) YOLO results.

Figure C.5: Detection on testing sample "2013-07-26UT04:10:13.61251914.marion.vr2" using all three detectors.

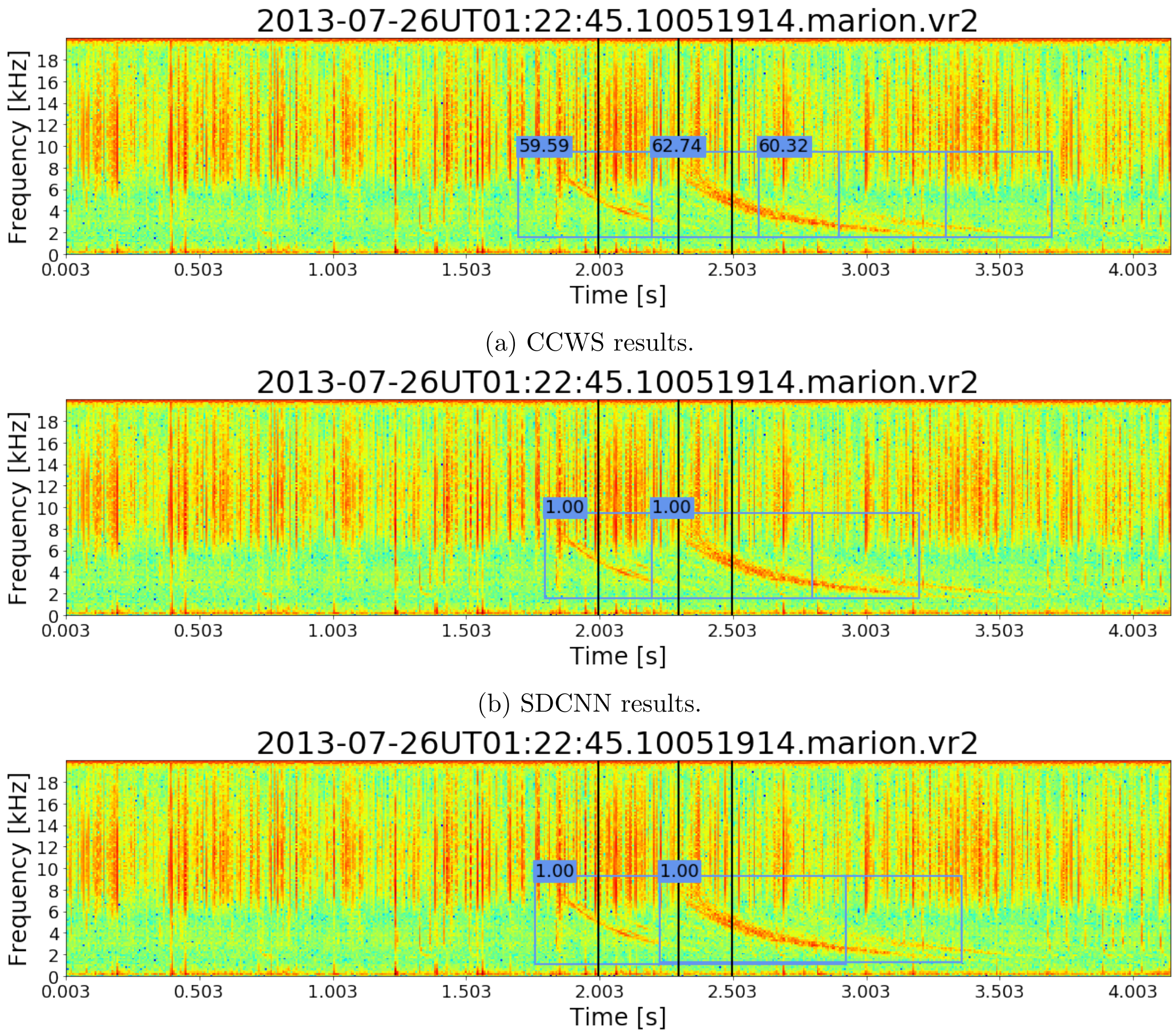


(a) CCWS results.

(b) SDCNN results.

(c) YOLO results.

Figure C.6: Detection of testing sample "2013-07-29UT12:14:42.25891922.marion.vr2" using all three detectors.



(a) CCWS results.

(b) SDCNN results.

(c) YOLO results.

Figure C.7: Detection on testing sample ”2013-07-26UT01:22:45.10051914.marion.vr2” using all three detectors.

Appendix D

Corrections

D.1 Examiner 1

D.1.1 Spectrogram approach is one simple and faster approach for processing time-varying spectra, but as discussed, it has low resolution. Hence why did the author not try other methods like wavelet or Stockwell transform or Wigner Ville transform? (Need not to implement just it is a question for future work)

Not only was it a simple and faster approach, but it is also a standard method of displaying whistlers. The author was aware of the possibility of using wavelets but did not prioritise exploring it. The other two methods mentioned were unknown to the author.

D.1.2 SNR methodology proposed by the author or taken from other sources is not transparent

The SNR for this research had to be created since there wasn't anything in the literature that could be used

D.1.3 Figure 5.11 presents threshold detection method. What is the mathematical background for this approach?

AWD whistler locations are used to collect the cross-correlation power level for whistler and noise. It is assumed that the whistler portion of the cross-correlation also contains noise. Fig 5.11.a shows the probability density function of both noise and target+noise (whistler portion). The Threshold is just the minimum dB value at which the target+interference PDF is higher than the noise PDF. Fig 5.11.b shows the cumulative PDF for both noise (which is Pfa) and target+noise (which is Pd). This figure shows the difference how Pfa and Pd change as a result of the dB value

D.1.4 What are trainable parameters for both networks?

Showed in Section 6.1.2/Classifier performance

D.1.5 Which network has less trainable parameters?

Sliding DCNN

D.1.6 Did the author test the performance networks without pre-processing?

Yes, the results are shown in Tables B.1, B.2, and B.3

D.2 Examiner 2

D.2.1 A fairly crude metric is used to evaluate performance. By accepting any event or events that fall within a detected bounding box as correct, the task becomes significantly easier than if the goal really was to identify the actual bounding box (start time, end time) of whistlers. Even though the postulated AWD bounding boxes are not 'ground truth', they can still be used to match against the detected boxes using measures such as Intersection over Union, and averaged using mAP. The latter is referred to in the context of applying YOLO on the COCO challenge but neither described nor used.

Using IoU or mAP requires the ground truth to be labelled with bounding boxes while our data is not.

D.2.2 Given that we have an estimate of the accuracy of the AWD system, what does this say about the measured results (using the output of the AWD system as proxy)? What is the actual variability (best and worst case) possible? This is worth discussing. Simply stating that the final system achieves a misdetection and false alarm rate of less than 15% is not quite true.

All the results for all cases (with no pre-processing and processing) have been added to Appendix B.

D.2.3 The 'figure of merit' per detection (requirement R11) is never discussed nor analysed, apart from the comment on page 93 with regard to which measurements could be used to provide a figure of merit. While there was indeed no requirement with regard to having a specific confidence in this estimate, some discussion of the results would be useful.

Although not explicitly mentioned in the paper, the software created to implement this requirement is hosted on github https://github.com/Kojey/MSc-whistler-waves-detector

D.2.4 Requirement R12 is somewhat superficially addressed. It is true that there is a start and end time provided, but since the end time is simply start time plus 1 second, actually only a start time is provided, and estimating an end time was left out of the scope of this study.

This is related to the fact that the ground truth does not have bounding boxes labelling, hence it is hard to verify that the estimated end time is correct. However, this was mentioned so that it can be used in the future for data labelled with bounding boxes.

D.2.5 The machine learning background provided (in Sections 2.6 and 2.7) is quite superficial. This is especially noticeable given how well the earlier background is dealt with (Sections 2.1-2.4). While it is fair to include the detail of these methods only in later sections, there is limited context provided with regard to the approaches selected.

There was an assumption that the reader would have the basics of machine learning. Moreover, the field itself is quite immense so only what was deemed essential was covered (perceptrons, then layers of perceptrons, which were the main components of the DCNN), moreover, the last ML algorithm used (YOLO) was also covered in the literature.

D.2.6 I expected to see more references to additional approaches to detecting whistlers, e.g. "Automatic Detection of Lightning Whistlers Observed by the Plasma Wave Experiment Onboard the Arase Satellite Using the OpenCV Library" by Ahmad et al. (2019); "Automated identification of discrete, lightning- generated, multiple- dispersed whistler waves in C/NOFS- VEFI very low frequency observations" by Jacobson et al. (2016), or even minor studies such as "Detection of Whistler Waves" by Jeng (2010). The Jacobson et al. (2016) paper is particularly rich in additional background references.

Indeed, those papers would have been informative, especially the preprocessing method by Ahmad et al. (2019). Unfortunately, the paper by Jacobson et al. (2016) and Jeng (2010) were not hit during the literature and the one by Ahmad (2019) was published when the literature was completed.