Cester, Lucrezia Maria Elisabetta (2022) *Remote measurements of heart valve sounds for health assessment and biometric identification.* PhD thesis.

https://theses.gla.ac.uk/83048/

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses
https://theses.gla.ac.uk/
research-enlighten@glasgow.ac.uk

Remote Measurements of Heart Valve Sounds for Health Assessment and Biometric Identification

Lucrezia Maria Elisabetta Cester

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Physics and Astronomy
University of Glasgow

January 2022

Abstract

Heart failure will contribute to the death of one in three people who read this thesis; and one in three of those who don't.

Although in order to diagnose patients' heart condition cardiologists have access to electrocardiograms, chest X-rays, ultrasound imaging, MRI, Doppler techniques, angiography, and transesophageal echocardiography, these diagnostic techniques require a cardiologist's visit, are expensive, the examination time is long and so are the waiting lists. Furthermore abnormal events might be sporadic and thus constant monitoring would be needed to avoid fatalities.

Therefore in this thesis we propose a cost effective device which can constantly monitor the heart condition based on the principles of phonocardiography, which is a cost-effective method which records heart sounds.

Manual auscultation is not widely used to diagnose because it requires considerable training, it relies on the hearing abilities of the clinician and specificity and sensitivity for manual auscultation are low since results are qualitative and not reproducible. However we propose a cheap laser-based device which is contactless and can constantly monitor patients' heart sounds with a better SNR than the digital stethoscope. We also propose a Machine Learning (ML) aided software trained on data acquired with our device which can classify healthy from unhealthy heart sounds and can perform biometric authentication. This device might allow development of gadgets for remote monitoring of cardiovascular health in different settings.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my supervisor, Dr Daniele Faccio, for inputting a constant stream of ideas full of intuition and foresight. I would also like to thank Dr Ilya Starshynov, who has been collaborating in the experiments reported in this thesis. I would like to express my gratitude towards Dr Ashley Lyons who has been guiding me throughout this journey.

I would also like to thank Miss Yola Jones that has believed in this project from its conception and her supervisors Professor Pierapolo Pellicori and Professor John Cleland for their collaboration in this project.

Declaration

I, Lucrezia M.E. Cester, declare that

- This thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree.
- Except where stated otherwise by reference or acknowledgment, the work presented is entirely my own.
- Works undertaken by others, which has mostly served as literature review, has been properly cited throughout the text.
- Some parts of this work have been previously published in [4, 5].
- The second, third and fourth chapters are a literature review of other works, which are properly cited.
- The fifth, sixth and seventh chapters open with an introductory review of the literature (works are properly cited) and are then followed by my personal contributions.

List of Acronyms and Abbreviations

CAA	Computer Aided Auscultation
ECG	Electrocardiogram
OS	Opening Snap
PCG	Phonocardiogram
CAD	Cardiovascular Artery Disease
Na	Sodium
Ca	Calcium
K	Potassium
Hz	Hertz
SEM	Systolic Ejection Murmur
CAD	Coronary Artery Disease
FT	Fourier Transform
t-SNE	t-Distributed Stochastic Neighbor Embedding
SVM	Support Vector Machine
WT	Wavelet Transform
CNN	Convolutional Neural Network
NN	Neural Network
RNN	Recurrent Neural Network
DNN	Deep Neural Network
LSTM	Long Short Term Memory
CMOS	Complementary Metal Oxide Semiconductor
ADC	Analog to Digital Converter
SPAD	Semiconductor-based single-photon avalanche diode
LDV	Laser Doppler Vibrometry
BS	Beam Splitter
QWP	Analog to Digital Converter
PBS	Polarising Beam Splitter
NCC	Normaised Cross Correlation
SNR	Signal to Noise Ratio
W	Watts

CW	Continuous Wave
PF	Photon Force
DSC	Depth Sulci Cutis
DCC	Depth Cristae Cutis
PTFE	Polytetrafluoroethylene
SPECT	Single Photon Emission Computed Tomography
PET	Photon Emission Tomography
MRI	Magnetic Resonance Imaging
ESC	European Society of Cardiology
TV	Total Variation
EMD	Empirical Mode Decomposition
IMF	IntrinsicMode Functions
MFCC	Mel Frequency Cepstrum Coefficients
DCT	Discrete Cosine Transform

Chapter 1

Introduction

The Topol Review published in 2019 reported that genomics, digital medicine, AI and robotics will become an integral part of the healthcare system in the next two decades [1]. In this regard, this thesis aims to propose a digital health solution which is based on digital medicine and AI.

Strictly speaking, digital medicine uses software and algorithmically driven products to diagnose and treat diseases [2]. The field of digital medicine was born around 2007 when the IPhone first came out. The connectivity of mobile devices with the internet opened the doors to the individual generating health data on their own and in a real-world environment. The field of biosensors opened with apps that would calculate the number of steps a person took in a day. Then many new devices which can acquire different health parameters appeared. These innovations are leading healthcare in the hands of the individual in a non-hospital setting [3].

The term “Artificial Intelligence” was first coined by John McCarthy as “the science and engineering of making intelligent machines” [6]. Machine Learning is a branch of artificial intelligence in which the machine learns from data and makes decisions on its own without having been specifically programmed [7]. These "intelligent" computational algorithms, which can learn from data and then apply their knowledge to new unseen data, are opening new doors to the automated delivery of health care by computers which can perform quantitative analysis on patients’ data and derive insights to diagnose, manage and treat a wide variety of medical conditions [8]. The most important advantages of using machine learning algorithms in the healthcare domain are to learn more complicated and subtle patterns that a human could not discern, thus enabling better outcomes for patients, and also to fill a gap of healthcare assistance which doesn’t fulfil everyone’s needs, especially in developing countries but also in developed countries [9].

In recent years a new interest has arisen for the field of computer aided auscultation (CAA) of the heart sounds. The reason for this renovated interest in the field was the technological progression of new sensing devices and the advent of artificial intelligence software as a medical device. According to the World Health Organisation, heart diseases are the leading cause of death worldwide. Cardiovascular diseases can be hard to diagnose because often the patients are asymptomatic [10]. Constant thorough examinations of the world population, which would consist in the use of complex medical devices such as MRIs, PETS, ECGs and the expertise of medical staff, is not feasible. This is where machine learning comes into the picture.

This thesis presents a new sensing device and complimentary software which can acquire contactlessly and from a distance heart valve sounds, identify the person it is monitoring and diagnose whether or not the person is in need of further medical attention. Such a device would be most useful for the future "smart homes" or in areas where the ratio patients to doctors is very high.

This was the first time, to the best of our knowledge, that heart valve sounds and cardiovascular sounds were acquired from a non-chest location with a better SNR than the digital stethoscope. The goal of this work was to show it is possible to diagnose heart diseases remotely and contactlessly. This has been achieved and the results are shown in Chapter 6. This was, in my opinion, the most remarkable milestone achieved in this work, namely, we show that, for the first time, it has been possible to diagnose reduced ejection fraction without touching the patient, from 1 meter distance and in just a few seconds of data acquisition.

This opens the door to various possibilities, such as primary care early diagnosis of heart failure. In addition, we have also shown that with just 4 minutes of training data from each subject, it was possible to build an AI algorithm that would recognize the different subjects from their heart sounds. This paves the way for remote in-home or care home constant monitoring of people's heart status, which is very important since death by cardiovascular diseases is the leading cause of death worldwide.

This work also shows that it is possible to retrieve a high frequency range of heart sounds from the neck, where the stethoscope can't and that this frequency range is important for biometric authentication and heart health diagnosis.

In the process of testing the device, we have shown that sound retrieval from light goes beyond the memory effect range and thus can be performed when the light needs to go through scattering material before reaching the detector. We have also simulated the results so to give an explanation as to why that is the case.

This thesis is structured as follows.

Chapter 2 is an introduction of the anatomy of the cardiovascular system and the various illnesses that can affect it, specifically, how these illnesses change the normal sounds the heart makes. It will also mention the current methods to identify heart diseases and what is the role of diagnosis through heart sound.

Chapter 3 will discuss the working principles of various computational algorithms which will then be used for reducing the dimensionality of the data which will then be fed to machine learning algorithms for the automatic classification of heart diseases and people.

Chapter 4 will provide an introduction to machine learning with specific emphasis on algorithms which we will use in our work to classify heart health and perform people classification.

Chapter 5 will introduce sensing technologies and various techniques to acquire heart valve sounds contactlessly and from a distance and then it will contain the description of our devised method. It will talk about the experiments and simulations that we carried out to test the validity of our device which will serve to automatically acquire heart valve sounds contactlessly and from a distance.

Chapter 6 will delve into our experiments. First we will show that, for the first time to the best of our knowledge, with our device we managed to retrieve heart valve sounds from people's necks which lie in a higher frequency range and with better SNR that what can be acquired with existing methods. We will also show how the new frequency range that we can retrieve contains useful information for diagnosing heart conditions. Then we show that the algorithm we built managed to classify healthy from unhealthy individuals with the heart sounds collected with our device from 10 subjects.

Finally in Chapter 7 there is a brief introduction on various biometric authentication methods and their pitfalls. Then we will show that our devised algorithm can classify correctly the subjects from their heart valve sounds acquired with our device. Finally we will show that for the purpose of biometric authentication the high frequency range of the heart valve sounds provides better authentication accuracy.

Chapter 2

Heart Sound: A Review

A brief overview of the physiology and anatomy of the heart muscle will enable us to understand all the different diseases which can affect the heart and how they impact its normal functioning. Since death by heart diseases is the leading cause of death worldwide, studying this organ and its pitfalls is necessary to develop new technologies to monitor and assess the heart status and prevent these deaths. In fact, heart diseases, if detected in time, can be managed with appropriate medications and lifestyle changes [11].

This chapter will talk about cardiovascular diseases, how they can be detected and how our devised method fares compared to other diagnostic techniques. Then it will explore the anatomy of the heart, delving into its electrical and mechanical activity. We will see that many of the heart malfunctioning manifest in mechanical changes of blood flow. This thesis will be focused on how to detect these changes and make use of them to detect heart diseases and perform biometric authentication.

2.1 Heart Health Assessment Methods

Cardiovascular diseases (CVDs), are a group of heart disorders that affect the structure and function of the heart. These disorders, according to the World Health Organization, are the leading cause of death in developed countries and one of the leading causes in developing countries, taking an estimate of 17.9 million lives each year [12, 13]. It is possible to not let these conditions become critical by detecting them in time so that doctors can prescribe lifestyle changes and pharmacological prevention. However these conditions often do not present symptoms until it's too late so and there is no method to identify those whose condition is already or is becoming pathological. Even risk factors, such as age and sex, do not help in identifying people at risk [14]. A way to identify those who need medical attention would be a technology which is cheap, non-invasive and widely spread and that can monitor the heart condition constantly.

In order to monitor the heart condition one would need to extract parameters which can be used for diagnosis. Currently, there are various methods to diagnose problems with the heart. The cheapest method, the ECG, is used to check the electrical activity; this tool can diagnose:

- Arrhythmias. These are conditions in which the heart beats too slowly, too fast or irregularly. We have seen in the second chapter that the ECG registers the electrical activity for each heart beat, this can show the irregularities of the rhythm [15].
- Coronary heart disease. In these conditions the heart's blood supply is blocked by a build-up of fat, which means that part of the heart might not be getting enough oxygen. Thus ECG waveforms will be different from normal [16].
- Heart attacks. Conditions in which the flow of blood to the heart is blocked. Because of these some heart cells might be dead and the abnormal shape of the ECG waveforms will detect this [17].
- Cardiomyopathy. Condition in which the heart walls become thickened or enlarged and thus may not be pumping efficiently and potentially even block blood flow [18].

The ECG has many advantages: it is a cheap medical tool, it is non-invasive, and easy to use. However, it also has some limitations: it presents difficulties in detecting structural abnormalities in heart valves and does not detect heart murmurs.

Another diagnostic technique are X-rays. X-rays are used to form an image of the heart which allows to check:

- Enlargement of the heart.
- Calcium build up, which is an indication of a heart attack being possible to happen.
- Congenital heart defects (conditions one is born with and which affect the structure of the heart) [19].

Another commonly used technique is the echocardiogram. An echocardiogram is done to check if the chambers and the valves of the heart are working properly [20]. This instrument is said to be the most accurate in diagnosing heart failure because it checks diastolic and systolic dysfunctions [21].

It is then possible to diagnose heart problems through blood tests. Blood tests can be used as a diagnostic tool to check for an imbalance of specific proteins in the blood [26].
Another technique allows diagnosis through the use of radiation; this technique is nuclear heart scans. Nuclear heart scans, which use single photon emission computed tomography (SPECT) or cardiac positron emission tomography (PET), are used check:

- The flow of blood.
- Damaged or dead heart muscle tissue.
- How well the heart pumps (Ejection Fraction) [27].

The last technique consists in diagnosing through MRI. Heart MRI scans help diagnose pericardial diseases (diseases affecting the outside lining of the heart), heart tumors, congenital heart disease, cardiomyopathy, heart valve disease, and whether the heart is pumping correctly without the additive risks associated with using radiation as in nuclear scans [28].

Symptoms of heart failure can be very subtle followed by sudden death. Therefore early detection of heart failure is crucial given that modern drug treatments help with symptoms and quality of life, slow down the rate of disease progression, and improve survival [30]. The problem with the current methods of heart health assessment is that these technologies are very expensive, in fact according to the World Health Organization [31], nearly 80% of deaths due to cardiovascular disease occur in low- and middle-income countries where this equipment is not available. But even in first world countries the rate of screenings and tests are not enough to detect and prevent Heart Failure, especially because death comes before symptoms that might warrant a visit to the hospital. Thus in this thesis we propose a cheap method that would allow to constantly monitor the heart condition through heart valve sounds so that diseases can be detected early. In order to understand why our device can diagnose heart sounds and why these sounds can be used to train machine learning algorithms to diagnose heart conditions and perform biometric authentication and classification, one needs to understand how heart sounds are produced by the heart. This will be the focus of the last part of this chapter.

2.2 Role of the Heart Sounds in Heart Health Assessment

According to the ESC (European Society of Cardiology) guidelines, heart failure is defined as 'a complex clinical syndrome that can result from any structural or functional cardiac disorder that impairs the ability of ventricles to fill with or eject blood ' [21]. We have seen methods to check for heart failure by looking at images of the pumping heart or by checking its electrical activity. But it is also possible to check the status of the ventricles through the sound they make. In fact there is a vast range of diseases that can be detected by heart sound auscultation. However diagnosis through auscultation is not widely carried out. The reason for this is that diagnosis through heart valve sounds requires considerable training, it relies on the hearing abilities of the clinician, thus making it a very subjective, and specificity and sensitivity for manual auscultation are low since results are qualitative and not reproducible. Furthermore, sounds recovered with the acoustic stethoscope are compromised with noise. The electronic stethoscope has filters which remove the noise, however studies which compared the electronic stethoscopes with standard devices have concluded that acoustic stethoscopes give more accurate diagnosis [22, 23]. Another problem with the stethoscope is that due to differences in construction materials, each stethoscope has a different response to specific frequency ranges, which is a problem since different frequency ranges are associated with different diseases [24]. In order to collect heart sounds, the stethoscope is placed by the clinicians on the chest of the patient and is considered to be an 'art 'with considerable training required in order to distinguish subtle sounds and murmurs but also to learn how to apply pressure on the stethoscope as it can affect the quality of the sound retrieved. Furthermore, learning how to diagnose diseases by auscultation requires considerable training and the diagnosis are very subjective [25].

However since the emergence of electronic stethoscopes a new interest has developed towards a new field called 'computer-aided auscultation '. This field has flourished thanks to the recent developments of sound acquisition sensors, advancements in digital signal processing and machine learning techniques. Acoustic based automatic diagnosis of cardiac dysfunctions has attracted much attention in recent years.

We built a device and complimentary software which can detect heart valve sounds contactlessly and from a distance. This device presents some advantages compared to the acoustic and electronic stethoscopes: since it is sensitive to nanometer-wide moevements, it can acquire heart sounds from peripheral blood vessels, thus it can be placed in peripheral locations where the sound is not corrupted by breathing noise. Its sensitivity also allows it to capture the high frequency range of the heart sound. It also a single response to frequencies. Because it is contactless, it can acquire heart sounds remotely, which would enable contactless and therefore potentially more precise phonocardiogram measurements but could also be a key component for future intelligent ambients with remote continuous monitoring of health in the home or hospital [32] thus allowing to capture sporadic yet potentially dangerous events that can signal early onsets of heart failure.

Heart sounds are generated by the closing of the valves between ventricles and big arteries and closing of the valves between atria and ventricles, blood flowing and chambers expanding. Since heart failure comprises all those conditions which affect the ability of the ventricles to fill and eject blood, there are many diseases that can be detected by checking if the sounds the valves make are abnormal. In this thesis we will show a list of all the diseases that can be diagnosed through heart sounds and schematic phonocardiograms of these sounds to show the timing and frequency content that these abnormal sounds make. The frequencies that compose the heart sounds lie in the low frequency range. In more detail, heart sound signals consist of a first heart sound (S1 - with a tone that can range between 50 and 150 Hz) separated by the systolic pause from the second heart sound (S2 - with a tone that can range between 80 and 200 Hz) [33]. Other extra heart sounds such as S3, can be either normal or sign of a disease, while all other extra sounds and high frequency murmurs, if present, are signs of cardiac disease and can range between 20 to 1000 Hz [34]. These extra abnormal sounds will contain different frequency ranges and happen at different timing depending on the pathology affecting the cardiovascular system.

The reason why constant, cheap and non invasive monitoring of the heart sounds is important is because many diseases of the heart cause changes in heart sounds before other symptoms appear [29]. This is the reason why the device will be very effective if used in people's homes where it can continuously monitor people's heart's status, thus enabling the detection of sporadic yet potentially fatal events and monitor conditions which develop over time. In order to be used for early diagnosis in people's homes as people go on in their daily activities, it is desirable that the device can also automatically identify the person it is monitoring (biometric authentication). How this is done will be shown in the last chapter of this thesis. This thesis will present a laser-based PCG system, which allows for contactless, continuous and remote detection of heart sounds for cardiovascular disease detection and people authentication and classification.

2.3 The Circulatory System

The cardiovascular system includes blood, blood vessels, the heart, and the lymphatic system. The circulatory system contributes to the homeostatis of the body by delivering oxygen and nutrients to the cells and removing waste. It also allows for the transport of heat, produced by the work of the muscles, to the skin. It also transports hormones to where they are needed [35].

2.3.1 Blood

The blood has three functions:

- Transportation of carbon dioxide, oxygen, nutrients, waste, hormones etc.
- Regulation of pH, temperature, osmotic pressure.
- Protection against diseases and clotting to prevent blood losses from cuts.

The blood is composed of red blood cells, white blood cells and platelets. Red blood cells' role is to carry oxygen. This is possible through hemoglobin, a protein which binds oxygen. The white blood cells' function is to protect against infections and cancer. The platelets function is to to form blood clotting to prevent loss of blood when an injury occurs [36].

2.3.2 Blood Vessels

In the blood vessels the blood moves in bulk, which means all the components move together in the same direction. The blood manages to reach every organ because the venous pressure is kept lower than the arterial pressure by the pumping of the heart. The blood is pumped out of the heart through the arteries and comes back to the heart through the veins. The systemic circulation refers to the left heart pump which ejects blood from the left ventricle via the aorta. All arteries of the systemic circulation branch from the aorta. The aorta then branches in successively smaller vessels until the blood reaches the capillaries from where it exits and it begins its return to the heart via the venules which then coalesce into the veins from which blood returns to the right heart pump, specifically into the right atrium. In many veins, especially those in the limbs, given they need to carry blood upwards, there are valves which make sure the blood flows in the right direction and prevent backflow.

Blood vessels' inner and outer walls have unequal pressures which supplies the driving force for flow. Because friction develops between moving blood and the stationary vessels' walls, the resistence (how difficult it is to create blood flow through a vessel) is given by the Equation:

$$Q = \frac{\Delta P}{R} \tag{2.1}$$

where Q is the flow rate (volume/time), ΔP the pressure difference (mmHg), and R the resistance to flow $(mmHg * time/volume)$. While the resistance to flow is given by

$$R = \frac{8L\eta}{\pi r^4} \tag{2.2}$$

where r is the inside radius of the vessel, L the vessel length, and η the blood viscosity [35]. Blood flow only occurs when there is a pressure difference. It is regulated through the radii of vessels and a very important function of the heart is to keep pressure within arteries higher than than in the veins. The average pressure in systemic arteries is approximately 100 mmHg, while in the veins it decreases to near 0 mmHg in the great caval veins.

The velocity of blood flow is inversely related to the vascular cross-sectional area, such that velocity is slowest where the total cross-sectional area is largest [36].

2.3.3 Heart

The heart is a muscular pump whose purpose is to collect blood from the tissues of the body and pump it to the lungs and collect blood from the lungs and pump it to all the tissues of the body. The heart functions as a pump, it imparts pressure to the blood so that it flows [35].

Atria and Ventricles

The heart is composed of two upper chambers, called atria, whose function it to collect the blood, while two lower chambers, called ventricles, are much stronger and function to pump blood. The right atrium and ventricle collect blood from the body and pump it to the lungs, and the left atrium and ventricle collect blood from the lungs and pump it throughout the body. The blood flows in one direction and this one directional flow is maintained and regulated by a set of four valves: tricuspid, bicuspid, pulmonary, and aortic. The atrioventricular valves (tricuspid and bicuspid) allow blood to flow only from atria to ventricles. The semilunar valves (pulmonary and aortic) allow blood to flow only from the ventricles to the great arteries [37]. As shown in Figure 2.1, the venous blood returns to the right atrium through the venae cavae. Then it goes through the tricuspid valve into the right ventricles. Then it is pumped into the pulmonary artery. After passing through the pulmonary capillary beds, the oxygenated pulmonary venous blood returns to the left atrium through the pulmonary veins. The flow of blood then passes through the mitral valve into the left ventricle and is pumped through the aortic valve into the aorta [38].

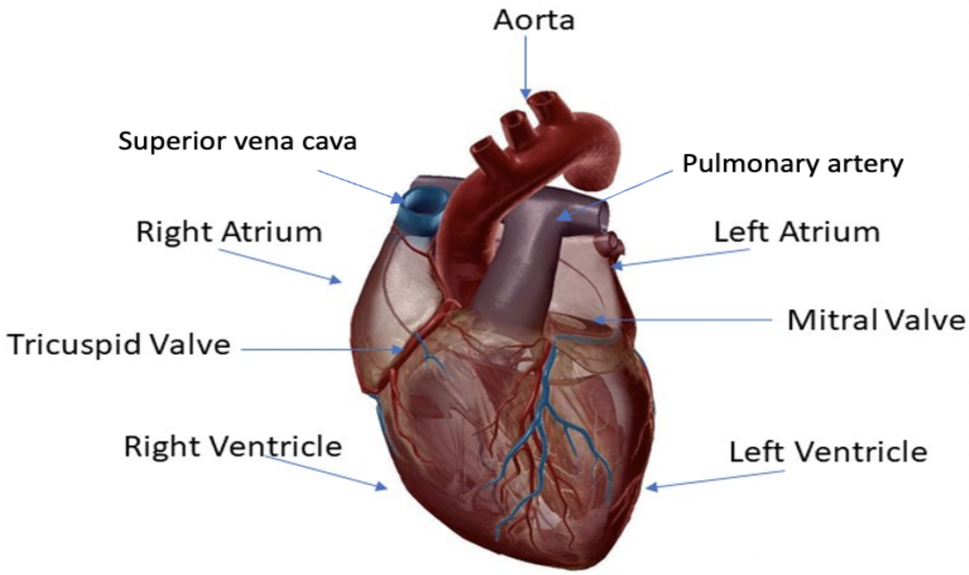


Figure 2.1: shows a schematic representation of the anatomy of the heart. The de-oxygenated blood enters through the inferior vena cava into the right ventricle, then passes through the tricuspid valve and enters the right ventricle. From there it is pumped into the lungs through the pulmonary artery. Then it returns to the left atrium, it passes through the mitral valve and reaches the left ventricle and finally it is carried through the body by the aorta. Figure made through the 3D4Medical software.

The ventricles are chambers-like structures, and the valves are structurally designed to allow flow in only one direction which open and close in response to pressure. The atria receive blood returning to the heart, while the ventricles receive blood from the atria. The right atrium receives deoxygenated blood from the body through the superior and inferior vena cava. The right ventricule pumps the blood into the lungs where it is oxygenated. The oxygenated blood is returned to the left atrium through the pulmonary veins and then it goes into the left ventricule through the cardiac valves. The left ventrivule then pumps the body to the aorta from which it is delivered to the whole body.

When the ventricles contract, the tension generated causes the pressure within the chamber to increase. This causes the pressure in the ventricle to exceed the pressure in the pulmonary artery and aorta thus the blood is forced out of the given ventricular chamber given that blood travels from higher to lower pressure. This active contractile phase of the cardiac cycle is known as systole. During the systole, because the pressure is higher in the ventricle than in the atria, the tricuspid and mitral valves are closed. When the ventricle relax and the pressure in the ventricles falls below that in the atria, the valves open; the ventricles refill and this phase is known as diastole. The aortic and pulmonary (semilunar or outlet) valves are closed during diastole because the pressure inside the ventricles is lower than the pressure of the arteries.

Cardiac Valves

As already mentioned, the cardiac valves purpose is to make the blood flow in the appropriate direction. The right atria and right ventricle are separated by the tricuspid valve from which the deoxygenated blood flows through. After blood passes from the right atria to the right ventricule, it goes through the pulmonary valve which is situated between the right ventricule and the pulmonary artery. So at this point the deoxygenated blood reaches the lungs. From here, the oxygenated blood from the lungs goes back to the left atrium from the pulmonary vein. From the left atrium blood flows to the left ventricule through the mitral valve. It then passes through the aortic valve into the aorta, which transports oxygenated blood throughout the body.

The phase of the heart when the chambers contract is called 'systole 'and the relaxation phase, during which the heart fills up with blood is called 'diastole '. This is called a "cardiac cycle".

- Atrial systole and ventricular filling:
  At this part of the cardiac cycle, the pressure in the heart is low and the blood fills the atria on both sides. Then the atrioventricular valves open and blood flows into the ventricles.

- Ventricular systole:
  Now the atria relax and the tricuspid valve and mitral (which are located between the atria and corresponding ventricle) close. As the atria relax, the ventricles begin to contract. This increases the pressure within the cavity which begins to exceed the pressure within the arteries thus forcing the opening of the aortic and pulmonary valves and the blood then flows from ventricles and into these vessels.

- Isovolumetric relaxation:
  At this point the ventricles relax therefore the ventricular pressure drops thus the blood in the vessels momentarily backflows while aortic and pulmonary valves close [39].

2.4 Electrical Activity of the Heart

The heart is comprised of two types of cells:

- Contractile cells:
  which make up 99 % of the total cells of the heart and are responsible for the mechanical work of pumping.

- Autorhythmic cells:
  which are responsible for conducting the action potential that leads to the contraction of the contractile cells. They do so by sending an electric impulse through the heart.

The membrane potential is due to the difference of concentration of ions between the membrane sides. By convention the polarity (positive or negative) of the membrane potential is stated in terms of the sign of the excess charge on the inside of the cell. The membrane is a concentration of a double layer of lipids that isolates the cell. It has a circular form with an inner side and an outer side. Inside the membrane there are proteins which cross it from side to side (transmembrane proteins) which allow the passage of ions (which carry an electrical charge). The membrane has a membrane potential which can be calculated with the Nernst equation

$$E_{ion} = \frac{61}{Z} * log\frac{C_o}{C_1} \tag{2.3}$$

E_{ion} is the equilibrium potential for a single ion kind in mV. Z is the ion valence (number of electrons in the outermost shell of the atom). C_0 is the concentration of the ions outside the cell in millimoles over liters. C_1 is the concentration of the ion inside the cell calculated in millimoles over liters. This is the basic membrane potential of the heart ventricle cells which are called authorythmic cells. From this standard membrane potential, the authorythmic cells undergo depolarization which is caused by increased Na+ and Ca+ ions entering the cell and an outward K+ current. So the cell membrane potential slowly decreases from -60 mV to -40 mV. This is a critical level, known as threshold potential. At this point there is an action potential, now the inside of the cell becomes positive because L-type channels open and Ca^{2+} cells flow in and the outside becomes negative. So the membrane potential becomes 2 mV. An action potential is a depolarization which propagates along the authorythmic cells creating an electrical activity of the heart. The electrical activity controls the mechanical activity of the heart. The authorythmic cells generate their own action potential as just explained, while the contractive cells stay at a membrane potential of -90 mV until they receive the electrical activity of the authorythmic cells, which generates an action potential in them [35].

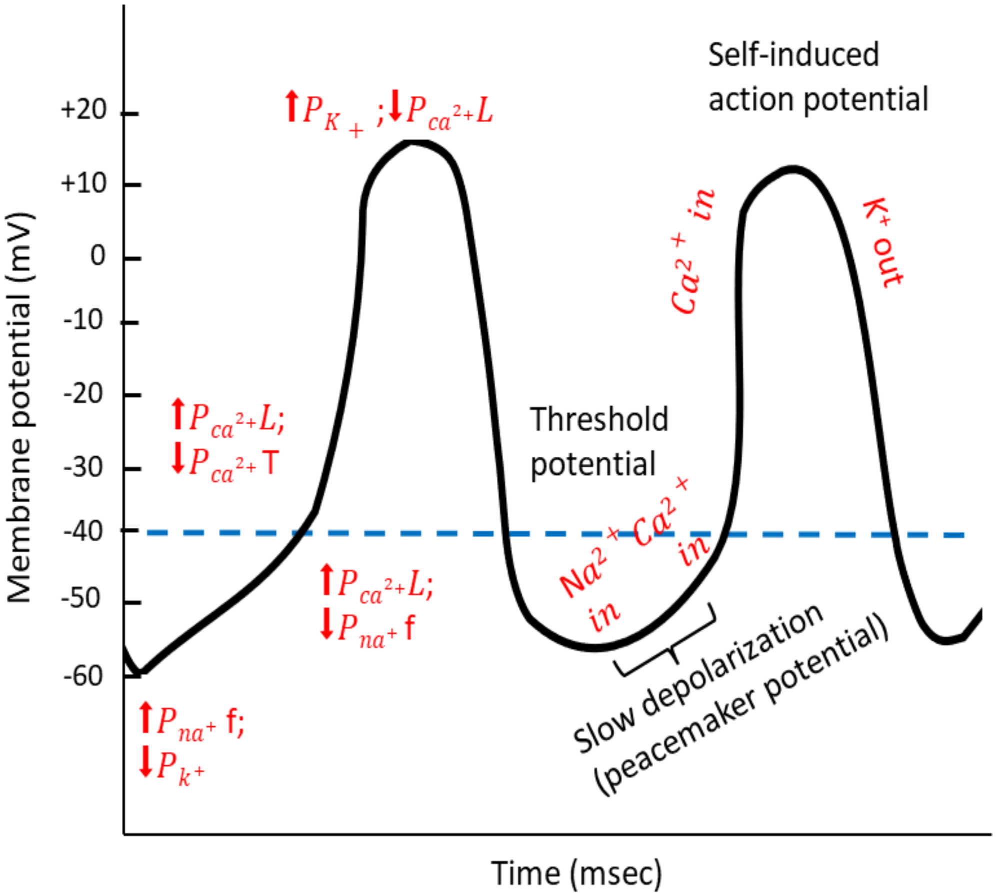


Figure 2.2: shows the action potential of the autorhythmic cells. At first the positively charged Na+ and T-type Ca^{2+} ions travel inside the cell membrane, increasing the membrane potential from -60 mV to -40 mV. This is the threshold level, L-type Ca^{2+} channels open and these positively charged ions flow inside the cell during the action potential phase. While in the falling phase the K+ ions channels are open and they travel outside the cell.

The action potential generates a muscle twitch or contractile response of the cells, which is what is seen macroscopically when the heart contracts. The electrical activity just described is conducted by the body fluids and a small part reaches the skin where it can be recorded and this is what constitutes ECG graphs. An example of what an ECG looks like is shown in Figure 2.5, where the green line shows the ECG time trace. The P-wave represents the atrial depolarization. The QRS wave signals the ventricular depolarization and the T wave represents the ventricular repolarization.

A problem with the electrical activity of the heart means a problem with its mechanical activity. Getting the ECG from patients means being able to check abnormalities in heart rate, abnormalities in heart rhythm and myopathies. Abnormalities in cardiac heart rate can be checked by checking the distance between two consecutive QRS waves, the total number of beats per minute also gives an indication of heart health. Abnormalities in rhythm refers to a spacing between consecutive QRS waves which differs in length. Cardiac Myopathies signal a damaged heart muscle and appear on the ECG when part of the heart muscle becomes necrotic.

As we have seen, the electrical stimulus induces the heart to contract. The mechanical movement of the heart during one cardiac cycle (one full heart beating) consists of the heart contracting and emptying the blood into the body and relaxing and refilling. During a cycle the heart goes through systole and diastole, more precisely its motion can be divided in

- Mid ventricular diastole:
  this moment in time of the mechanical cycle of the heart corresponds to the moment in time when the TP segment of the ECG takes place. This moment in time, from Figure 2.2 corresponds to the interval after ventricular repolarization and before atrial depolarization. Repolarization means that the ions return to their resting state where the membrane is at -60 mV, which corresponds with relaxation of the myocardial muscle.

- Late ventricular diastole:
  this is the moment of depolarization: when the electrical activity causes muscular activity. This is the P wave of the ECG. Because of this electrical activity the atria contract.

- Onset ventricular diastole:
  at this moment the electrical impulse excites the ventricles which have just been filled from atria contraction. The QRS complex shows the electrical signal which induces ventricular contraction.

- Onset of ventricular diastole:
  at this point the ventricle relaxes and on the ECG this moment is signed with the T wave which signals repolarisation [35].

2.5 Heart Sounds

A normal healthy heart produces 2 sounds during the cardiac cycle: S1 and S2. Figure 2.3 shows the phonocardiogram (PCG) (time trace of heart sound amplitude) of the sounds a normal heart makes during one cycle. The PCG is a time trace of the recording of the acoustic sounds and murmurs produced by mechanical events of the heart valves and associated vessels. These sounds are produced by acoustic vibrations of the valves, muscles and blood flow [29]. A normal cardiac PCG cycle comprises the first heart sound (S1), the systolic pause segment after S1, the second heart sound (S2), and the diastolic pause segment after S2 [40]. In certain individuals a third heart sound (S3) can be heard. The timing S3 with respect to the other two sounds can be seen in Figure 2.6. The fourth heart sound (S4) and the heart murmurs can be heard in systolic interval and diastolic interval segments.

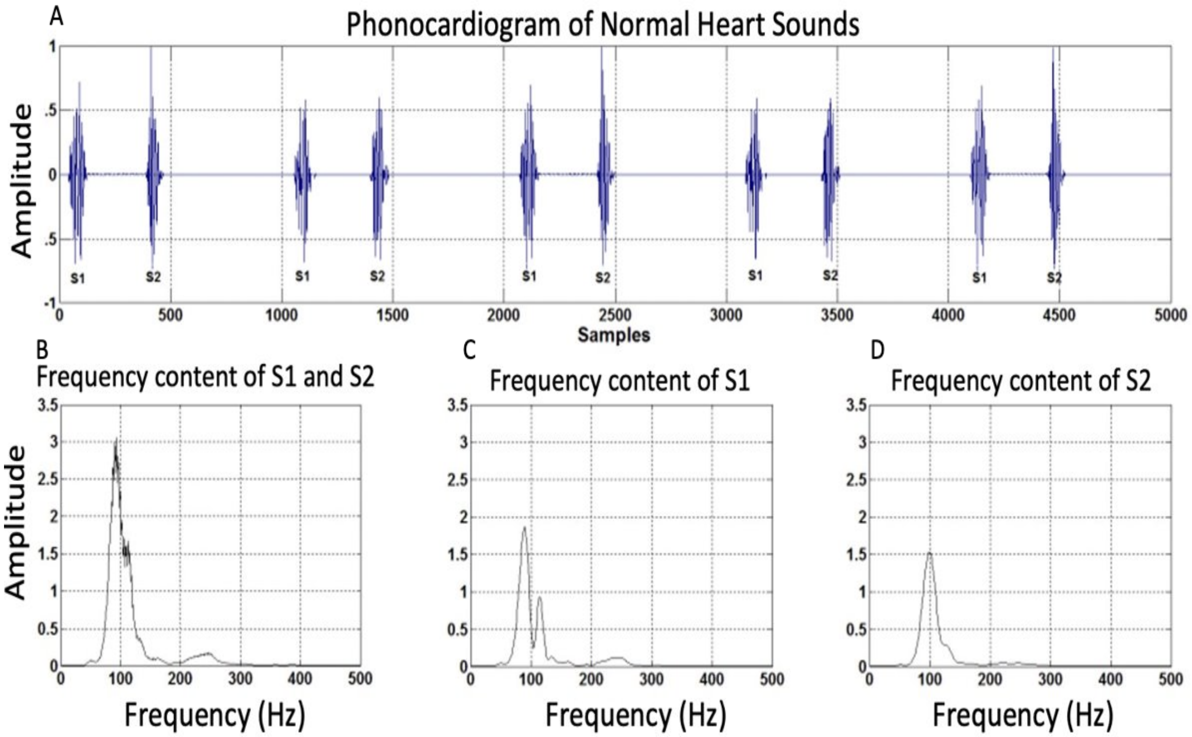


Figure 2.3: (a) shows a time trace of the heart sound that has been acquired with a stethoscope, the gold standard for PCG acquisition. Figure 2.3 (b) shows the frequency content of S1 and S2. Figure 2.3 (c) shows the frequency content of S1 and Figure 2.3 (d) shows the frequency content of S2. As it can be seen from the frequency content, there is no appreciable energy of heart valve sound above 300 Hz. Figure adapted from [44].

When analysing data acquired with the stethoscope it has been found that the spectral content the various sounds the heart makes is as follows:

- S1 on average has a time duration in the range of 70-150 ms. The frequency content of the sound is of 50-150 Hz, and just like human voice, every person can have a different S1 sound whose frequency content can take any pitch and frequency combinations in the range of 50-150 Hz.

- S2 on average has a time duration in the range of 60-120 ms. The sound can be any combination of the frequencies in the range 50-200 Hz.

- S3 on average has a time duration in the range of 40-100 ms. The sound can be any combination of the frequencies in the range 50-90 Hz.

- S4 on average has a time duration in the range of 50-80 ms. The sound can be any combination of the frequencies in the range 50-80 Hz.

Extra sounds that the heart can make are murmurs. While S1,S2,S3,S4 are generated mostly by the valves and the atria and ventricles contracting (more on this topic in the next sections), the

murmurs are caused by turbulent blood flow and can be heard in a frequency range between 20-1000 Hz [41, 43]. Murmurs are caused by two conditions>: stenosis and regurgitation. Stenosis is a condition in which the heart valves open less than normal and regurgitation is a condition in which a valve may not close completely and the blood flows backwards [44]. From this data we can see that the S1 sound is longer in duration and more low pitched than S2. The diastolic period (the time duration from S2 to S1) is longer than the systolic period (the time duration from S1 to S2) [42].
Another important point to take in consideration is that according to studies [45], when acquiring heart sound data with a stethoscope, the majority of the energy resided in the frequency range below 200 Hz and the bigger the size of the heart the less energy was found above 200 Hz.

In this chapter we will show the different representations of the sounds that a diseased heart makes and the gold standard for auscultation of these sounds. We will also discuss the causes of the different heart sounds and show their timing with respect to the ECG and also show the respective sound amplitudes.

2.5.1 Stethoscope

The gold standard for heart sound auscultation is either the acoustic or electronic stethoscope. In order to hear S1 and S2 sounds, the diaphragm of the stethoscope is placed on the patient's chest, the sound waves produced by the heart travel to the skin on the chest, which vibrates, this in turns vibrates the diaphragm, creating acoustic pressure waves which travel up the tubing to the listener's ears. In order to hear heart valve sounds the stethoscope is placed in these locations (see corresponding locations in Figure) [46]:

(1) Left of sternum, 2nd rib down- pulmonic valve.

(2) Right of sternum, 2nd rib down- aortic valve.

(3) Left of sternum, 4th rib down- tricuspid valve.

(4) In line with left nipple, 5th rib down- mitral valve [47]. The data about heart sound frequencies provided in the previous section is based on stethoscope measurements.

The stethoscope can also be used to hear pressure sounds from the arm, by placing it under an inflating cuff or from the aortic artery on the neck to hear certain types of murmurs. However S1, S2, S3 and S4 and other kinds of murmurs can only be heard from the chest [47]. This is an important premise which is at the basis of the interest of our findings which we present in this thesis. While the digital and electric stethoscopes can only acquire heart valve sounds from the chest and can only acquire frequency content up to 300 Hz (according to literature), with our method we can acquire heart valve sounds from the neck and we can acquire frequency content up to 750 Hz.

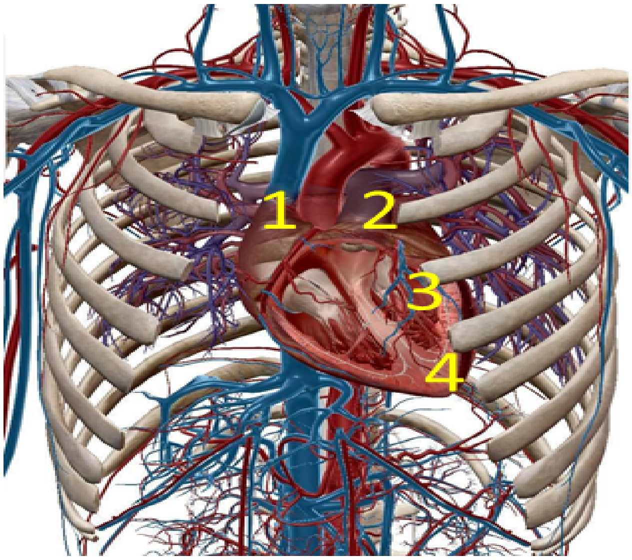


Figure 2.4: shows the various locations that clinicians use to hear heart valve sounds with the stethoscope. Figure made with 3D4Medical Software.

.

Below we will give an overview of all the different sounds the heart makes with complementary figures. Some sounds will be represented with a simple line, not because that is the nature of the frequency of the sound, but because the frequency content of that sound is not of clinical relevance, rather the presence or absence of that sound is of clinical relevance.

2.5.2 Normal S1 sound

The S1 sound that the heart produces is made by the closure of the mitral and tricuspid valves. This happens at the beginning of ventricular systole.

S1 is produced by three actions:

- movement of blood within the ventricles.
- cardiac vibrations from the ventricle walls.
- closing of the mitral and tricuspid valves.

The S1 sound can be split in two in certain individuals. The first component, referred to as M1, is produced by the closure of the mitral valve; the second component, T1, is produced by the closure of the tricuspid valve [48].

S1 has some characteristics which depend on:

- The intensity: The intensity of S1 is directly proportional to the force of ventricular contraction.

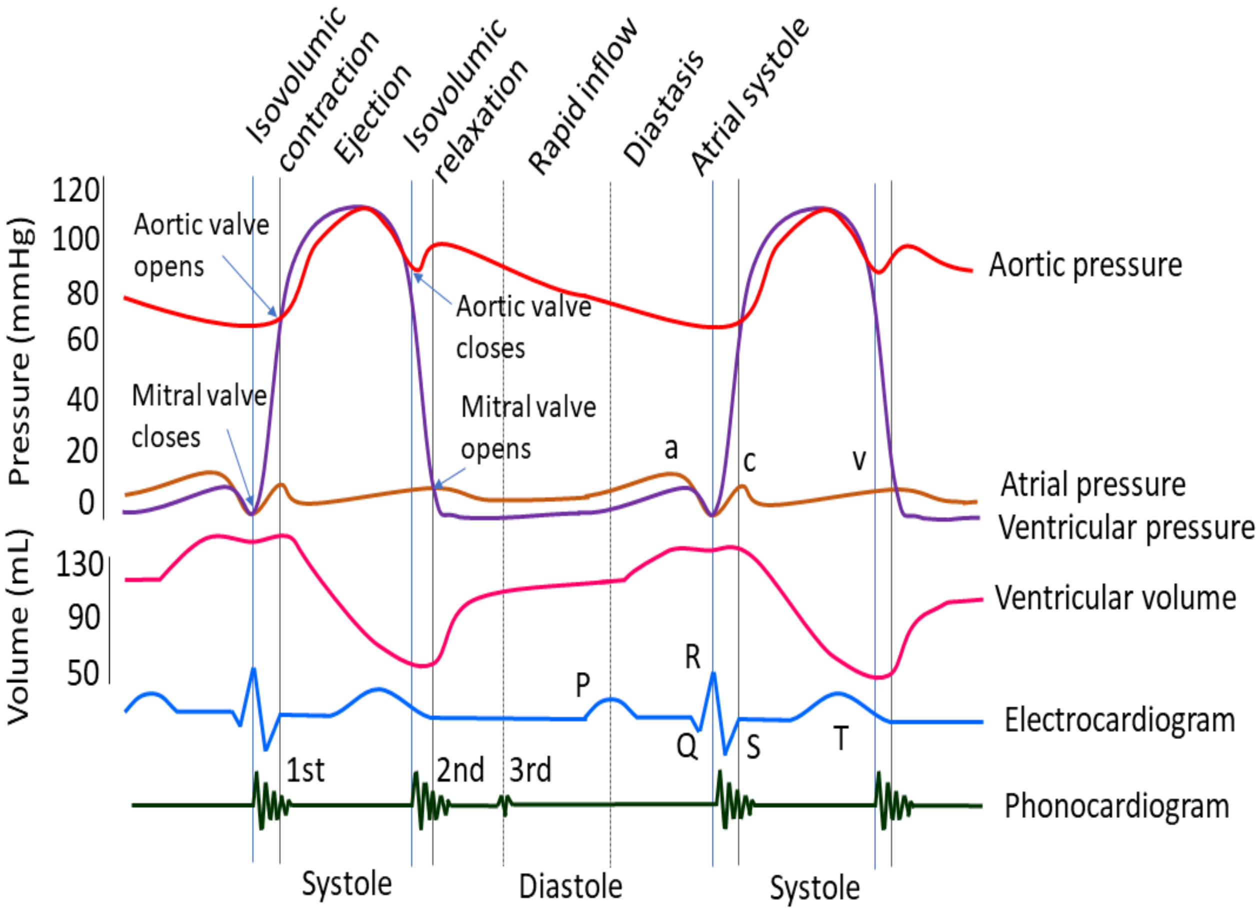


Figure 2.5: shows the various parameters of the heart cycle that we have discussed so far. At the bottom we can see the time trace of the heart sounds. The first heart sound happens at the beginning of the systole and the second at the beginning of the diastole. We can see from the ventricular volume line that during the systole the ventricles contract and push blood out, therefore there is a dip in the line, while during diastole the ventricles refill. The ventricular pressure, blue line, is high during systole because the ventricles contract, shown in the blue line as the big increase during systole.

- PR interval: From Figure 2.5 we can see how the PR interval deriving from the electric impulse of the heart matches with the sounds produced during the cardiac cycle. If the PR interval is short it causes the mitral and tricuspid leaflets to open more widely as the ventricles contract, impacting the sound.

- A slower heart rate produces more intense vibrations when the valves close thus producing a louder S1 and an S2 of longer duration [48].

2.5.3 Abnormal S1 sound

If the electrical activation and contraction of the right ventricle is delayed than the time interval between M1 and T1, which is usually of 20-30 ms, will be widened. This is because there will be a delay in the tricuspid valve closure. An abnormally wide time split between M1 and T1 occurs when the subject might be suffering from heart conditions [50].

2.5.4 Normal S2 sound

S2 is produced by the cardiac vibrations happening at the closing of the aortic and pulmonic valves, which happen at the start of diastole, after the ventricles have contracted, causing the opening of these two valves. Then the ventricles relax and these two valves close, producing the sound. As the pressure of the ventricles fall rapidly (we can see from Figure 2.5 blue curve that the pressure the ventricles exerts is much higher than the pressure that the atria exerts) it causes a slight backflow of blood from the aorta and pulmonary artery. It is also caused by the sudden deceleration of blood in the aorta and pulmonary artery. Normally, the aortic valve (A2 sound) closes slightly before the pulmonic valve (P2 sound) because the pressure is higher in the aorta than in the pulmonary artery. Because there is more pressure on the aortic valve, the sound is louder. This split does not usually signal a problem with the heart. The characteristics of the S2 sound are [49] :

- The intensity of the S2 sound is directly proportional to the amount of pressure on the pulmonic and aortic valves.
- It has a slightly shorter duration than S1.

The splitting of S2 can be best heard during inspiration because inspiration reduces the pressure in the pulmonary artery, which increases the blood flow through the veins which returns to the right side of the heart. This delays emptying of the right ventricle, prolonging right ventricular ejection time, and thus it delays the closure of the pulmonic valve, which results in the P2 happening at a longer interval of time after A2 [48].

Abnormal S2 sound

As with the S1 sound, the S2 sound is also caused by the closure of two valves: aortic and pulmonic, thus S2 is also composed of two sounds which are actually very close in time. It is possible however, to listen to these two sounds distinctively the one from the other (A2 first and P2 afterwards) during inspiration and expiration of the patient. P2 is usually louder when the patient is affected with certain diseases [51, 52]. On the other hand, the intensity of A2 decreases when the patient is affected by certain other diseases. An abnormal S2 splitting is related to valvular dysfunction which cause the normal S2 split to be absent during both phases of the respiratory cycle [48]. Another kind of split is the narrow minded split, which is heard during inspiration and expiration but it is shorter than the persistent split. Another cause A2-P2 splitting during expiration is delayed electrical activation of the right ventricle, which delays P2. Another kind of abnormal splitting heard during expiration or inspiration is the widened S2 split which might be due to a delayed closing of the pulmonic valve which is the result of certain heart diseases [48].

2.5.5 S3 sound

S3 is caused by ventricular filling rather than by valve closures. S3 sound can be either physiological or be the sign of disease. S3 can be heard after S2. S3 is caused by vibrations occurring during rapid, passive ventricular filling. Early in diastole the mitral and tricuspid valves open and the ventricles fill and expand. The more vigorously the left ventricle expands, the greater the chances are that an S3 will occur. S3 is physiological for children, very active teens, patients with anemia, that have fever or are pregnant. When an older individual has an S3 sound it could be caused by acute myocardial infarction (MI), acute alcohol withdrawal, cocaine abuse or stroke. In older adults and elderly patients, an S3 may be the first indication of heart failure [48].

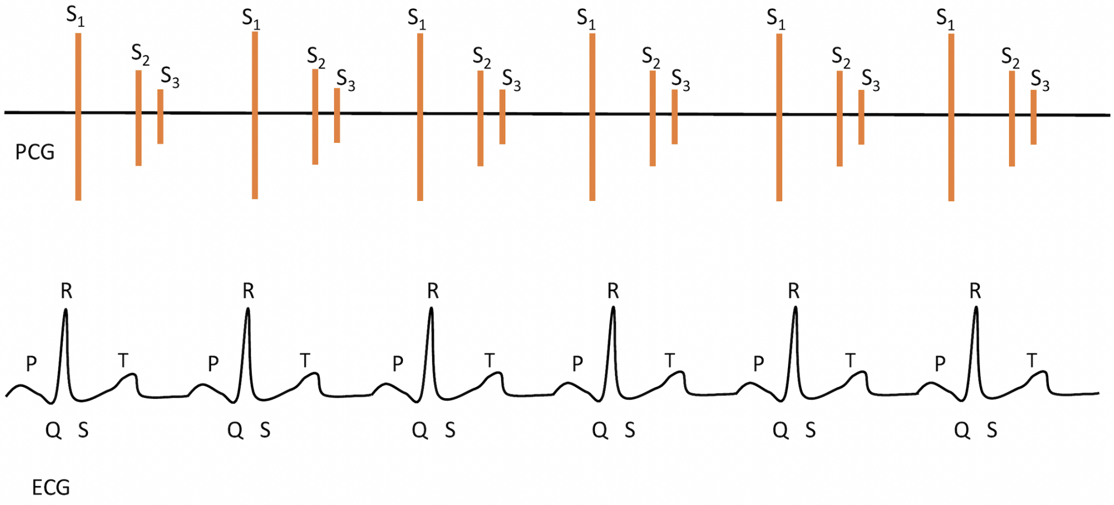


Figure 2.6: shows on the top the schematics of an S3 sound. This S3 sound is produced by the left side of the heart. As discussed, S3 heart sound is only normal if heard in patients younger than 20 years old or very active. As it can be seen from the time trace of the ECG, the S3 sound happens between the T-P wave.

Abnormal S3

An abnormal S3 is caused by an increased blood volume and inflow velocity into the left ventricle. A pericardial knock is an S3 sound which occurs less than 0.14 second after S2.

Right Sided S3

S3 is usually caused by the left ventricle. When S3 is caused by the right ventricle the sound is always produced by a diseased heart.

2.5.6 S4 sound

S4 is due to the atrial contraction at the end of diastole when the atria are still contracting, this further stretches the ventricles as they fill. The vibrations caused by this stretching and filling gives rise to the S4 heart sound

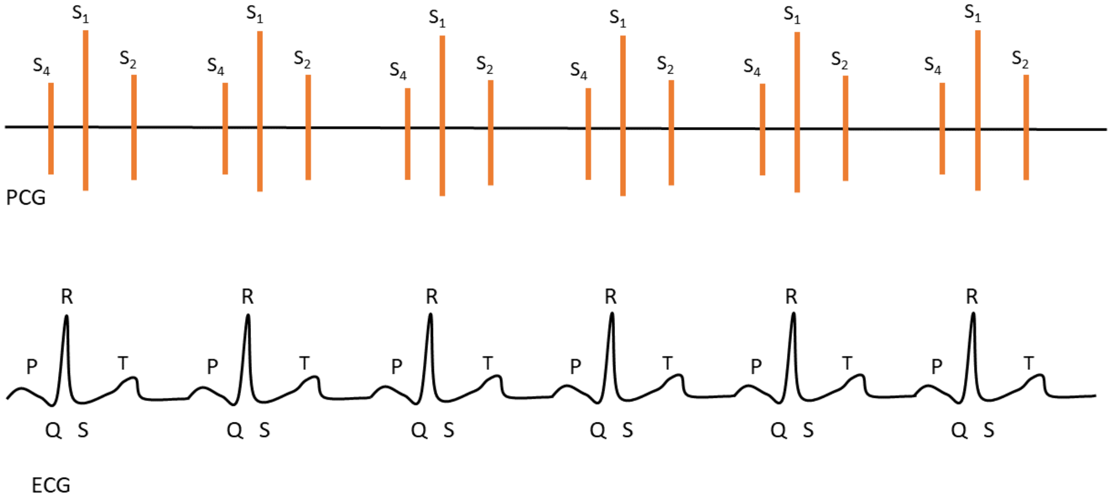


Figure 2.7: shows the schematics of an S4 sound. S4 heart sound is only normal if heard in athletes. It occurs during the P-R interval of the ECG, and in the PCG trace it shows before S1.

Abnormal S4

S4 is always abnormal expect when heard in athletes. S4 can also be generated in the right ventricle [48].

2.5.7 Other Systolic and Diastolic Sounds

We will now discuss some of the other heart sounds the heart produces when affected by a disease.

Opening Snap

As we have previously seen, at the end of ventricular systole after the aortic and pulmonic valves close, ventricular pressure falls. When ventricular pressure is less than atrial pressure, the mitral and tricuspid valves open. In a healthy heart, the mitral and tricuspid valves open silently. However, for diseased individuals, atrioventricular valves will open more rapidly than normal and make a sound known as an opening snap (OS). This opening snap can be caused by either the mitral or the tricuspid valve. Different diseases are involved depending which valve caused the sound.

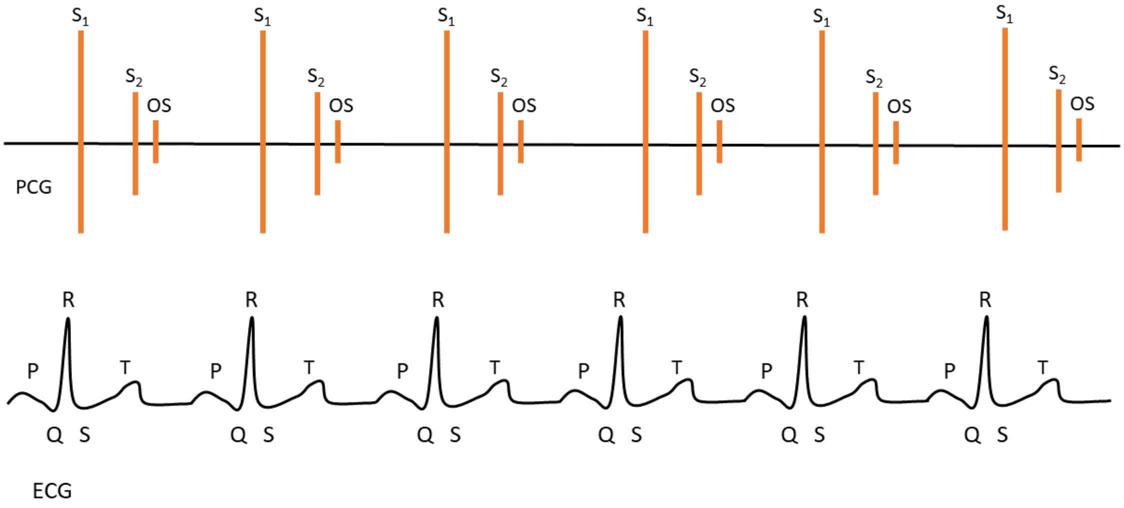


Figure 2.8: shows the schematics of the timing of the opening snap is relation to the timing of S1 and S2 sounds. The opening snap sound takes place right after the T-wave if the PCG is looked in comparison with the ECG.

Systolic Ejection Sound

Systolic ejection sounds are caused by the opening of the aortic or pulmonic valves. It is abnormal regardless which valve has caused the sound. Depending on the valve which has caused the sound these can be called [53]:

- Pulmonic Ejection Sound:
  A high pitch click that is heard near S1. It's location in a PCG and relative to an ECG is shown in Figure 2.9.

- Aortic Ejection Sound:
  Systolic ejection sounds are due to defects that obstructs or narrows the aortic valve opening. Its time relationship to S1 and S2 and ECG relationship is shown in Figure 2.10.

Mid-Systolic Click

A mid-systolic click is caused by a prolapsed mitral valve.

2.5.8 Heart Murmurs

Blood usually flows in a laminar fashion, which means that all the blood layers flow in unison with one another. Laminar blood flow does not produce any sound. If the blood, which usually travels in bulk and does not make any sound starts to experience an acceleration or encounters a flux of blood from a different direction of flow due to a faulty valve, the flux is not laminar anymore but it becomes turbulent (it does not travel in bulk anymore). This turbulent blood flow produces sounds. Turbulent blood flow can be due to:

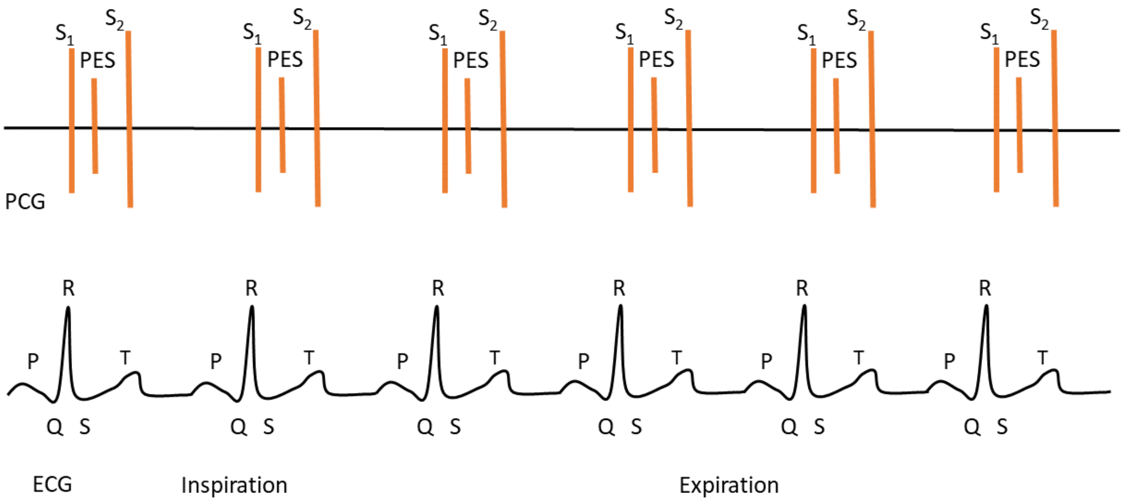


Figure 2.9: shows the pulmonic ejection sound in relation to the timing and amplitude of S1 and S2. It's timing position is close to S1. Compared to the timing of the ECG, the sound is heard after the QRS complex.

- Stenotic and Insufficient Valves:
  A stenotic valve does not open completely, thus the blood which must flow through very fast produces a sound that is similar to the whistling sound produced by the lips when air if forced through the narrow opening.
- Insufficient Valve:
  An insufficient valve does not close correctly. Thus the blood flows backwards. This produces a swishing murmur.

Furthermore, the timing of the murmur is of significance in diagnosing the pathology since if it happens between the first and second heart sound then it is a systolic murmur (due to the valves between atria and ventricle) while if it happens after the second hear sound and before the first heart sound is called a diastolic murmur, which is due to the valves between the ventricles and the big vessels.

2.5.9 Systolic Murmurs

Systolic Ejection Murmurs

During ventricular systole, because the ventricles need to pump the blood to the rest of the body, thus applying a greater force than the atria need to, the flow of blood is not laminar but turbulent. This gives rise to an innocent systolic ejection murmur (SEM). This sound is normal for children and in individuals with very specif conditions. If heard on anyone else this sound is a symptom of a dilation in either the aorta or pulmonary artery .

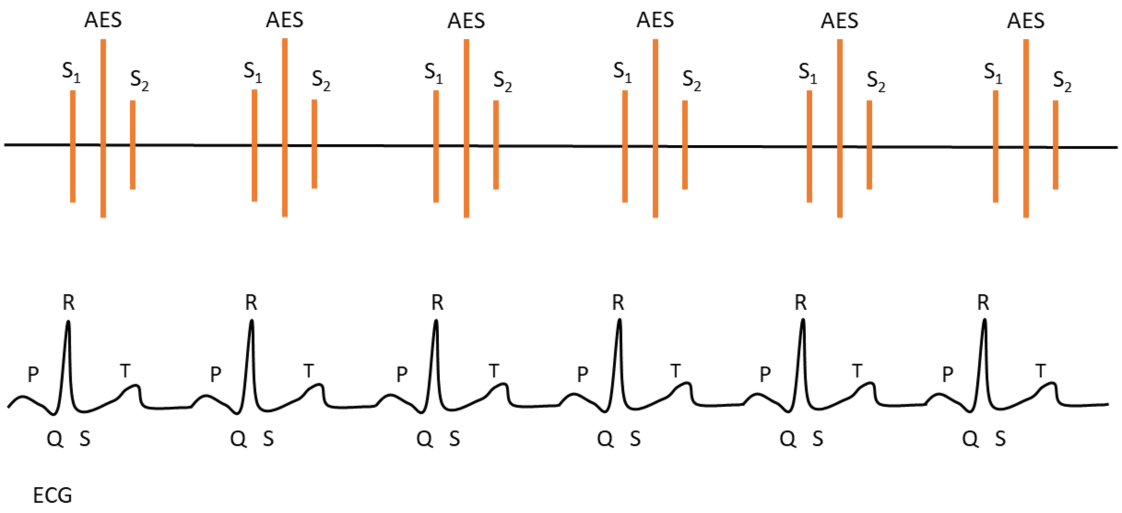


Figure 2.10: shows the aortic ejection sound in relation to the timing and amplitude of S1 and S2. It's timing position is close to S1. Compared to the timing of the ECG, the sound is heard after the QRS complex. It is a soft sound and can be as high in amplitude as S1, however it is shorter in time.

Supravalvular Pulmonic Stenosis Murmurs

Supravalvular pulmonic stenosis is a type of right ventricular outflow obstruction that occurs above the pulmonic valve. The murmur that this obstruction creates has a crescendo-decrescendo configuration but occasionally it can be continuous.

Pulmonic Valvular Stenosis Murmurs

A pulmonic valvular stenosis murmur's sound is produced by a pulmonic stenosis valve which means it has a problem in the way it opens. In this murmur, the S1 sound is normal. The murmur begins after S1 and it ends before S2. The intensity of P2 is normal.

Subvalvular Pulmonic Stenosis Murmurs

When the obstruction is beneath the pulmonic valve, the murmur sound is the same as the pulmonic valvular stenosis murmur sound, but it is not initiated by the PES.

Remaining Diseases

There are many other diseases which cause a change in the normal heart valve sounds. For this reason in the tables 5, 2, 4 it is presented a summary of all the remaining not yet discussed different heart valve sounds produced by all possible diseases.

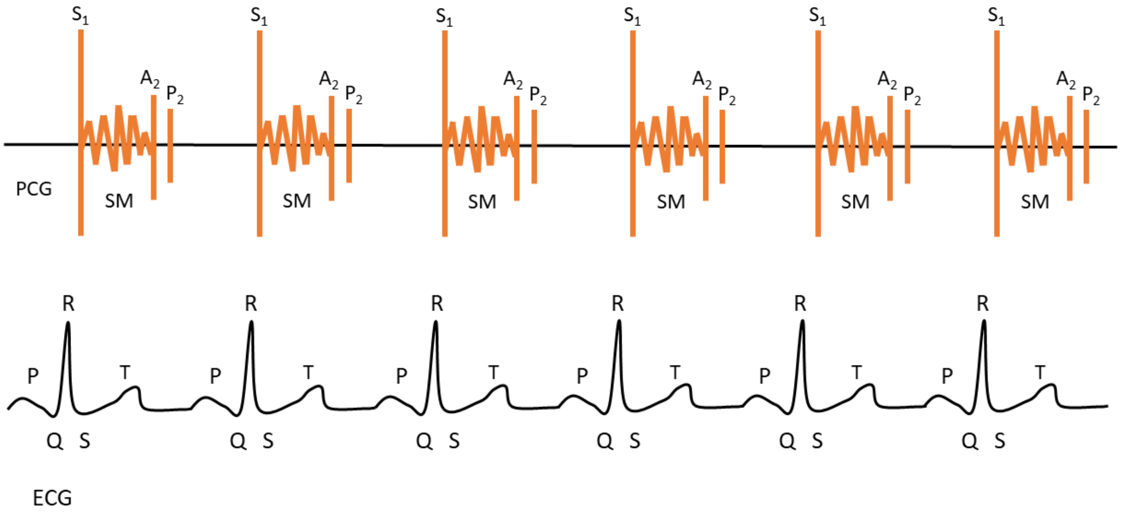


Figure 2.11: shows the systolic ejection murmur. It happens early during systole and ends before S2. It's heard after the QRS complex of the electrocardiogram.

2.6 Cardiac Output

The amount of blood pumped by the heart per minute depends on the heart rate and stroke volume.

- Heart Rate:
  The average resting heart rate is 70 beats of the heart per minute while the average amount of blood that is pumped when the heart contracts is 70 mL. This means that the heart pumps around 5 liters of blood per minute. Since the body on average contains about 5 liters of blood, this means that the whole amount of blood contained in the body is pumped by the heart each minute.

- Stroke Volume:
  The stroke volume is the amount of blood pumped by the ventricles with each heart beat [35].

2.6.1 Blood Pressure

In order for the valves between the ventricles and the major arteries to open, the ventricles must contract in such a way as to generate a pressure which will be higher inside the ventricles than in the major arteries. This is the arterial blood pressure and it is called afterload. In individuals which have valve problems, the pressure the heart needs to produce is much more elevated than normal. This can lead to heart failure [35].

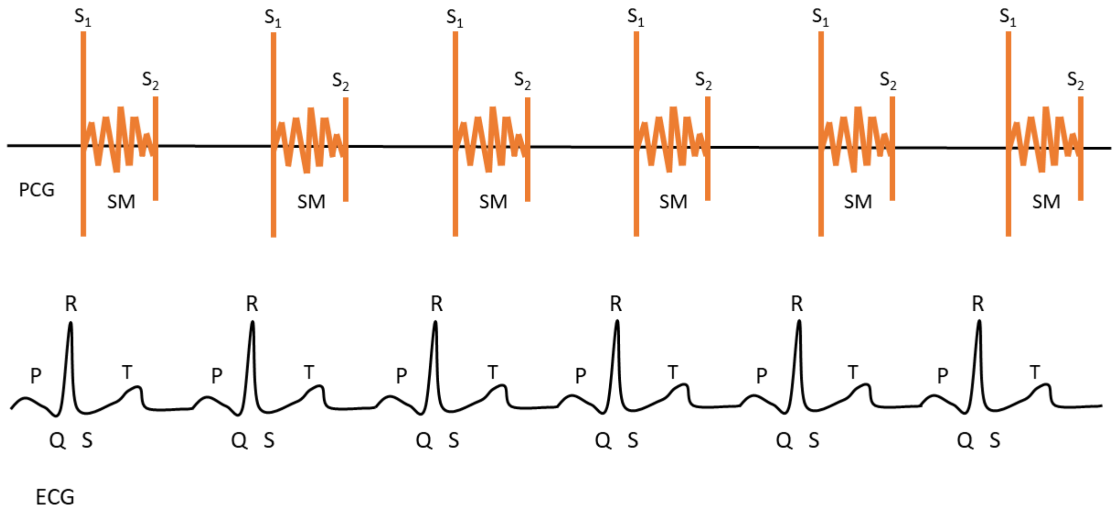


Figure 2.12: shows the timing and amplitude of supravalvular pulmoninc murmur. As it can be seen, the murmur first has a crescendo and then a decrescendo but it occasionally is continuos. It begins after S1 and ends before a S2. On the ECG waveform, it corresponds to timing after the QRS complex begins and ends just before the T wave ends.

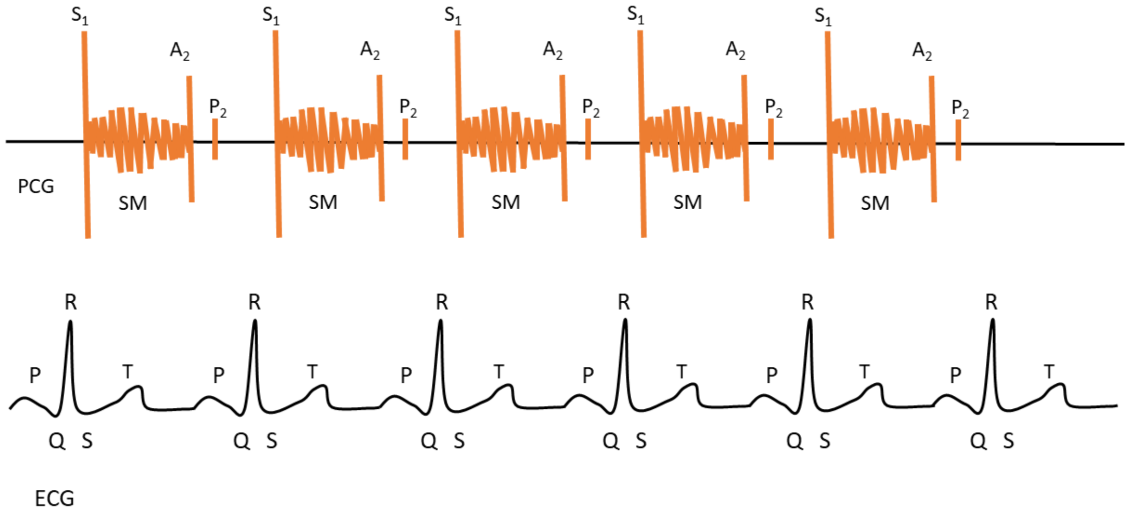


Figure 2.13: shows the amplitude vs time of the pulmonic valvular stenosis murmur. The murmur has a crescendo and decrescendo shape. It ends before S2. It's accompanied by a PES. It starts after the QRS complex of the ECG and ends before the end of the T wave.

2.7 Heart Failure

Heart failure ensues when the heart cannot keep up with the demand for supplies of nutrients and removal of waste from the body, because as seen in the very beginning of the chapter, the heart must maintain homeostasis. There are two types of heart defects that can lead to heart failure: systolic heart failure, where the heart has a problem pumping the blood and diastolic heart failure which is when the heart has a problem in refilling.

2.7.1 Defect in Systolic Heart Failure

Systolic heart failure happens due to decreased cardiac contractility. This means that the heart cells contract less, thus pump less blood. If the condition goes on for too long both ventricles may continue to weaken and eventually fail. This condition can be due to: damage to the heart muscle due to a heart attack, defects to the valve which imply prolonged high blood pressure [35].

2.7.2 Diastolic Heart Failure

While systolic heart failure is related to the ability of the ventricles to contract properly to pump the blood around the body, diastolic heart failure is related to the heart's ability to expand properly to fill up with blood. In these conditions the atria do not expand properly thus they do not fill up sufficiently and as a consequence less blood will fill the ventricles that although will pump properly, they will not pump enough blood [35].

2.8 Coronary Artery Disease

In physiological conditions, as the body demands more oxygen under stress conditions, the heart can keep up with this need by pumping more blood around the body. However, when someone is affected by coronary artery disease, pathological changes within the coronary artery allow less blood to be able to flow through the vessels, thus the oxygen need to the organs is not met. This means that during rest conditions the individuals affected might (or might not) still have enough oxygen, but in conditions of stress and exercise the heart cannot supply the demand of oxygen.

Death by coronary artery disease is the leading cause of death in western countries. It accounts to about 50% of deaths worldwide. This amounts to a number greater than the deaths from all cancers combined. In this section we will briefly see how coronary artery diseases lead to heart attack.

2.8.1 CAD diseases

The various CAD diseases are:

- Vascular Spasm. This is a disease which manifests in an abnormal spastic constriction that narrows the coronary vessels.
- Atherosclerosis. Progressive degenerative disease which leads to vessels blockage through plaques which fill the blood vessels.

2.9 Conclusion

In conclusion, we explored in this chapter the various diseases that can affect the heart. We have seen that many diseases can be diagnosed by heart valve sounds because they manifest themselves in a way that changes the heart valve sounds and which is different for different diseases. We have seen that from the frequency content, pitch and timing of the heart sounds it is possible to diagnose many heart diseases. Although as we have seen, normal heart valve sounds usually consist of an S1 and an S2 sounds, both S1 and S2 can vary within a vast frequency range in amplitudes of frequencies, pitch, duration etch the same way the human voice varies from person to person. For these reasons in this thesis we will show a method to distinguish healthy vs unhealthy heart valve sounds acquired from test subjects and since the sounds the heart valves make are unique from person to person, we will also show a way to perform people identification through these sounds.

Chapter 3

Wavelets in Classification Problems

Our brains have developed in such a way that we are now able to interpret the world around us by selectively picking out only the most important information and discarding an endless multitude of irrelevant data. This allows us to process faster with the sparse and relevant bits of data that are needed. In order to make algorithms that can solve classification problems as efficiently as humans do, we need to make algorithms which implement these qualities: efficient selection of only the most relevant details.

To extract useful information from time signals we can decompose them over elementary waveforms. In this chapter the chosen elementary waveforms are called wavelets. These are adapt to extract only the most important components of the signal. The father of wavelets is the French matematician Stéphane Mallat, and in this chapter we will discuss his work on wavelets, analysing some of his books and peer reviewed articles to try to understand how wavelets pick the hay out of the haystack of information like our brains do.

3.1 Sparse representation

The Fourier Transform (FT) uses frequency filtering operators to represent any function of time as a sum of sinusoidal waves e^{iwt}. That is because sinusoidal waves define an orthogonal basis. Any signal can then be represented as:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(w) e^{iwt} dw \tag{3.1}$$

where the amplitude of each sinusoidal wave is given by how much it is correlated with the original signal f $\langle f(t), e^{iwt} \rangle$. In this equation, w is the frequency. This representation of the signal is called Fourier Transform and is given by

$$\hat{f}(w) = \int_{-\infty}^{\infty} f(t)e^{-iwt}dt \tag{3.2}$$

So the new representation we obtain of our function is the collection of Fourier coefficients. These coefficients allow us to discover the properties of the function. By looking at how fast the Fourier coefficients decay, we can infer the regularity of the function. If we look at a function in the Fourier domain, we will see that a function composed of low frequencies will decay fast as it doesn't have high frequencies coefficients, thus all the energy will be concentrated near 0. So, for a regular function (a function composed of low frequencies), the decay in the w direction will be fast.

However, although the Fourier Transform will show us that the function presents irregularities through the high frequencies coefficients, it will not allow us to locate in time where these irregularities can be found. Therefore we have global information of a function but not localized information [54]. In order to obtain localized information we need to use different basis functions than sine waves.

The pioneer of the Morlet (Gabor) wavelet function were Dennis Gabor in 1946, which introduced a Gaussian-windowed sinusoid for time frequency decomposition [55].

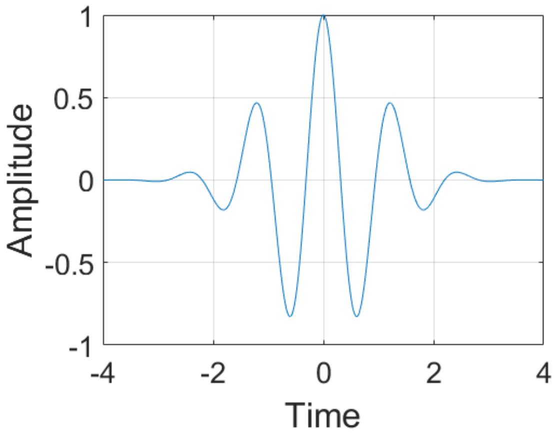


Figure 3.1: shows the shape of a Morlet wavelet. Compared to the sine basis, the wavelet basis is localized in space and oscillates.

The above figure shows the Morlet wavelet, whose formula, for convenience, we will denote just by defyning it as Ψ.

$$\int \Psi(t)dt = 0. \tag{3.3}$$

Because wavelets are localized in time, in order to cover the full function space, they need to be translated, in order to do so, to the wavelet formula we will add a translation parameter u, which denotes the location of the wavelet [56].

$$\Psi_{u,s}(t) = \frac{1}{\sqrt{(s)}}\Psi(\frac{(t-u)}{s}). \tag{3.4}$$

where u is the location of the center frequency. And then because we want to explore all scales at which we can squeeze and dilate the wavelet, we also introduce a parameter s [57]. The wavelet transform is then a convolution with our signal and all the wavelets with different locations and scales. What Morlet and Grossman realized is that the function can be reconstructed from the wavelet coefficients [57].

Hitherto we have seen a basis that not only recovers frequency information as the Fourier Transform does, but also recovers time information. This is a very important extra bit of information offered by wavelets, which allow us to locate the time at which a certain frequency happens, information that cannot be retrieved with the Fourier Transform.

If we look at figure 3.2, at the top we can see a time signal f(t) delineated with a black line and underneath its wavelet transform. The wavelet transform is a 2d image because there is a point for each wavelet at different locations and scales (u,s). The grey color indicates coefficients which are equal to 0, the whites are positives and the black are negatives. We can see that the large amplitude coefficients correspond to points of f(t) where there are sharp transitions, singularities of the function. So, within this image, we want to keep the important coefficients which we can select by using an orthogonal wavelet basis. Because there is less information in time for low frequencies and more information in time for high frequencies, we can use this as a boundary condition to remove some of the coefficients. So we just keep scales which are powers of 2 (shown in the wavelet transform in Figure 3.2 to highlight the concept of where the power of 2 coefficients are and how they are scaled), and for any scale the wavelet is going to be translated by the scale 2^j so that the number of coefficients kept is just a fraction of those which compose the image [58]. The idea is here is that we will keep less coefficients for low frequencies, where we know he function isn't changing much and we will keep more coefficients at the high frequencies.

Hitherto we have established we will only consider wavelets which have scales of 2^j in order to

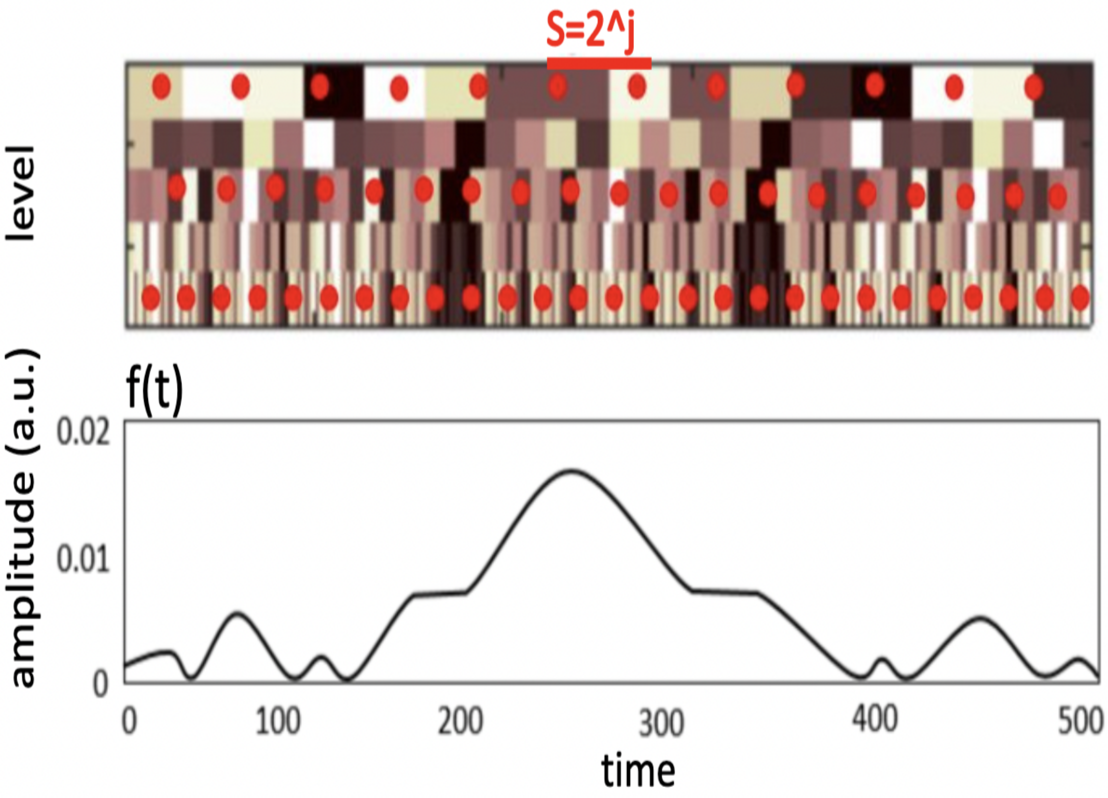


Figure 3.2: shows a time signal f(t) and above its wavelet transform, which is 2D image because each coefficient is computed by convolving the signal with a particular dilation and translation of a family of wavelets. The red dots show the sparse acquisition of coefficients that orthogonal basis wavelets acquire, the selected coefficients are separated by a distance 2^j.

get a sparse representation of the signal [54].

$$\Psi_j(t) = \frac{1}{\sqrt(2^j)\Psi(\frac{-t}{2^j})}. \tag{3.5}$$

As we can see from Figure 3.3, the wavelet filters are bandpass filters which are dilated with respect to each other by a factor of 2j. These filters in the frequency domain are multiplied with the Fourier Transform of the signal. What this discretization with power of two sampling means is that we can reconstruct our signal from this 2j specific scale if the sum of all dilated wavelets is 1.

$$f(t) = \sum \int \langle f, \Psi_{u,2^j} \rangle \Psi_{u,2^j}(t) du. \tag{3.6}$$

Another property of wavelets is that they capture with many coefficients and high amplitudes of these coefficients the singularities of the signal and represent with few coefficients and very low amplitude of these coefficients the parts of the signal which are regular. And this is how wavelets obtain sparse approximations. It is proven that the local maxima of the wavelet trans-

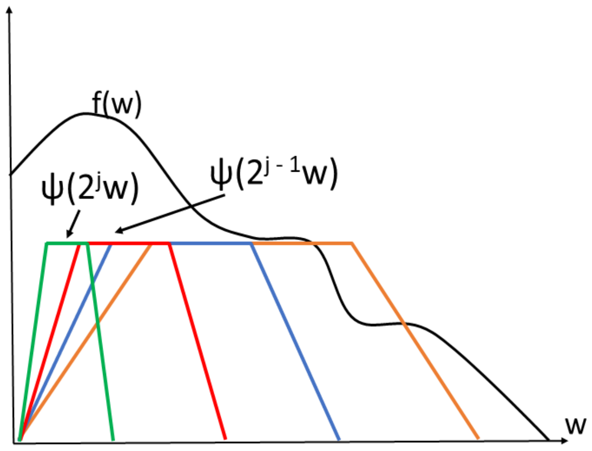


Figure 3.3: A representation of wavelet filters (shown in different colors), which act as bandpass filters, dilated the one with respect to the other by a factor of 2j. In black is represented the FT of the signal they are trying to recover: f(w).

form modulus detect the locations of irregular structures [54]. This is because we can keep only the coefficients above a certain amplitude threshold and still reconstruct a good representation of our original signal. So we are putting many coefficients near singularities so that we can reconstruct the shape of the function while taking few coefficients where the function is regular so that we can reduce computation time and extract only the important information of the function.

What P. Burt then realized, is that it is possible to analyze signals at different scales by averaging through convolution and only keeping one sample out of two of the original signal and then repeating the process [59]. Then Mallat followed this line of thinking and used wavelets to approximate signals at different resolutions by projecting them into spaces of progressively smaller sizes. The relationship of multiscale resolution of signal representation and wavelets is that with different translations of wavelets we can reconstruct the regularities and singularities of the signal. We can extract detail and approximation coefficients by taking two wavelets translated differently the one with respect to the other in the Fourier domain. This is possible because wavelets are bandpass filters. So if we take a wavelet filter that covers the low frequency of the signal we can convolve it with the signal to reconstruct the approximation coefficients while a highpass filter will recover the detail coefficients. The process is then repeated. The low frequency signal is again convolved with a lowpass and highpass wavelets bandpass filters.

This concept is shown in Figure 3.4 On the top of Figure 3.4 we can see the structure of the fast wavelet transform. The signal f(n) is convolved with two shifted bandpass wavelets, one which covers the high frequencies and is shown in the graph below in red, and another which goes from $\frac{Fn}{2}$ to 0 but is not shown on the graph and which would cover the low frequencies. Fn is the maximum frequency of the signal as dictated by the Nyquist theorem. After the signal goes through the first highpass (h(n)) and lowpass (g(n)) filters, it is downsampled by 2 because

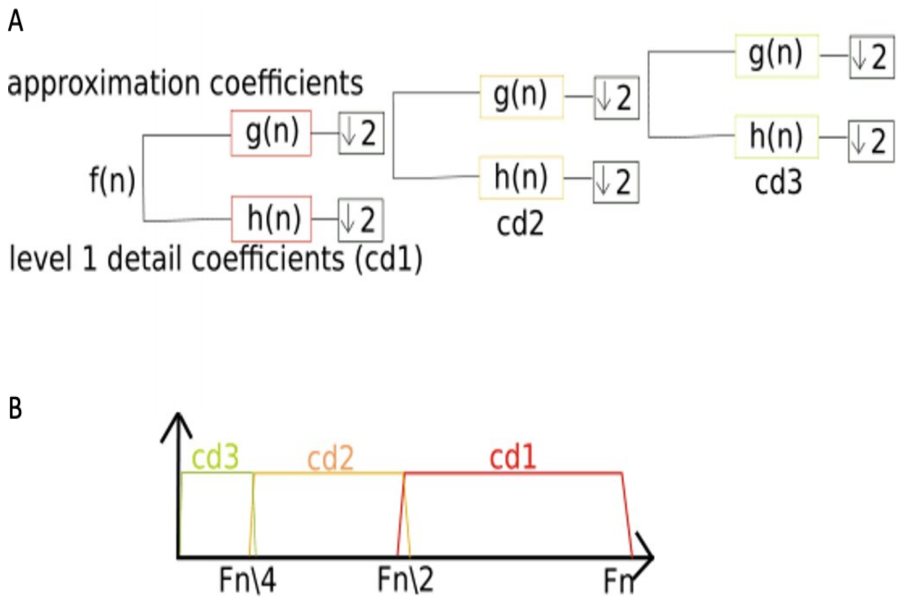


Figure 3.4: Figure A shows the structure of the fast wavelet transform: a signal, f(n) is convolved with a low g(n) and a high h(n) pass wavelet filters to produce approximation and detail coefficietns respectively. These coefficients are then subsampled by 2 and are called level 1 coefficients. The process then repeats in the Figure up to obtain level 3 coefficients. Figure B shows the wavelets highpass h(n) filters used to obtain the different levels coefficients in the respective colors corresponding to the levels shown in the structure above.

in each case half of the frequencies have been dropped so we only need half as many samples to construct the signal. The process then continues forward and the frequencies between $\frac{Fn}{2}$ to 0 are now convolved with a lowpass and a highpass filter and then the signal is subsampled by 2 [60].

What we have just explained in this brief chapter is quite remarkable: we can use wavelets to perform multiscale processing which we can use for signal classification. Wavelets allow us to contruct a multiresolution algorithm which allows us to study the signal at different resolutions while keeping only the most essential details at each resolution level by focusing on the changes of the signal rather than its regularities. If we take an image as a signal example, wavelets would only keep the edges and discard uniform light amplitude values. We can muse here that the human brain can recognize an image just from the outlines of it, and we have just developed an algorithm which exploits this concept for signal classification.

In the next section we will explore how such a multiscale wavelet algorithm can help us solve classification problems.

3.1.1 Wavelet Families

As we have seen in the previous section, we can use wavelets to perform a decomposition which allow to decompose the signal at different resolutions.

One parameter that we can choose in such a decomposition is the wavelet filter we want to use. The shape of the wavelet needs to be chosen depending on the signal to be decomposed. Since the wavelets filters are convolved with the signal, the closer in shape the wavelet is with the signal the better it will represent it.

When choosing which family of wavelets to pick, there are some parameters to consider. The first is the vanishing moment of the wavelet, which tells how fast the wavelet decays in time. The number of vanishing moments are proportional to the wavelet oscillations for orthogonal wavelets. The larger the number of vanishing the moments the more oscillations the wavelet filter will have. There are many wavelet families to choose from, from example, Daubachies and Morlet wavelets, in wach family there are different wavelets, Morlet1, Morlet2, etc. The number 1, 2, etc. next to the wavelet refers to the number of vanishing moments (loosely proportional to oscillations) the wavelet has. The larger the number of vanishing moments the larger the number of samples points the filter has (filter length). The number of vanishing moments has a correlation to the smoothness of the function. The larger the number of vanishing moments the smoothest the function is.

Another parameter regarding wavelets family differences is the symmetry. Some wavelet filters are symmetric, while others are semi or asymmetric.

The next parameter is orthogonality. If the wavelet is orthogonal, the wavelet transform preserves the energy of the signal the wavelet is convolved with [61].

- Feature extraction:
  If we want to extract closely spaced features, we should choose wavelets with low vaninisging moments

- Denoising:
  Orthogonal wavelets like Symlets and Daubachies are good for denoising because of their orthogonality which implies it conserves the energy of the signal. These wavelets are good for 1D signals.
  For 2D signals, bi-orthogonal wavelets are the appropriate wavelet choice for denoising because they are symmetrical, which means they have a linear phase, which means it will not introduce linear distortions for image reconstruction.

- Compression:
  For compression wavelets with higher vanishing moments should be selected because they select fewer coefficients and the majority of coefficients is neglected. They should be used for the opposite reason that low vanishing moments are used. The latter can recover closely spaced features while the former ignores them. Furthermore, as we have already mentioned, the higher vanishing moment means that the wavelet is smoother, thus a smoother reconstruction of the image can be achieved.

- Discontinuity detection:
  To better represent a signal which changes in time it is better to use wavelets with a high vanishing moment because this type of wavelet is better suited to detect the moment the signal has a change in amplitude time (for example from being stationary to suddenly presenting a bump).

3.1.2 Thresholding Methods

We have seen so far how we can decompose the signal to perform a multi-resolution and sparse analysis through the fast wavelet transform. After we have retrieved the coefficients, some will have high values and some will have low values. The coefficients with low values will be those representing noise while those with high values will be the coefficients representing signal. Thresholding allows to remove those low amplitude coefficients representing noise.

The two most common thresholding methods are hard and soft thresholding. With the hard thresholding method, all coefficients which fall below a specified range are set to 0. While in soft thresholding, the coefficients below a specified range are set to 0, but all the other coefficients are attenuated by a constant.

3.2 Scattering Invariant Deep Networks for Classification

A typical classification problem would consist of, given one of the written digits shown in Figure 3.5, to classify which number class it belongs to. The problem is not straightforward because there is a huge variability within each class. Therefore it is necessary to extract features from the signal that are representative of the object even if the object is modified in some way (translated, rotated, shifted etc). Classification problems are widely solved with convolutional neural networks, whose operations in the hidden layers are possibly not discernible [62]. In this chapter I will show how the classification problems can be solved with a scattering transform, which is a series of intuitively elegant mathematical operations whose scattering paths resemble the structure of a convolutional network so much so that scattering transforms have been used to develop CNN-like architectures to solve the same classification problems as CNNs with similar and better accuracy results, depending on the problem [63].

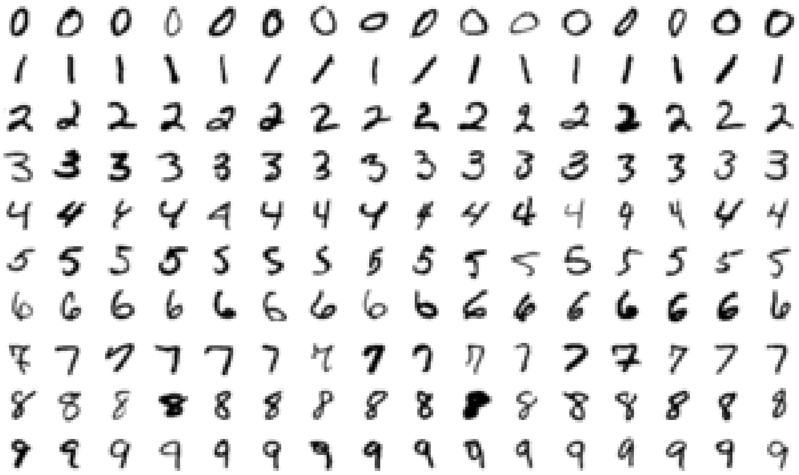


Figure 3.5: This figure shows a sample image from the MNIST test dataset [67]. The MNIST database is commonly used for training and testing in the field of machine learning.

3.2.1 Metric for Classification

As previously mentioned, a classification problem would be, given one of the many variants of the number 0 in the MNIST figure, to train an algorithm to automatically recognize that the number belongs to the class of 0s. This is not a trivial task since the number cannot be classified by its shape since this can be deformed, rotated, rescaled, translated, etc. Therefore the main obstacle in classification problems is the need to extract stable features that represent the signal through appropriate kernels.

In order to classify which class an object (which could be a 1D time signal or a 2D image etc) belongs to, the core issue is to understand how to build up a distance that will reflect the similarities of signals within the same class and that will differentiate objects which do not belong to the same class. There has been extensive research on kernels for classification [64–66] to find the distance between signals to be defined as (3.7),

$$d(f,g) = || \Phi(f) - \Phi(g) || \,. \tag{3.7}$$

Kernels are functions which output the similarities between signals. in order to classify which class an object belongs to, one needs to look at the distance between the signal f and the signal g, but f and g will be represented with a non-linear operator Φ where Φ is the kernel. So the first step of the classification problem is to find a kernel which will extract features that will be distanced from each other in a way so that the classification problem can be solved.

3.2.2 Building an Operator to Extract a Sparse and Stable Representation of a Signal

In order to build algorithms which can classify audio signals, experts have studied the mechanisms of signal processing of the auditory organs [68]. It has been discovered that in order to decipher audio signals, the cochlea, a spiral organ which is part of the inner ear, has detectors all along its spiral shape which behave like linear filters. These linear filters are shifted and dilated with respect to one one another. These are linear bandpass filters called wavelets [61].

In the previous sections we explored how wavelets can be used to extract a sparse representation of the most important features of the signal. In the following sections instead we will explore how wavelets can be used for classification problems, where, contrary to the previous applications, in which wavelets were used to reconstruct signals, now wavelets will be used to construct an invariant operator. This operator will need to use wavelets to eliminate non-essential information.

The first step to extract the important information out of the signal is to make sure that the wavelet transform is a stable operator. This is important because, for example, in certain cases there seems to be a perfect set of coefficients that should be picked in order to classify the signal, but as soon as the system is perturbed, those coefficients don't work anymore.

In the world of machine learning the novelty is that the algorithm should find the non-linear operator to be applied to the signal by itself. In this chapter I am instead proposing to find a non-linear operator to extract features which are modelled by hand. So it might look like we are taking a step back compared to what we could achieve with the state of the art machine learning algorithms, which adapt to the data to find the proper filters. But that is not the case. This is because many problems have the same common denominator: translations, deformations, rotations, scaling. We want to use the wavelet transform to filter our audio signals because it's the same filters that our ears use. So we can now represent the signal as the inner products of a family of wavelets with the signal [58] as shown in the equations below.

$$\langle x, \Psi \rangle = \sum_n x[n]\Psi_\gamma[n]. \tag{3.8}$$

$$\langle x, \Psi \rangle = \int x[t]\Psi_\gamma[t]dt. \tag{3.9}$$

In the equations above x is our signal and Ψ_γ is a family of wavelets dilated and shifted with respect to each other.

We also require that this new representation of the signal is stable. Stability means that the energy of the integral (in the case of a continuous signal, equation 3.9) or the summation (for a discrete signal, equation 3.8) of the inner products between all the chosen wavelets and the signal must be equal to the original signal.

$$|| x ||^2 = \sum_{\gamma} | \langle x, \Psi_{\gamma} \rangle |^2 . \tag{3.10}$$

This stability of the representation of the signal means that if we add another signal ε to our original signal x, $\langle x, \Psi \rangle$ is not going to change significantly, which means we are using an operator which is stable to additive perturbations [54].

So far we have shown that we can represent our signal as $\langle x, \Psi \rangle$, now if we want to reconstruct a sparse representation of our original signal, eliminating what is not useful for classification, given that energy is conserved as shown in equation 3.11, we can use another family of vectors called the dual frame, the process is shown in the equation below.

$$x^{\text{ł}} = \sum_{\gamma} \langle x, \Psi_{\gamma} \rangle \Psi_{\gamma} \tag{3.11}$$

The idea is to eliminate all the coefficients which do not play an important role for classifying the signal. In this way we will obtain a sparse representation of the signal. In order to get the most sparse representation of the signal one must choose a orthogonal basis as the wavelet family of filters, so that the representation of the signal is not redundant at all. This is because the dot product of two vectors, which are orthogonal to each other is 0, thus when such orthogonal wavelet vectors are convolved with the signal, they will not overlap with each other, thus won't return the same coefficients of the signal twice. The final signal representation will then be

$$x^{\text{ł}} = \sum_{\gamma} \alpha_{\gamma} \Psi_{\gamma}. \tag{3.12}$$

The equation above is stating that our signal is now represented by a selected few wavelets of a wavelet family and a subset of coefficients. The method to pick these - redundant sparse coefficients α_{γ} is not a banal one, and furthermore, for variations of the original signal such as translations, rescalings, rotations, deformations, additive noise, the set of sparse coefficients that represent the signal changes if we only use equation 3.13. This is not good, because, yet we have obtained a sparse representation of the signal which is stable, but only in a specific regime (which will be described later). So there is another problem to be solved, namely how to make these sparse coefficients invariant to small deformations, additive noise, rotations, rescaling, translations etc.

So now the goal is to build an operator which will reduce variability within each class without loosing important information about the signal. The operator that allows us to fulfill all of the previously laid down conditions is the scattering transform. This operator allows us to start from signals which are impossible to classify with a simple Euclidean distance, as shown in Figure 3.6 A, to a representation of the signal $\Phi(x)$, where only invariant features are considered. In this way our signal is better divided into neat classes as shown in Figure 3.6 B. Now the euclidean norm of the features of two different objects, mathematically represented as $|| \Phi(Signal1) - \Phi(Signal2) ||$, gives us meaningful information about the similarity of these two signals.

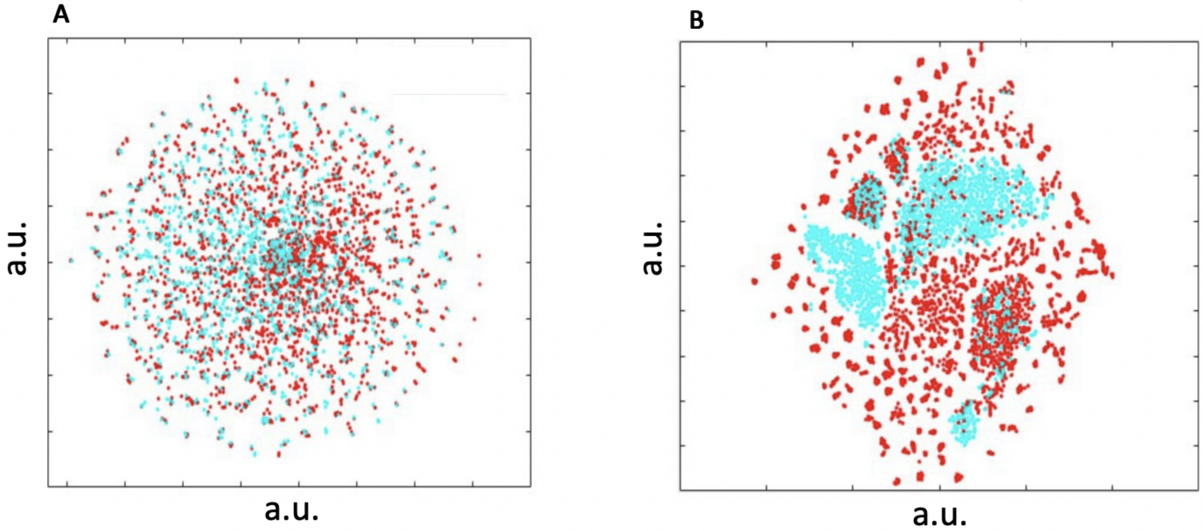


Figure 3.6: Figure 3.6 shows through t-SNE visualization the conceptual effect that the scattering transform has on data belonging to two different classes. Assuming the blue color belongs to the features of a class and the red color to the features of a different class, Figure A is a 2D representation of the signal, while Figure B is a 2D representation of invariant features acquired with the scattering transform.

In order to better visualize this concept, one can look at Figure 3.6. This figure has been obtained by using a t-SNE (t-Distributed Stochastic Neighbor Embedding) algorithm, which is an unsupervised, non-linear technique primarily used for the visualization of high dimensional data [70]. In other words, the t-SNE takes some input data and it allows a visualization of the features in a 2D space, so that it can be easier to check whether features belong to the same class or not. In this case, the t-SNE has been used to visualise the result that the wavelet scattering transform has on a dataset. The t-SNE data visualization in Figure 3.6 A shows the raw data and the t-SNE in Figure 3.6 B shows the effects of the scattering transform on the data, which is now clustered in two groups. In fact, in Figure 3.6, we can see that there are blue and red dots. The blue dots represent the signals of one class, while the red dots represent the signal of another class. In Figure 3.6 A, it is impossible, with a simple Euclidean norm to classify which signals belong to each class because the two classes overlap. In Figure 3.6 B, the scattering transform has been applied to the signals of both classes. That is why the signal is now represented by a set of features $\Phi(signal1)$, $\Phi(signal2)$, which are invariant to translations, deformations, rotations etc. and furthermore the scattering transform reduces the distances between signals of the same class and increases the distance between signals belonging to different classes (a property of it that will be discussed later). Therefore, the output is that the two classes, blue and red are much better separated and now the classification problem can be easily solved by applying an SVM (the theory of the SVM will be discussed later).

3.2.3 Stable Translation Invariants

So far we have seen that in order to solve classification problems we need to find an operator Ψ which is translation, rotation, deformation and additive noise invariant and also that gives a sparse but meaningful representation of the signal. At present we have only shown that wavelets can be part of this operator and have explained why they are the right filters for audio signals and why they provide sparsity. However we have not yet discussed how we can make a translation invariant operator. In this section we will tackle that problem. Let's begin our discussion by understanding how to make an operator which is invariant to translation, the "easiest" (conceptually speaking) variability source. We want to build an operator such that for a translation of a signal x so that x becomes x + c [69],

$$\Phi(x) = \Phi(x_c). \tag{3.13}$$

This problem can be solved, for example, by using the Fourier Transform as operator. This is because if our signal x is translated, only the phase of the Fourier Transform is going to change. In other words, if we have a sinusoidal wave at 300 Hz, for example, and we shift this sinusoidal in the time domain, the frequency content of the sinusoidal doesn't change, and the real part of the FT will still return a peak at 300 Hz. Only the phase of the signal will have changed. So, in order to make a signal translation invariant if shift our signal x by an amount c so that x is now x+c, we could just take the FT of a signal x,

$$X(w) = \int x(t)e^{-iwt}dt. \tag{3.14}$$

and then remove the phase by taking the modulus of the signal

$$X_c(w) = X(c). \tag{3.15}$$

3.2.4 Stable Deformations Invariants

We have just seen that by taking the modulus of the FT we obtain a translation invariant operator. But what about deformations? If we add a small deformation to the signal such that

$$x_{\tau}(t) = (t - t(\tau)). \tag{3.16}$$

it will have a very huge impact at high frequencies as shown in the figure below.

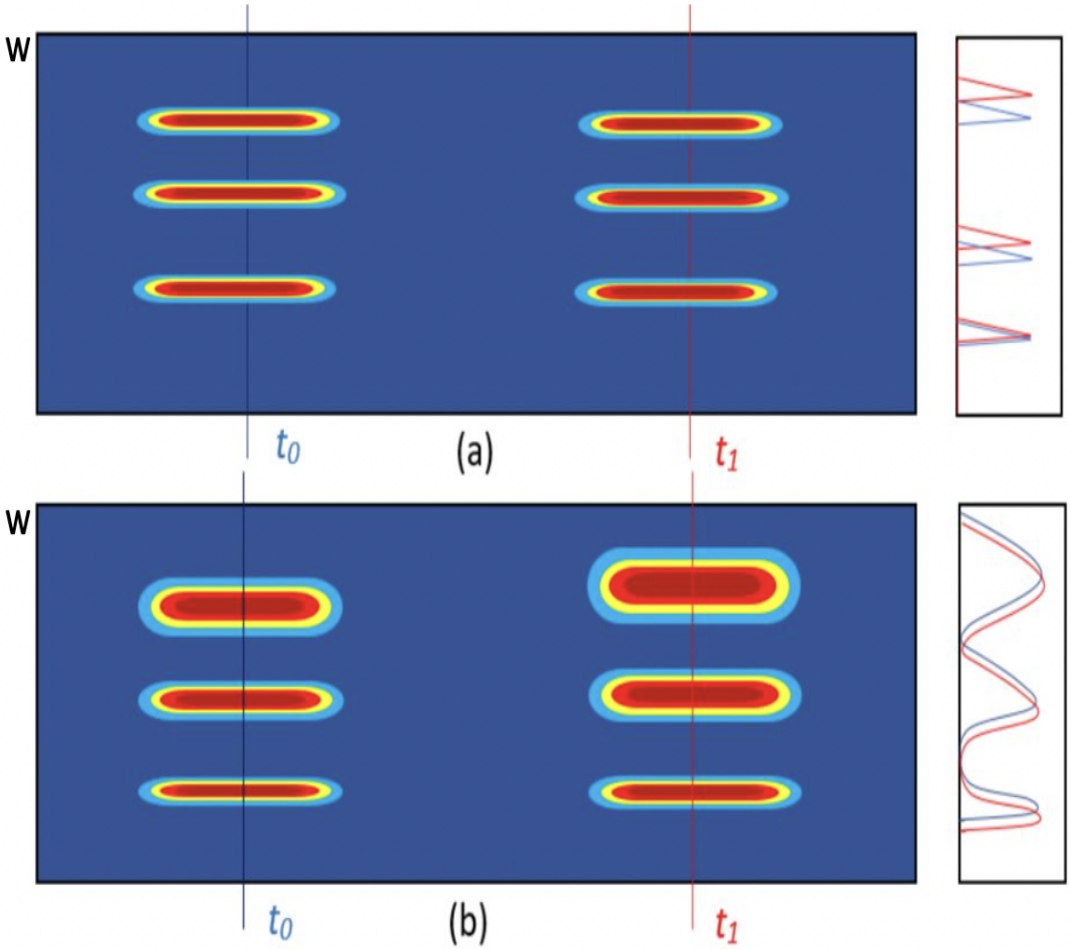


Figure 3.7: Figure 3.7 shows the spectrogram of a signal x(t) and its dilated version $\hat{x}_{\tau}(w)$ and in the rectangle to the right the respective absolute values of the Fourier Transforms for x(t) in blue and $\mid \hat{x}_{\tau}(w) \mid$ in red. It is possible to see that when the signal is dilated, the Fourier Transforms at high frequencies do not match. Figure b) shows the same concept but now the signals have been averaged in time. Now the absolute values of the FT are more aligned.

In Figure3.7 we can see the spectrogram of the FT of a signal x(t) whose FT is $\hat{x}(w)$. And to its right we can see the spectrogam of $\hat{x}_{\tau}(w)$ which is the spectrogram of the FT of Equation 3.16. To the side of the spectrogram, inside the rectangle, we can see the FT of the two signals. The signal x(t) is a musical note with different harmonics and $\hat{x}_{\tau}(w)$ is the same note but the signal has been dilated in time. In the rectangle to the side we can see, drawn in blue, the FT of x(t), while in red the FT of $\hat{x}_\tau(w)$. It can be seen that at low frequencies the two FT almost overlap, but at higher frequencies they do not [69].

So now we are facing a new problem. How can we build an invariant which is stable at high frequencies without loosing the high frequencies information? We have just seen that the modulus of the FT allows us to build a translation invariant, but not a dilation invariant. So we need to refine our invariant with a different operator.

We have previously discussed that wavelet filters are a good choice for audio signals, now We will try to see if the Wavelet Transform can be our new deformation and translation invariant operator.

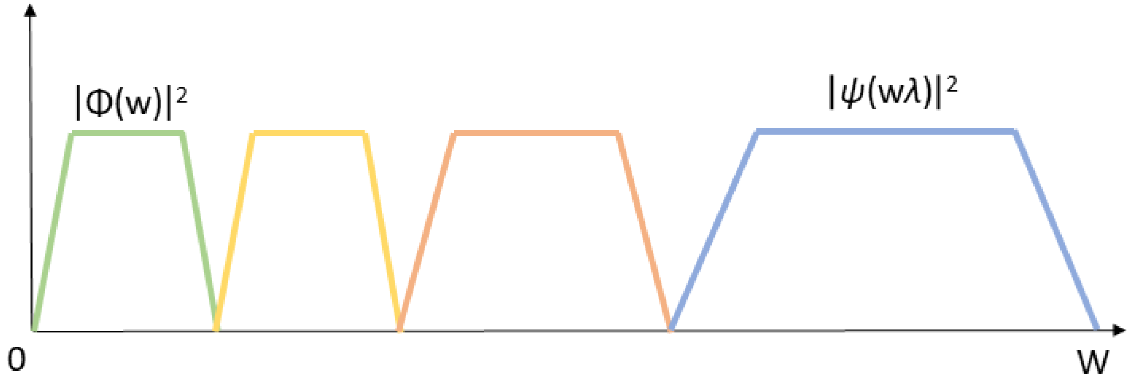


Figure 3.8: Figure 3.8 shows a visual representation of wavelets, which are shifted bandpass filters. At each layer of the scattering transform, one such filterbank of translated wavelets is used to convolve with the signal and give as many outputs as the number of wavelets. In the Figure, the Q factor=3.

In Figure 3.8, the Q stands for Q bandwidth, which is the number of wavelets per octave. The Wavelet Transform is given by the convolution of the signal with all the wavelets such that

$$Wx(t) = (x \circledast \phi(t), x \circledast \Psi_\lambda(t)). \tag{3.17}$$

This equation shows that the WT is frequency domain is a convolution between the FT of the signal $\hat{x}(w)\Psi_\lambda(w)$ plus the convolution of the signal $\hat{x}(w)$ with a low frequency filter $\phi(w)$ which the wavelets don't cover [69]. The convolution can also be described as the inner product of x with all the translations (number given by Q) of the wavelets

$$\langle x(t), \Psi_\lambda(u-t) \rangle. \tag{3.18}$$

So we can say that the coefficients that the WT outputs are the decomposition of the signal in a dictionary composed of all the wavelets $\Psi_\lambda(u-t)$. We are decomposing our signal with all the dilations and all the translation of a family of wavelets. it's a very redundant process. We also want to make sure that the operator is unitary by making sure that

$$|\hat{\Phi}(w)|^2 + \sum_\lambda |\hat{\Psi}_\lambda(w)|^2 = 1. \tag{3.19}$$

In this way we assure that the wavelets cover the frequency domain which ensures each part of the signal can be convolved with the wavelets. This concept can be shown with the equation below where

$$|| Wx ||^2 = || x \circledast \phi ||^2 + \sum_\lambda || x \circledast \Psi_\lambda ||^2 = || x ||^2 \,. \tag{3.20}$$

The equation above says that the energy of the Transform $|| Wx ||^2$ equals the sum of the energy of all the coefficients recovers the energy of the signal $|| x ||^2$. An important property of the wavelets filters is that when you deform a wavelet (by dilation, for example) you are left with a wavelet, which is just dilated [72].

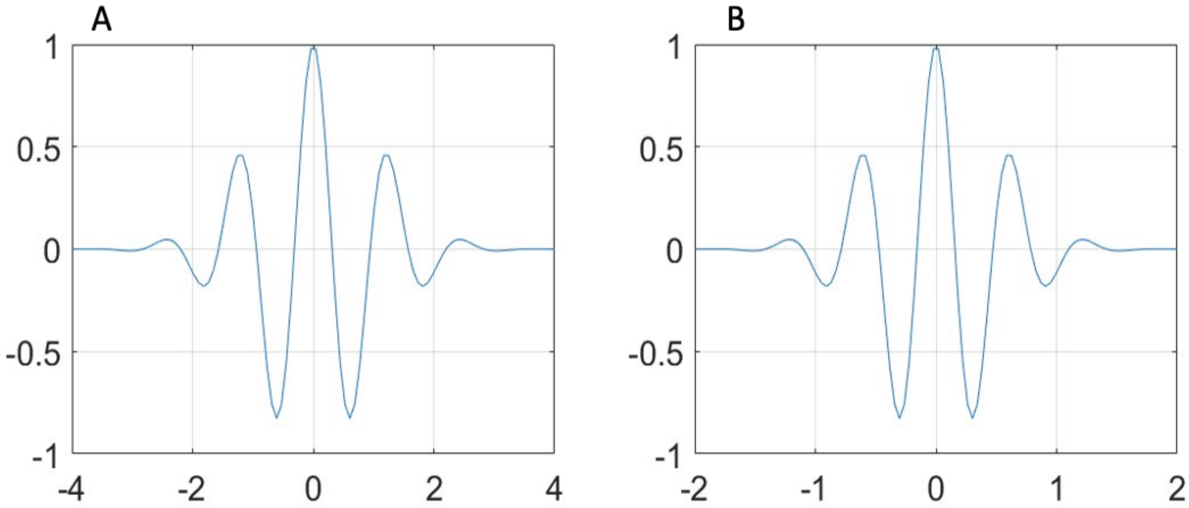


Figure 3.9: Figure 3.9 A shows a visual representation of a Morlet wavelet and 3.9 B the same wavelet but dilated. As discussed, a squeezed wavelets retains the shape of the original and is used to pick up high frequencies features.

3.2.5 Wavelet Scattering Path

As we have seen so far, the WT is deformation invariant. However, unlike the FT, taking the modulus of the WT does not make translation invariant.

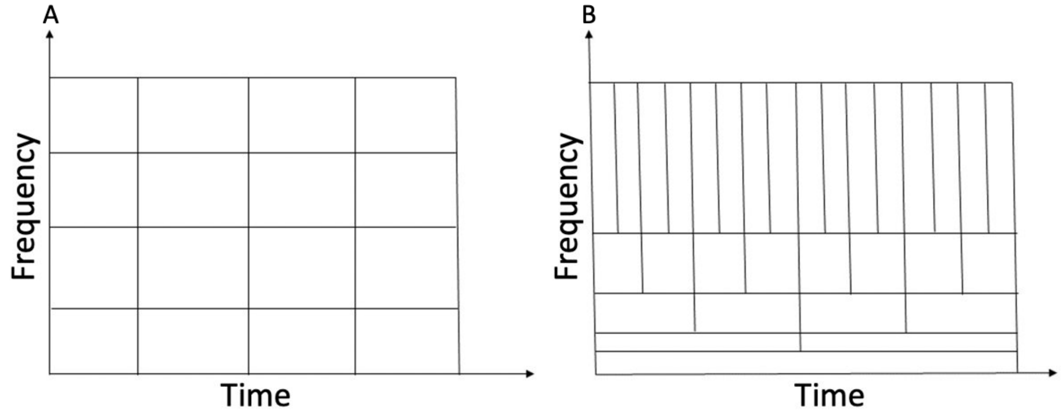


Figure 3.10: Figure A shows how the windowed FT decomposes the signal, which is, by separating it into intervals of the same length. In Figure B we see how the WT decomposes the signal, which is in intervals such that we get high frequency resolution but low time resolution for the low frequency components of the signal and and high time resolution but low frequency resolution for the high frequency components of the signal.

We have already seen that at high frequencies wavelets are very localized in time and at low frequencies they are very localized in frequency. At high frequencies wavelets are very sensitive to translations, so one needs to take the modulus of the Wavelet Transform and then average in the time domain, although this means loosing resolution. But if we average in time, because of the oscillatory nature of the wavelet transform we will get 0 [73].

$$\int (x) \circledast \Psi_\gamma(t) dt = 0. \tag{3.21}$$

So we now need to find a translation invariant which is nonlinear such that

$$\int M((x) \circledast \Psi_\gamma(t)) dt. \tag{3.22}$$

where M is the non-linear mask.

We also want our scattering invariant to be stable under additive perturbations. Which means that if some noise is added to the signal, our operator will still be able to extract very similar features as the noise wasn't there. So we need to impose that

$$|| Mh || = || h || . \tag{3.23}$$

So we want an operator for which the 2-norm is preserved under additive perturbations. So we apply the modulus whose equation is

$$\mid\mid h \mid\mid = (\int_{-\infty}^{+\infty} h_{real}(t)^2 + h_{imaginary}(t)dt)^{1/2}. \tag{3.24}$$

The modulus computes the envelope of the signal, also called the analytical part of the signal, we have gotten rid of the phase and now we have a positive signal which is translation and dilation invariant.

$$\mid\mid Mg - Mh \mid\mid = \mid\mid g - h \mid\mid . \tag{3.25}$$

So we take the modulus of the Wavelet Transform as our new operator and then we average with a low pass filter.

$$\mid \Psi_{\lambda 1} \circledast x \mid . \tag{3.26}$$

But now we have lost a lot of information by averaging. In order to recover the lost information we will need to build deep layers, and here comes the similarity in the structure of the wavelet scattering transform and CNNs. So now we take the signal $\mid \Psi_{\lambda 1} \circledast x \mid$ and we compute its wavelet transform again

$$\mid \Psi_{\lambda 1} \circledast x \mid \circledast \Psi_{\lambda 1}. \tag{3.27}$$

But these coefficients are not translation invariant so we need to take the modulus and average once again

$$\mid\mid \Psi_{\lambda 1} \circledast x \mid \circledast \Psi_{\lambda 1} \mid \circledast \phi. \tag{3.28}$$

To note is that $\lambda 1$ refers to all the wavelets in a family of wavelets, which in a scattering transform can be specified by the value Q, and the same is true for $\lambda 2$. So, when we build a wavelet scattering transform, we can specify how many wavelets we want per scattering path [74].

The modulus is a contracting operator so, given 2 complex numbers a and b,

$$\mid\mid a \mid - \mid b \mid\mid \leq \mid a - b \mid . \tag{3.29}$$

which means that the modulus operator brings signals closer together.

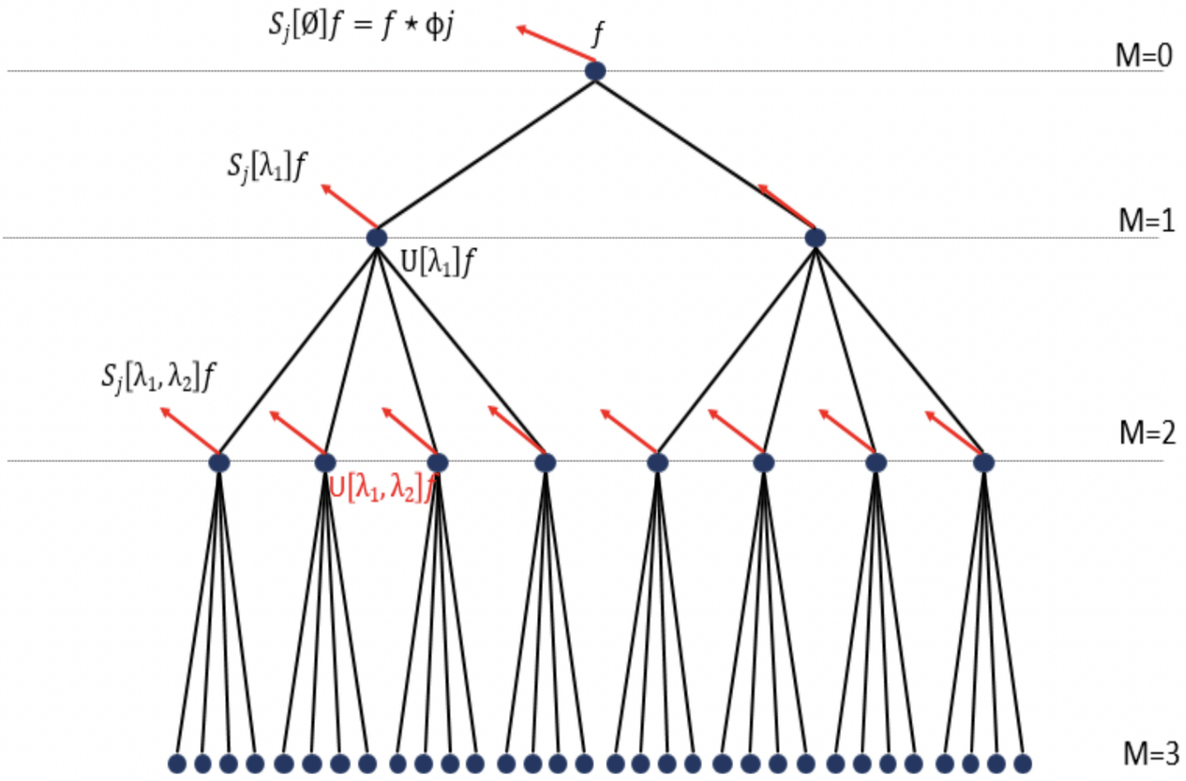

$S_j[\emptyset]f = f \star \phi j$
f
M=0
$S_j[\lambda_1]f$
M=1
$U[\lambda_1]f$
$S_j[\lambda_1, \lambda_2]f$
M=2
$U[\lambda_1, \lambda_2]f$
M=3

Figure 3.11: Figure 3.11 shows the architecture of a scattering network with Q (of wavelets per path as described in a previous section) = [4,4,4], which means there are 4 wavelets per scattering path, thus 4 nodes per layer. At m=0 we have our input data, at m=1 we have in blue the nodes, where each node contains the features of the signal being convolved with one of the wavelets in the filterbank and then the modulus is taken. While the black arrows coming out of each node are the outputs, which are convolved with a low pass filter.

In Figure 3.11 at m = 0, we get the first output of the scattering transform, $f \circledast \phi$, which as we saw before, is the convolution of our signal f with a low pass filter. These are our first scattering coefficients and now they exit the network, therefore taking some energy away from the network. This is important to notice because the sum of all the coefficients in the network =1 as we saw before. This means that since some of the coefficients exit the network at each layer, as we move to deeper and deeper layers, the energy at each level keep decreasing until it reaches 0. We can see that the first difference with a standard neural network is that the output is not going to be at the end, but its going to be at each layer. The second difference from a standard neural networks is that the filters are not learned by the network, we are using wavelets as filters.

One important thing to notice is that this structure shown in Figure 3.11 which has 4 nodes at each layer and 3 total layers, can obviously be changed to fit individual purposes. The way this can be changed is by choosing how many wavelets one wants to use per scattering path (layer). In the Figure 3.11 there are 4 wavelets per node, but one can choose the number of wavelets per node. And one can also choose how many layers of the scattering network to have.

At the output of the first layer m=1, we will obtain 4 new sets of coefficients, a $U[\lambda_1]f$ per each node, which means that at each node we will have a scattering transform coefficients given by the formula $\mid x \circledast \Psi_{\lambda 1} \mid$ where λ_1 stands for all the wavelets of the first scattering path, which means, since the structure has 4 wavelets in the first scattering path, which we can denote as $\Lambda 1, \Lambda 2, \Lambda_3, \Lambda_4$ that the 4 sets of scattering coefficients of the first layer are $\mid x \circledast \Psi_{\Lambda 1} \mid \circledast \Phi$, $\mid x \circledast \Psi_{\Lambda 2} \mid \circledast \Phi$, $\mid x \circledast \Psi_{\Lambda 3} \mid \circledast \Phi$, $\mid x \circledast \Psi_{\Lambda 4} \mid \circledast \Phi$, which are the equations corresponding to the outputs of the black arrows shown in Figure 3.11.

The blue nodes of the first layer have as outputs $\mid x \circledast \Psi_{\Lambda 1} \mid$, $\mid x \circledast \Psi_{\Lambda 2} \mid$, $\mid x \circledast \Psi_{\Lambda 3} \mid$, $\mid x \circledast \Psi_{\Lambda 4} \mid$ and these are fed into the second layer.

The outputs of the scattering transform at the second layer are given by $\mid\mid x \circledast \Psi_{\lambda 1} \mid \circledast \Psi_{\lambda 2} \mid \circledast \Phi$. The Q factor of the second node is 4, so there are 4 wavelets per each node of the first layer, thus we will obtain 16 data sets of scattering coefficients in the second node. For each of the nodes in the second layer there is an output which are the scattering coefficients for that node (there are 16 nodes in the second layer for the architecture shown in Figure 3.11) and those coefficients are given by the equation $\mid\mid x \circledast \Psi_{\lambda 1} \mid \circledast \Psi_{\lambda 2} \mid \Phi$, but, as shown before for the first layer, there are 4 wavelets each node so the outputs of the second layer will be: $\mid\mid x \circledast \Psi_{\Lambda 1} \mid \circledast \Psi_{\kappa 1} \mid \circledast \Phi$, $\mid\mid x \circledast \Psi_{\Lambda 1} \mid \circledast \Psi_{\kappa 2} \mid \circledast \Phi$, $\mid\mid x \circledast \Psi_{\Lambda 1} \mid \circledast \Psi_{\kappa 3} \mid \circledast \Phi$, $\mid\mid x \circledast \Psi_{\Lambda 1} \mid \circledast \Psi_{\kappa 4} \mid \circledast \Phi$ where κ_1, κ_2, κ_3, κ_4 are the 4 wavelets in the second scattering path which are used to convolve with the inputs deriving from the first 4 nodes. So, each output of the first node is convolved with these 4 wavelets, resulting in 16 new outputs. The equations above showed the outputs of the second layer, but what is instead fed into the nodes of the third layer is $\mid\mid x \circledast \Psi_{\lambda 1} \mid \circledast \Psi_{\lambda 2} \mid$.

As we have discussed previously, the full energy of the signal is conserved, which means that if we sum the coefficients obtained from the last layer plus all the previous outputs we get back the energy of the original signal. This also implies that the energy of the deeper layers reaches zero. Furthermore, it is very important to notice that out of all the outputs nodes, the majority of them will output 0. That is because if we look at the wavelets in each scattering path, these will be bandpass that in increasing order will be bandpass for low frequencies up to high frequencies. Since at each iteration we are averaging, therefore taking the envelope of the signal, our information lies in the low frequencies. So, for all the wavelets in a a scattering path of each layer, only a few wavelets will bandpass signal that contains information. Because we know which paths give us information from the signal and which don't for the reason just explained. At the end, the number of nodes whose outputs are non zero is in $Nlog(N)$.

To summarize, these are the outputs of the scattering transform at each layer

$$x \circledast \Phi. \tag{3.30}$$

$$\mid x \circledast \Psi_{\gamma 1} \mid \circledast \Phi. \tag{3.31}$$

$$\mid\mid x \circledast \Psi_{\gamma 1} \mid \circledast \Psi_{\gamma 2} \mid \Phi. \tag{3.32}$$

$$\mid\mid x \circledast \Psi_{\gamma 1} \mid \circledast \Psi_{\gamma 2} \mid \circledast \Psi_{\gamma 3} \mid \Phi. \tag{3.33}$$

3.2.6 Information Recovered at Different Layers

The information retrieved from the first layer of the scattering transform is easy to understand: we are just recovering frequencies with shifted band-pass filters. At the second layer things start to get more complicated. To understand what is the information recovered by the scattering transform at each layer let's look at Figure 3.12 where we can see the scalogram of a signal, in the top layer the signals are distinguished from each other in the horizontal axis while on the vertical axis we have the higher harmonics of the same signal which produces the repetitive

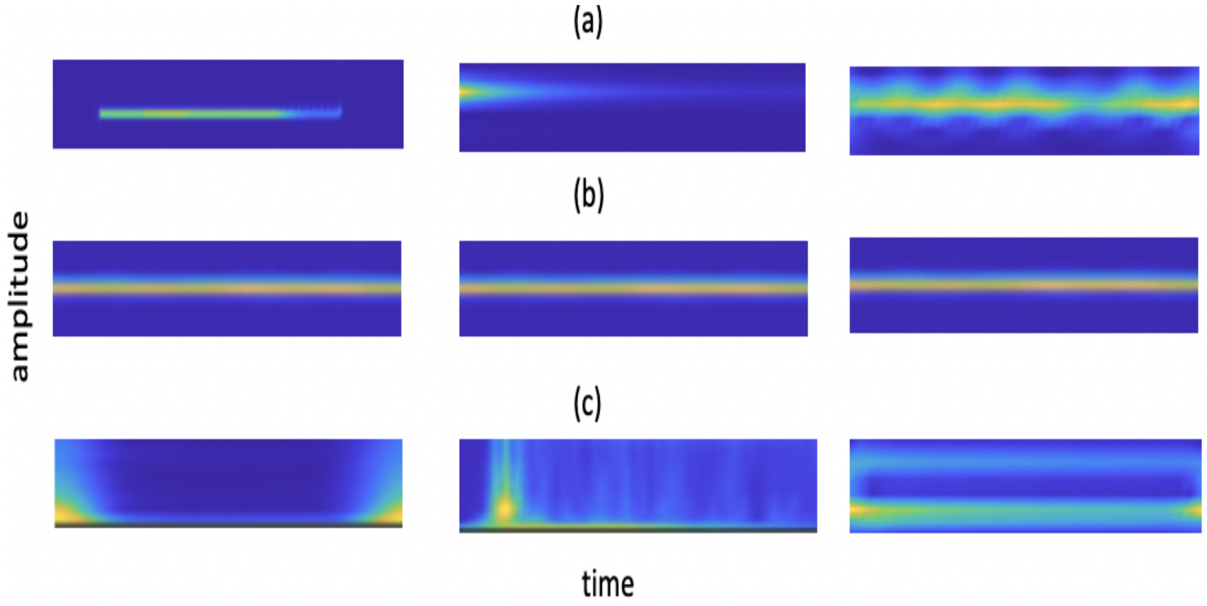


Figure 3.12: Shows the information obtained by the wavelet scattering transform at different layers [69]. In the top part we can see the spectrogram of 3 signals, where the first three signals and they all have a pitch of 600 Hz with different attacks. In the second layer we see the pitch content. While in the third layer we see the attach content.

lines. The 3 signals of the first layer all have the same pitch ε=600 Hz. However the first three signals all have a different amplitude modulation. The first signal has a smooth attack, then a sharp attack, then a tremolo of frequency η.

In Figure 3.12 b we see what we would see at the first layer of the scattering transform (see Eq.3.31: the average of the signal with a low pass filter, the high frequency information has been lost and the three signals are now indistinguishable the one from the other.

In Figure 3.12 c we see the information recovered at the second layer of the scattering transform. The information recovered in this layer is the information of the attack. The first signal, since it has a very smooth attack (n.b. in music theory, attack refers to the time taken by the sound to begin from 0 amplitude to reach its peak in amplitude [76]), has a very low frequency spectrum of the attack. Then for the second signal, since we can see it has a very sharp attach and then decays smoothly, we can see that the attack information has high frequencies at the beginning, as expected and then as it decays smoothly in time it presents the same low frequencies for the attack as the first signal did. For the third signal, which is a tremolo (trembling effect), we have a sine wave as attack, which is exactly the modulation of the signal.

Let's now look at what is obtained at the second layer of the scattering network from a mathematical point of view. Let's suppose we have a simple signal which is a summation of sine waves

$$x(t) = \sum_{m} cos(w_m t). \tag{3.34}$$

This signal in the Fourier domain just looks like different delta functions at the different frequencies w_m. When the Wavelet Transform is computed, each wavelet isolates a frequency band (because we have seen wavelets are just bandpass filters with particular shapes). Then for each convolution of our signal with one of the wavelets in the scattering path we compute the modulus square (remember, we want to look at the second layer, whose output is given by the equation 3.32 and take the average. The modulus square of the wavelet transform of the signal is

$$\mid x \circledast \Psi_{\lambda}(t) \mid^2 = c + \sum cos(w_m - w_m^l)t. \tag{3.35}$$

this equation shows that in the output of the second layer we get a first term that shows the energy plus a term which shows that we get the difference between the frequencies of the signals, which are the interferences of the signal. In the third layer we measure the interferences of the interferences.

The scattering transform operator $|| Sx ||^2$ which follows the following properties:

$$|| Sx - Sy || \leq || x - y || . \tag{3.36}$$

Which means that if there is a small additive noise (or other sources of difference between signals belonging to the same class described throughout this chapter), the scattering transform makes it so that the signal is still placed within the same class.

The second property of the scattering transform operator is that all the energy is conserved.

$$|| Sx ||^2 = || x ||^2 . \tag{3.37}$$

The operator is also stable to deformations as we have seen and we can write that as

$$|| Sx - Sx_{deformed} ||^2 = c * sup \mid \nabla t(\tau) \mid| x || . \tag{3.38}$$

Which tells us that if the signal 2 is deformed compared to signal 1, the distance between the two signals is going to be of the order of the deformation.

3.2.7 Comparison with Neural Networks

In this brief section we will discuss how the wavelet scattering transform relates to a deep neural network.

At the beginning of this chapter we have stated that in signal classification problems we are given signals and labels and we need to construct algorithms that given a signal, regardless of the conditions of the signals, can identify which class it belongs to. The problem is that each signal is a function of a huge amount of variables.

It was in 2010 that Yann LeCun obtained promising results by implementing an old concept: neural networks [77].

In the neural network architecture devised by Yann LeCun an image is convolved with filters in the same way in the wavelet scattering network a 1d signal or a 2d image is convolved with wavelet filters. The next step in Lecun's network is subsampling, which also took place in the wavelet scattering transform. Then, in Lecun's network a non-linearity is added by taking the absolute value of the coefficients. In the case of the wavelet scattering transform the non-linearity came from taking the modulus of the coefficients. The process is then repeated.

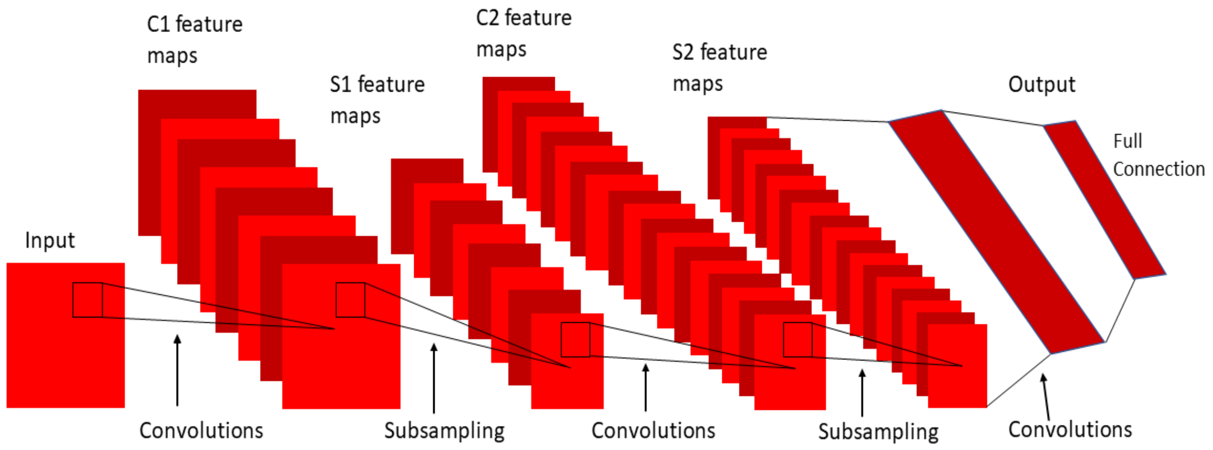


Figure 3.13: An example of a typical NN architecture. An input image is convolved with a non linear filter which slides across it. The result, is subsampled and the features are used as inputs for the next layer.

The difference between the neural network of LeCun and neural networks in general and the wavelet scattering transform is that the filters, instead of knowing them a priori (wavelets) as in the scattering transform, they are learned by the neural network. In this way they adapt to each problem. In order to learn the filters we put a constraint such that if we insert an input for which we know the class it belongs to, the output should match that. So the filters will need to minimize the error between inputs and outputs. The astonishing thing is that the first layer of these networks are wavelet filters [78].

3.3 Machine Learning Classification with Kernels

What we have talked about recurrently but not in depth in this chapter is the idea of kernels, which are machine learning algorithms which output the similarities between signals by replacing $\langle x, x^1 \rangle$ with $k(x, x^1)$ where $k(x, x^1) = \langle \Phi(x), \Phi(x^1) \rangle$ where Φ is an operator which maps x into a feature space [79].

So far we have explored the wavelet scattering transform as our kernel. In this section we will explore other classification kernels. But the main idea that we can summarize from the wavelet scattering transform and generalize for all classification kernels is that they are supposed to extract features which are similar for signals belonging to the same class even when these signals are slightly different the ones from the other.

3.3.1 Convolutional Neural Neworks as kernels

Another set of classification algorithms which extract relevant features which are invariant to irrelevant variations of the signal are convolutional neural networks. We have introduced the wavelet scattering transform as an operator which can extract key information from images and sound signals with a set of fixed filters and we have also already established the difference between the wavelet scattering transform and a neural network. Now we explore CNNs, algorithms whose filters are learned online, thus will be different for each problem.

Convolutional Neural Networks (CNNs) are a biologically-inspired trainable architecture that can learn invariant features. This is quite a remarkable quality which make these networks suited to solve a vast variety of problems. The architecture of a CNN is structures as follows: in each layer of the architecture the inputs and outputs are features. Each stage is composed of three layers: a filter bank layer, a non-linearity layer, and a feature pooling layer. A typical ConvNet is composed of one, two or three such 3-layer stages, followed by a classification module. While we have talked about kernels as operators which act on our input data for SVMs and wavelet scattering transforms, a kernel for a CNN is a filter which the algorithm automatically learns and acts upon the data through convolution (note all the similarities with the wavelet scattering transform) through

$$y_j = b_j + \sum_i k_i j \circledast x_i \tag{3.39}$$

where y_i is the output feature map, x_i the input data, b is a trainable parameter and k is also trainable and is the kernel we just talked about (a kernel is a filter for CNNs). The next layer is comprised by a nonlinear operator. The final layer is called the pooling layer. At this stage an average of the features is computed, which reduces the resolution thus making it robust to small variations.

For sound signals, the sound will be represented in image form in order to be classified by a CNN through a spectrogram or scalogram or a time trace of amplitude vs time. Then a sliding window shown in Figure 3.14 in pink takes the amplitude values which will be connected to one node of the first layer according to a weighted average. Each pixel in the window will have a different weight. We can establish how many features we want to detect at each layer.

3.4 Conclusion

In this chapter we have seen how we can use wavelet filters in order to denoise signal. We have also described an interesting operator: the wavelet scattering transform. This operator

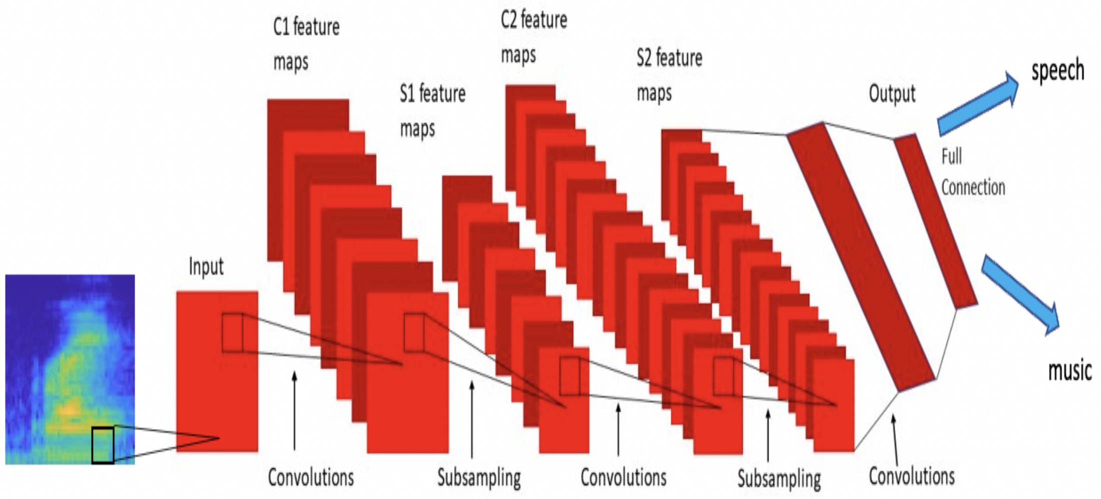


Figure 3.14: The architecture of a convolutional neural network [82].

can extract stable features out of a signal so that classification algorithms will obtain an higher accuracy. In Chapter 6 and 7 we will use these properties of wavelets to firstly denoise and then perform feature extraction to classify heart health and perform biometric authentication.

Chapter 4

An Introduction to Machine Learning

From the previous chapter we have seen a method to extract important features from the data. Then it is time to use models to make predictions from the data.

Machine Learning is a branch of artificial intelligence which allows us to make predictions based on available data. In Machine Learning, instead of giving a program a set of instructions on how to accomplish a task, we write programs to analyze the data and decide on its own how to use it to accomplish a task [85]. Although the goal of machine learning is for the algorithm to make decisions by itself, humans still play a role. A human needs to acquire and organize data, perform feature extraction and feature selection, select a learning algorithm and hypertune the parameters, apply the algorithm to the data, validate the results, check whether they are good or not, check for biases etc.

In chapter 6 and 7 we will apply machine learning algorithms to the heart valve sound data that we acquired as described in chapter 5. The machine learning algorithms will be used to automatically recognize a person from its heart valve sounds and whether that person's heart is healthy or not. This chapter will therefore give a brief introduction on machine learning algorithms. It will start by delving into linear models, which work well with small training data sets, as we have, and then it will veer into deep networks. It will also talk about RNN (recurrent Neural Networks) which, as it will be discussed later, are well suited for our type of data which is sequential.

4.1 Classes of Problems

There are different kind of classes of problems in machine learning. Below we will illustrate 4 standard classes.

4.1.1 Supervised Learning

In supervised learning, the system is given inputs and told which specific outputs should be associated with them. This means that the data set is organized in pairs. To each input x there will be an output y. Once the machine has learned a model to connect the x values to the y values, then it will be able to predict the y values given random x values. This kind of problem is called a classification problem, which is a type of machine learning problem.

Supervised learning can be divided in two branches

- Classification:
  In this kind of problem the algorithm needs to classify the data into classes.
- Regression:
  The algorithm needs to predict values of a continuous variable.

In Supervised learning at first we need to make a hypothesis, which is a first guess of the relationship between the inputs ($\vec{x}$) and the outputs ($\vec{y}$).

$$y = h(\vec{x}, \vec{\Theta}) \tag{4.1}$$

where h is the hypothesis function we want to find and Θ are a set of parameters which will determine which h function is the one which yields the best relationship between x and y.

In order to check whether our initial hypothesis is correct, we will use a loss function, which will compare the predicted values (the values which the function guessed) to the true values (the correct output values). The goal is for the loss function to have a very small value when making predictions on unseen data. This means that the hypothesis needs to be general enough to not only make good predictions on data already seen, but also new unseen data. In other words, a loss function maps decisions to their associated costs. The loss function finds the loss associated with every training sample. Taking the average of the outputs of the loss functions over all the training samples gives the cost function [86]. However to start we need to pivot our problem into finding a hypothesis function which has a small loss on the training data.

$$\xi_n = \frac{1}{n}\sum_{i=1}^{n} L(h(\vec{x}), y) \tag{4.2}$$

where $h(\vec{x})$ are the predictions and y the true values. This equation measures the training and testing sets errors [87].

4.1.2 Unsupervised Learning

In unsupervised learning the algorithm is provided with unlabelled data (data without the answer paired to it), thus the algorithm must find correlations between the input data points.

There are different kinds of unsupervised learning methods.

- Density Estimation:
  the goal is to calculate the probability of outcome of a random variable in a sample [88].
- Clustering:
  involves partitioning data sets into subsets, names clusters [89].
- Dimensionality reduction:
  the goal is to reduce the dimensionality of the data into [90]. The number of input variables or features is referred to as its dimensionality. Dimensionality reduction aims to reduce the number of input features.

4.1.3 Reinforcement Learning

Reinforcement learning is a subfield of machine learning which is essential in problems where the environment is constantly changing and each change could bring infinite possible new configurations. The machine needs to make a decision based on its own internal state and the changing environment's state to maximize the predefined goal [87].

4.1.4 Sequence Learning

Networks such as CNN (convolutional Neural Networks) and DNN (Deep Neural Networks) cannot deal with time series data. Therefore for problems which involve sequential data, such as text, audio or video, recurrent neural networks are used. These networks can update the current state based, not only on the current input like the before mentioned networks do, but also on previous states. The LSTM (Long Short Term Memory) network can even handle long term dependencies of the data [91].

4.2 Supervised Learning

Now we will delve into some examples of supervised learning models.

4.2.1 Linear Classifiers

Once we have chosen a problem class (which were discussed in 4.1), we can choose an algorithm. Below there will be a discussion on various algorithms.

Linear Classifiers

A linear binary classifier has an equation of the form

$$h(\vec{x},\vec{\Theta},\Theta_0) = sign(\overrightarrow{\Theta^T}\vec{x}+\Theta_0)\begin{cases} +1 \text{ ,if } \overrightarrow{\Theta^T}\vec{x}+\Theta_0 > 0 \\ -1 \end{cases}$$

This will produce a line which divides the space. Any new unseen x-value can be plugged into sign ($\Theta^T x$ + Θ_0) to then check on which side of the line it resides in. However usually we will extract features from our data, so that we will work with an operator Ψ to extract features. Furthermore, for the binary classifier, the error is defined as

$$\xi_n(h) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases} +1 \text{ ,if } h(\overrightarrow{x^i}) \neq y^i \\ 0 \end{cases}$$

A hypothesis is a set of possible classifiers and a learning algorithm is a procedure which takes a data input and returns an output. Linear classifiers separate data using a hyperplane (which means a plane in more than 2 dimensions) and because of this they can only be used to classify data that is linearly separable. Linear separability means that the data must satisfy this condition

$$y^i(\overrightarrow{\Theta^T}\overrightarrow{x^i}) > 0 \tag{4.3}$$

Which means that the predicted value and the true value must be of the same sign. In order to use a linear classifier on non-linearly separable data, the data itself needs to be made linearly separable by extracting features through appropriate operators [92–94]. In fact, when the data is linearly separable, the separating hyperplane with maximal margin can be constructed in original input space, but when the data is linearly non-separating, the input vectors need to be mapped into a high dimensional feature space through some kernel functions. Then in this high dimensional feature space an optimal separating hyperplane can be constructed [95].

Another concept for linear classifiers is the margin. This is the distance between each data point and the hyperplane. The equation of this distance is shown below.

$$y\frac{\overrightarrow{\Theta^T}\vec{x}}{|| \Theta ||} \tag{4.4}$$

We will now briefly discuss some linear classifier algorithms.

Perceptron Algorithm

The perceptron algorithm was developed by Rosenblatt in 1957 [96]. The perceptron works by telling which side of the hyperplane the data lies in. The first step of this algorithm consists in taking the dot product of $\vec{x}$ (our input data vector) with $\vec{\Theta}$ which we have defined before as being a set of parameters. Usually this set of parameters which we have defined as $\vec{\Theta}$ is also defined as $\vec{W}$ for weights.
The equation of the hyperplane would be

$$\vec{x}\overrightarrow{W^T}+b \tag{4.5}$$

where $\overrightarrow{xW^T}$ would be the line and b stands for bias and it tells how far from the origin the line would be located.

To train the model we will check if our predictions match the labels of the training data. In order to do so we will use the equation

$$\begin{gathered}\text{if}\\ y(\vec{x}\overrightarrow{W^T}+b)\leq 0\\ \text{adjusts weight and bias}\end{gathered} \tag{4.6}$$

This equation shows that if our predicted y values and the true y values do not match then the biases and weights must be adjusted because it means the two values are on different sides of the hyperplane. The next step after the calculated weights produce an error which is acceptable is to validate the algorithm on unseen data, usually this data is a subset of the original set.

The loss function of the perceptron is

$$\xi_n(h)=\frac{1}{n}\sum_{i=1}^{n}\begin{cases}+1 \text{ ,if } h(\overrightarrow{x^i})\neq y^i\\ 0\end{cases} \tag{4.7}$$

where h() is the hypothesis. And if the data is linearly separable the loss = 0, otherwise the data isn't linearly classifiable and the perceptron does not find a solution.

Figure 4.1 shows the activation function of the perceptron. It is -1 if the data point lies on the opposite side of the predicted value and +1 if it is on the same side [103].

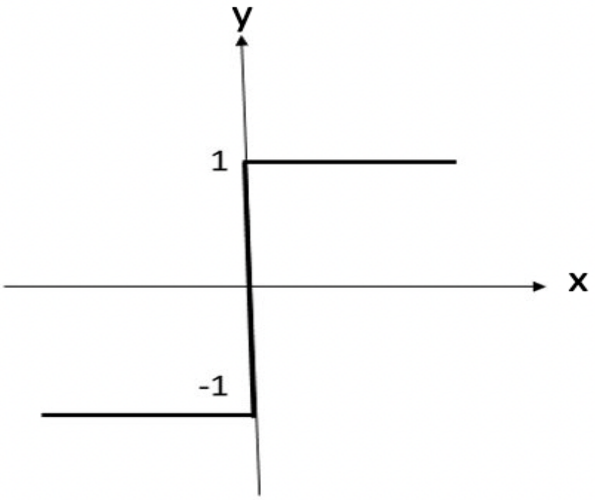


Figure 4.1: shows the activation function of the perceptron, a sign function, which extracts the sign of a real number.

Logistic Regression

In Logistic Regression it is possible to start considering machine learning problems as optimization problems. We now want to find parameters that minimise a loss function. We will also include a regularization term $R(\Theta)$ multiplied by a constant λ in the equation because this will ensure we are not overfitting to the training data. The function that we want to minimise is

$$J(\vec{\Theta}) = \frac{1}{n}\sum_{i=1}^{n}(Loss(h(\vec{x^i}), \vec{\theta}), y^i) + \lambda R(\vec{\Theta}) \tag{4.8}$$

Logistic regression in a linear classifier with a hypothesis given by

$$h(\vec{x}) = g(\overrightarrow{\Theta^T}\vec{x} + \Theta_0) \tag{4.9}$$

where

$$g(\vec{z}) = \frac{1}{1+e^{-\vec{z}}} \tag{4.10}$$

this is called a Sigmoid or Logistic function. Logistic Regression will output a number between

0 and 1 and this number corresponds to the probability y being = 1 or y being = 0 depending on how probable the outcome is. Figure 4.2 shows the sigmoid function

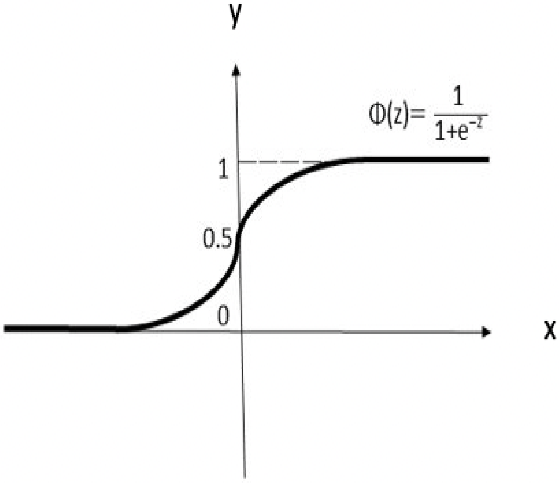


Figure 4.2: shows the activation function of the logistic regression classifier, a sigmoid function, which gives a probability between 0 and 1 of the predicted value corresponding to the true value.

This concept, namely that the output of this new hypothesis can be any value between 0 and 1, can be expressed as

$$P(y=1 \mid \vec{x};\vec{w}) = h(\vec{w},\vec{x}) \tag{4.11}$$

and

$$P(y=0 \mid \vec{x};\vec{w}) = 1-h(\vec{w},\vec{x}) \tag{4.12}$$

and this can be rewritten as

$$P(y \mid \vec{x};\vec{w}) = h(\vec{w},\vec{x})^{y} + (1-h(\vec{w},\vec{x}))^{1-y} \tag{4.13}$$

Now, in order to obtain parameters which will yield the maximum amount of correct predictions we can maximize the function

$$likeliwood(\vec{w}) = \sum_{m}^{i=1} y_i log(h(\vec{w},\vec{x_i})) * (1-y_i) log(1-h(\vec{w},\vec{x_i})) \tag{4.14}$$

In this case we are saying that the hypothesis function h(x) will take a value between 0 and 1, (the prediction values are only -1,1) and the prediction of a value being 1 will lie higher up in the sigmoid function, to a value close to 1. On the other hand, a prediction of -1 will output a value close to 0, so 1 minus a small value will return a value close to 1, that is why the goal is to maximize the likeliwood function.

We have introduced an interesting concept: machine learning algorithms working as an optimizer to minimise or maximise a loss function. The perceptron can only minimise a loss function if the loss is 0, which means if the data is linearly separable.
Our new loss function can be written as

$$(Loss(h(\overrightarrow{x^i}), \theta) = \begin{cases} 0 \text{ ,if } h(\overrightarrow{x^i}, \overrightarrow{\Theta}) = y^i \\ 1 \end{cases}$$

So we have shown from Figure 4.2 that the output can take any value between 0 and 1. However a classifier needs to give a discrete output, either -1 or 1. So we make a prediction threshold, which means that if

$$g(\overrightarrow{\Theta^T} x + \Theta_0) > 0.5 \tag{4.15}$$

then the classifier should yield +1 while if

$$g(\overrightarrow{\Theta^T}\vec{x} + \Theta_0) < 0.5 \tag{4.16}$$

then it should yield -1. The threshold 0.5 is arbitrary and it needs to be chosen in an appropriate manner relevant to the nature of the problem at hand.

Now in order to find the Θ value, one uses an optimiser algorithm: the gradient descent. The gradient descent works firstly by making a guess which will have an x and y value. This concept can be seen in Figure 4.3 B, where a first point is guessed inside a local minima and then the derivative of the function at that location is calculated. The guess will then be a point in the opposite direction in respect to the sign of the derivative. This is called the 1D case because the only parameter to be discovered in the Θ direction (x-axis) [103, 104].

The gradient descent equation is given by

$$\overrightarrow{\Theta^t} = \overrightarrow{\Theta^{t-1}} - nf'(\overrightarrow{\Theta}^{t-1}), \tag{4.17}$$

the algorithm will iterate until the tolerance value is reached which is given by

$$\mid f(\overrightarrow{\Theta}^t) - f(\overrightarrow{\Theta}^{t-1}) \mid < tolerance. \tag{4.18}$$

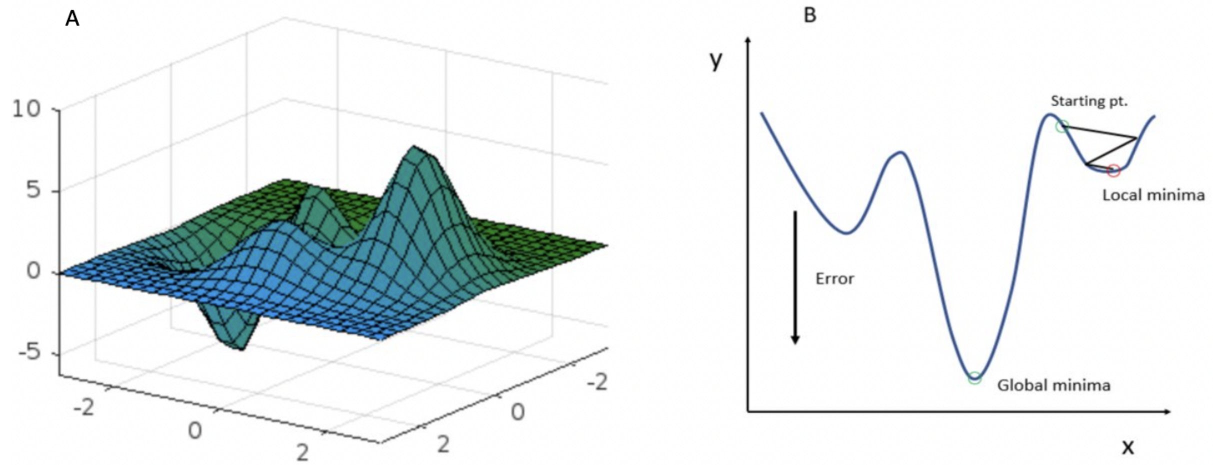


Figure 4.3: A shows the gradient descent in a 3D perspective, with global and local minima while 4.3 B shows a 2D representation of what the gradient descent algorithm does. The algorithm starts by making a guess and the prediction lies somewhere on the x axis and then the guess is improved and gets closer and closer to the local or global minima.

4.3 Feature Representation

So far we have discussed algorithms which can only classify data which is linearly separable (we have not introduced the concept of non-linear classifiers yet). This means that these algorithms per se can classify only a very specific class of problems. So the goal will be to transform the data in a way such that it will be linearly separable in a different space. In order to map the data into a higher dimensional space we will use an operator on our features. If we want to expand our features with a polynomial basis we will use such operator function on our data

$$\begin{aligned} h_{i,1}x_i &= x_i \\ h_{i,2}x_1 &= x_1^2 \\ h_{i,k}x_1 &= x_1^k \end{aligned} \tag{4.19}$$

Figure 4.4 shows what it looks like to take data which is not linearly separable in a 2D plane and project it into a 3D plane so that it becomes linearly separable [97].

4.3.1 Support Vector Machines

In a classification problem, we are faced with the task of finding a function which divides our samples into different classes in such a manner that samples of the same class are grouped

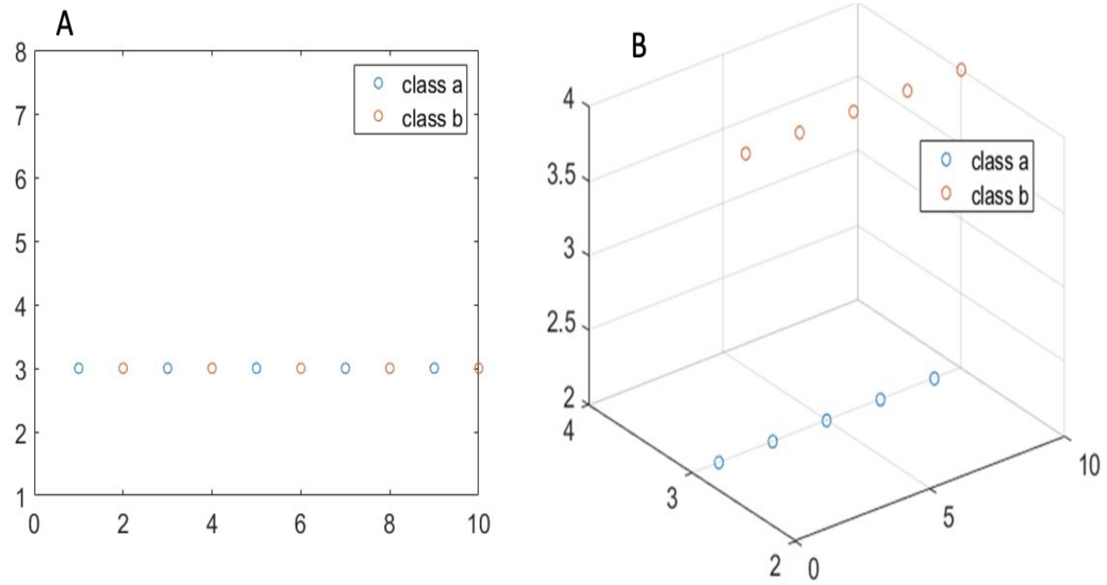


Figure 4.4: A shows a 2D plot of data belonging to two different classes denoted with a red and blue color. It is clear that there is no straight line that can be drawn that could separate the data, thus the data is not linearly separable. Figure 4.4 B shows the same data transformed into a 3D plane. In this configuration the data has become linearly separable.

together. In particular, if we use a support vector machine (SVM) as our algorithm for classification, this algorithm finds a decision boundary which divides the features of the data through hyperplanes. The goal of the SVM is to find the hyperplane which maximizes the distance between the closest features of the classes to the hyperplane. To find the hyperplane, we start by having a vector w, of yet unknown length, which is perpendicular to the hyperplane vector. We also have our samples class A= (x1,x2..xn) which we can represent as vectors which start at the origin and end at our sample location in the x-y plane. Now we want to project the vectors $\vec{xn}$ onto $\vec{w}$ and check whether it is above or below a certain threshold c. This is because the distance between each point in A and the hyperplane is given by $\frac{|\vec{w}\cdot\vec{x_n}+b|}{||w||}$. Therefore, since we have seen that to find the hyperplane we need to find

$$\vec{w}\cdot\vec{x}+b=0. \tag{4.20}$$

we can also set the contraints that, given a sample belonging to class A (see Figure 4.5)

$$\vec{w}\cdot\vec{x_A}+b\geq 1. \tag{4.21}$$

and for a sample belonging to class B

$$\vec{w}\cdot\vec{x_B}+b\geq -1. \tag{4.22}$$

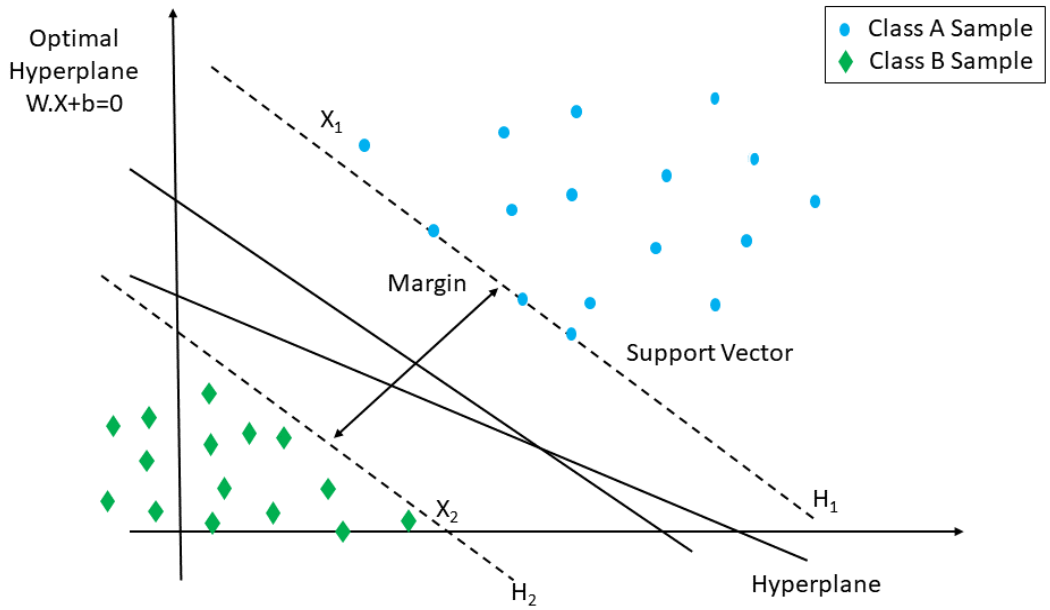


Figure 4.5: shows a representation of how an SVM divides the data by finding the hyperplane which is at a maximum distance from the margins where the closest points between the two classes reside. The SVM also has a kernel function which can project the data into a different space if the data is non-linear.

To make the mathematics easier, we introduce a variable y_1 which is = 1 for the samples belonging to class A and = -1 for the samples belonging to class B. Now we can rewrite the above equations as

$$y_i(\vec{x_i} \cdot \vec{w}) + b - 1 = 0. \tag{4.23}$$

And with this further constraint we are saying that all points which fall on the two dashed lines shown in Figure 4.5 need to be at a distance of 0 from the hyperplane and all other samples a distance of either 1 or -1 depending on which side they reside. Another boundary condition that we can impose is the Margin width, shown in the figure with the black arrow. The margin is equal to

$$\frac{(\vec{x_A} - \vec{x_B}) \times \vec{w}}{|| \, w \, ||}. \tag{4.24}$$

Now, replacing x_A and x_B with the Equation 4.23, we get $X_A = 1 - b$ and $X_B = 1 + b$ so Equation 4.24 becomes $\frac{2}{||w||}$. We want this width to be as large as possible, which means that we want to minimize the expression

$$min(\frac{1}{2 \mid\mid w \mid\mid^2}). \tag{4.25}$$

The last constraint is to bound two previous constraints together [79].

$$L = \frac{1}{2 \mid\mid w \mid\mid^2 - \sum \alpha_i (yi(\vec{x_i} \cdot \vec{w}) + b - 1)} \tag{4.26}$$

when we solve this equation for $\frac{\partial L}{\partial \vec{w}}$

$$w = \sum_i (y_i)(\vec{x_i})(\alpha_i) \tag{4.27}$$

and for $\frac{\partial L}{\partial b}$

$$\sum_i (y_i)(\alpha_i) = 0 \tag{4.28}$$

Now we are going to insert Equation 4.27 into Equation 4.26 and the result is that

$$L \propto \sum \vec{x_i} \cdot \vec{x_j} \tag{4.29}$$

Hitherto we have found the requisites to find a hyperplane that separates our data. But what if the data is not linearly separable? The SVM is a kernel that finds similarities through convolution the same way the wavelet scattering transform does.

The SVM also allows us, in fact, to transform the data and project it into a space where it is easier to classify. We said so far that we need to maximize dot products (see Equations 4.29, 4.27). This brings us to conclude that we want a function such that

$$K(xi, xj) = \Phi(\vec{x_i}) \cdot \Phi(\vec{x_j}) \tag{4.30}$$

There are different kind of kernel functions which we can use: gaussian or radial basis function, Linear, Polynomial or Sigmoid [80].

4.4 Regression

In regression we want to make a prediction. For linear regression we will use the hypothesis

$$h(\vec{x},\vec{\Theta},\Theta_0) = \overrightarrow{\Theta_T}\vec{x} + \Theta_0 \tag{4.31}$$

and the loss function will be (guess values - actual values)^2 this is an ordinary squared loss function. The goal is to find the hypothesis which minimises this error.

Now we want to set up an optimisation equation

$$J(\vec{\Theta},\Theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\overrightarrow{\Theta_T}(\vec{x^i}) + \Theta_0\overrightarrow{-y^i})^2 \tag{4.32}$$

which can be written as

$$J(\vec{\Theta}) = \frac{1}{n}(\vec{X}\vec{\Theta} - \vec{y})^T(\vec{X}\vec{\Theta} - \vec{y}) \tag{4.33}$$

In order to find the values which minimise this optimisation equation we will take the derivative with respect to each parameter and set it to 0, so we want the gradient. To this equation we also want to add a regularizer term to avoid overfitting. Gradient descent allows us to find the minimum of a function. After taking an initial guess of our Θ value we will move in the negative direction of the gradient till reaching a local or global minima [98].

4.5 Neural Networks

A unit of a neural network is called a neuron.

From Figure 4.6 we can see that the input to the neuron is a vector and to each input number is associated a weight, which we have denoted so far as $\vec{\Theta}$ values in our discussion of linear classifiers. The value of Θ_0 is an offset (now we will start denoting it as w as weight as this is common nomenclature for NNs). Then there will be a non linear activation function. The output will then be

$$f(z) = f(\sum_{i=1}^{n}\vec{x_i}\vec{w_i}) + w_0 \tag{4.34}$$

where this is the equation we have seen for linear classifiers, the difference is that now we see the function of the linear classifier as an argument for another function. The input vector will have a single output value. Now the optimization function to solve for this single neuron problem is

$$J(\vec{w},w_0) = \sum_{i} Loss(NN(\overrightarrow{x^i},\vec{w},w_0),\overrightarrow{y^i}) \tag{4.35}$$

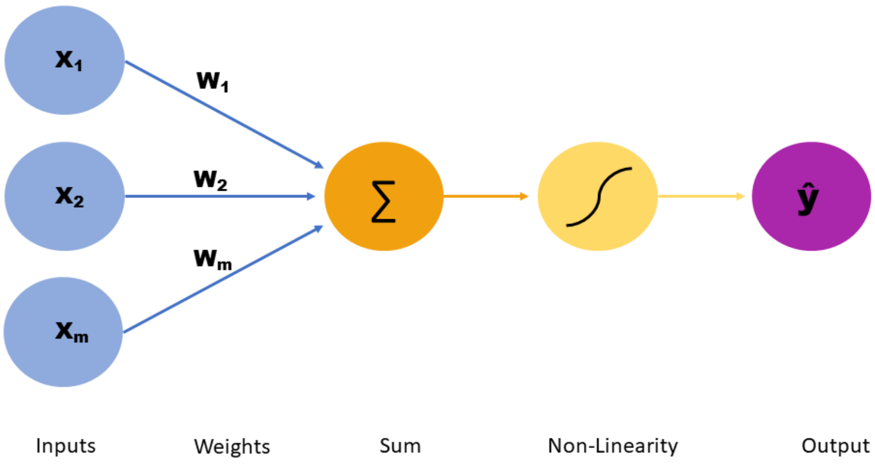


Figure 4.6: shows the structure of a neuron, a single neural network unit. The neuron takes as inputs a set of data points $x_1 ... x_n$ which are multiplied by weights and summed. Then the output goes through an activation non linear function. This produces the final output.

where NN is the output of the neural network.
A fully connected layer is one in which every input is connected to all the neurons (also referred to as units). The number of outputs will be equal to the numbers of units. The number of weights will now be the number of inputs which can be denoted by m times the number of neurons which can denote by n. The number of Θ_0 is n $\times$ 1. The activation function at the output will be

$$f(\vec{z}) = f(\overrightarrow{W^T}x + W_0) \tag{4.36}$$

The argument inside the parenthesis will have dimension nx1. So this layer is a transformation of a vector in m dimensional space to a vector of n dimensional space and it's described using a weight matrix, a vector of offsets and an activation function. Layers can be added together to form a network.

There are many non-linear activation functions that we can use in neural networks. One of this is the Sigmoid function

$$\frac{1}{1+e^{-z}} \tag{4.37}$$

Another activation function that is commonly used is the ReLU function (rectified linear Unit).

The reason for using non-linear activation functions is that otherwise, no matter how many layers we add to a network, we would still have a linear representation of the input vector as output.

The goal then it to train a Neural Network to find the weights which will minimise the loss function $\sum_{i=1}^{n} Loss(NN(\overrightarrow{x^i}, \vec{w}), \overrightarrow{y^i})$. The next step will be to take the gradient of the loss function. Depending on this value the weights will be updated. The weights are usually initialized with random numbers. When considering how big the step size should be each time the weights are updated one should choose a large step size in regions of low curvature and small step sizes in region of steep curvature [100].

4.6 Recurrent Neural Networks

Feed forward networks do not take into account the sequence in which previous data came in. In other words, it does not have a memory to update the weights based on past events. In the Sequential Models section we have introduced the concept of state machines, a model in which in order to get a n output y, we need not only to based the choice on the input x, but also on the previous state of the system.

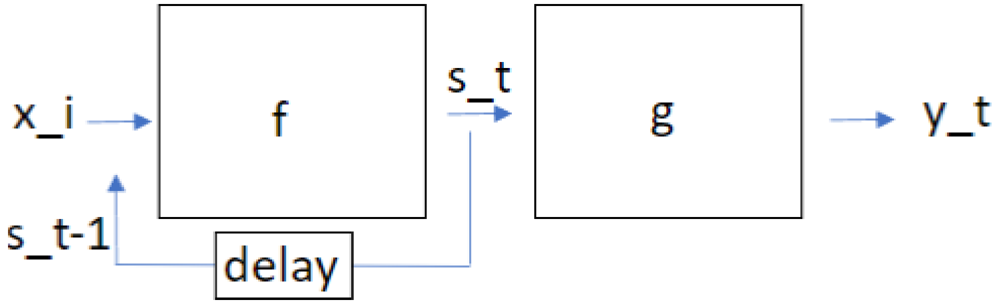


Figure 4.7: shows the logical structure of a recurrent neural network. Instead of just having a set of weights that map x to y and a forward process, y also depends on the previous states of the system.

A state machine starts in a state s_0 and then iteratively finds

$$\vec{s_t} = f(\overrightarrow{s_{t-1}}, \vec{x_t})\vec{y_t} = g(\vec{s_t}) \tag{4.38}$$

and so given a sets of inputs x_1,x_2... the output of the state machine is

$$\vec{y_1} = g(f(\vec{x_1}, \vec{s_0}))\vec{y_2} = g(f(\vec{x_2}, f(\vec{x_1}, \vec{s_0}))) \tag{4.39}$$

A neural network is a state machine with neural networks constituting functions of f and g. Neural networks are trained with supervised learning. Once again we will use a loss function which will check how many predicted values correspond to the true values to check the accuracy of the algorithm. The goal is then to find the weight values that minimise the loss on the training data [102].

4.7 Sequential Models

With sequential models we are interested in seeing how our inputs states evolve over time. In this new scenario, instead of being interested in finding a function which maps an input x to an output y, we are interested in a transition function f which given an input and its state is going to find the transition of the input into a new state. The difference of this transition function compared to the function that we have seen in the previous sections is that the transition function is not only dependent on the current input to give an output, but it is dependent on all the previous inputs (all history of x values which have been fed to the algorithm) [99]. This means that for an input sequence, given a first input x, the output will be

$$g(f(\vec{s_0},\vec{x_1})) \tag{4.40}$$

where $f(\vec{s_0},\vec{x_1})$ is the first state $\vec{s_1}$
so that when we want to find the output of x_2, the previous state will determine the outcome

$$g(f(\vec{s_1},\vec{x_2})) \tag{4.41}$$

4.7.1 Long Short Term Memory Network

A long short term memory network is a type of recurrent neural network well suited to learn the order dependence of a sequence and it is therefore well suited for problems such as speech recognition. Recurrent Neural Networks (RNNs) possess specific qualities:

- Store information for a duration that differs and is adapted from problem to problem.
- Are invariant to noise.
- The system parameters are trainable.

Because of these properties LSTM serve for applications where the signal is information which only makes sense as a sequence. Thus in order to classify an LSTM needs to retain some memory of what came before to make predictions at the next stages. In order to do this, LSTMs use outputs from previous steps as inputs for future steps. The inputs therefore, are not just composed by new data, but by new data plus an old output. The nodes of an LSTM contain a state which has a memory. The previous output, the new data and the memory all used for the calculation that is undertaken at each node. The output not only is fed into the next and previous nodes, but it is also used to change what is stored in the memory [83, 84].

4.8 Conclusion

In this chapter we have explored some machine learning models which serve as optimisation problems to find specific functions which allow to classify data. In the last two chapters we will see how these models can be used on heart valve sound data to classify healthy from unhealthy heart valve sounds and perform biometric authentication.

Chapter 5

Sound Recovery from Light

Continuous monitoring of heart parameters is vital to prevent deaths from conditions which are immediately hazardous, e.g., cardiac arrhythmia [105], heart attacks [106], and stroke [107], sudden death in infants [108], and epilepsy [109]. It can also be useful for prevention and early detection of heart diseases. Current methods (stethoscope, ECG) require contact with the patient's skin, therefore it is only feasible to monitor patients with these devices in intensive or intermediate care units. In this chapter we propose a method for remote and contactless detection of heart sounds which makes it feasible for continuous monitoring of heart sounds in a home setting.

This chapter will firstly expose current methods of remote sound detection, it will then discuss the existing methods used to retrieve sound from light in a contactless and remote manner and then we will discuss how our method is suitable to retrieve heart valve sounds specifically from speckle patterns reflected by the skin. There will be an introduction of sensing technologies and then it will delve into how these are used concurrently with post processing methods to retrieve sound from light. We will then present results of a comparison of two post processing methods: optical flow algorithm versus integration. These methods allow to retrieve sound from the light reflected from the vibrating surface and collected by the camera. The comparison will show which post processing method should be used, which depends on the parameters of the experiment at hand. We will show our results which showcase the relationship between various experimental parameters and SNR for each method.

5.1 Current methods of acquisition of heart valve sounds

Clinicians use stethoscopes in order to listen to the various heart sounds that the heart can produce because these sounds give an insight into the pathology that affects the patients. The stethoscope is placed on top of the chest above the heart. This is where heart valve sounds can be heard best with the digital stethoscope.

Many different electronic stethoscopes have been devised over the years [112, 171]. Although the stethoscope is the gold standard for heart auscultation, as we have already shown, it can only acquire heart valve sounds if placed on the chest above the heart. In recent years, methods to measure the heart sounds remotely have emerged. Such methods acquire heart sounds relying on laser light or radar systems. In order to extract heart sounds with radar systems, a Six-Port interferometer is used. This technology works by summing phase controlled superposition of two input signals S1 and S2 that are superimposed under four different relative and static phase shifts. The resulting four sum signals can be observed at the respective output ports of the structure [114]. However, the group that proposed heart valve sounds radar based acquisition presented in [113], only acquired PCG frequencies up to 80 Hz. This range is not enough to capture the full extent of frequency range that certain heart diseases produce.

Another method for contactless acquisition of heart valve sounds is laser doppler vibrometry. This method, firstly devised by [115], enabled the acquisition of heart valve sound from the chest. The frequency content obtained presented frequencies up to 350 Hz. The experimental set up and limitations of this method will be presented in section 5.4.

Finally, Zalevsky [134] obtained heart rate from subject's wrists, together with some of the lower frequencies of the heart valve sounds.

We will present our method which can retrieve high frequencies of the heart valve sounds contactlessly and from a distance with a better SNR than the stethoscope can and from people's necks rather than from their chests. This is because the stethoscope is not well suited to retrieve heart valve sounds from semi-periferal and periferal locations, such as from the arteries on the neck or the wrist. As we discussed in Chapter 2, certain illnesses, such as heart murmurs, lie in this higher frequency range. That is why we will now consider methods which are capable to detect nanometer vibrations of materials caused by sound vibrations which allow to retrieve the high frequencies that certain heart illnesses produce.

5.2 Brief History of Sound Recovery from Light

It was on June 3rd, 1980 that Mr. Alexander Graham Bell and his assistant Mr. Sumner Tainter discovered the photophone. Later on that year, in May, 1878, Mr. Bell spoke at the meeting

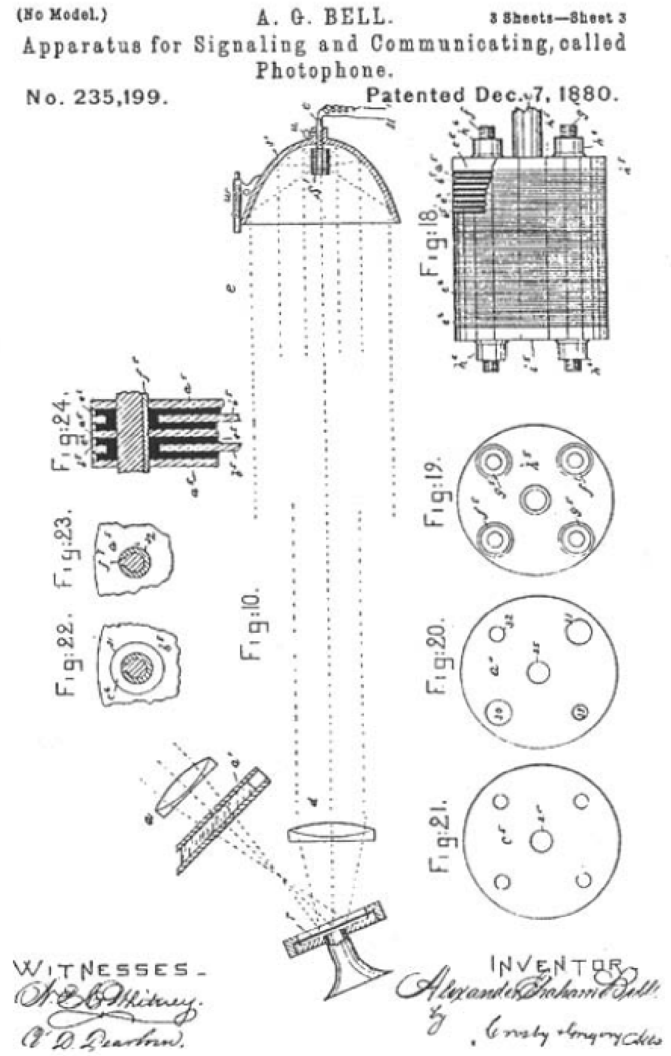


Figure 5.1: shows the schematic set up of the photophone [119]. On the bottom left it is possible to see the vibrating diaphragm and the selenium detector [120].

of the American Science Association in Boston introducing his experiment which allowed to retrieve sound through light. The transmitting instrument was placed on the top of the Franklin school house, in Washington, while the receiving instrument was eight hundred feet away, placed near a window of Mr. Bell's laboratory.

It was discovered that selenium's resistance changed depending on the amount of light that fell upon it [116].

His first experimental set up consisted of a grating which was attached to a silver coated glass plate attached to a diaphragm (in acoustics, a diaphragm is a thin membrane or sheet of various materials suspended at its edges [117]). Sound vibrations caused the grating attached to the diaphragm to vibrate thus changing the intensity of the light [119] that passed through it. These vibrations modulated the intensity of the light. Since there is a linear relationship between the change in light intensity and intensity of sound, light intensity modulation could be used to extract sound.

Bell was very excited for this discovery, claiming it was his most beloved invention, even more so than the telephone. In a letter to his father that he wrote a few days after his first successful recovery of sound from light he wrote: "I have heard articulate speech produced by sunlight!...Can imagination picture what the future of this invention is to be!... We may talk by light to any visible distance without conducting wire... In warfare the electric communications of an army could be neither cut nor tapped."

Since then, different methods to achieve sound retrieval from light have been developed thanks to technological advancements and they will be shown in the following sections.

5.3 Image Sensors

The ability to capture sound from light has been made possible by recent advances in imaging sensors. In this section we will discuss the principles of operation of these sensors.

In order to capture sound from light it was important to have a sensor which was sensitive to very low light levels and which had a high frame rate. For these reasons we collected light both with a CMOS and with a SPAD camera. Ultimately, because the SNR of the signal was similar for the two cameras, we chose to use the CMOS, since it's cheaper.

In the following section, the working principles of both cameras will be described.

5.3.1 CMOS Sensors

The complementary metal oxide semiconductor (CMOS) was invented in 1963 by Frank Wanlass. The sensing element of a CMOS detector can be a photogate, phototransistor or photodiode. Here the photodiode working principles will be discussed. CMOS sensors are made of photodiode pixels which convert light into an electrical signal via the photoelectric effect: when light hits the silicon pixel it will dislodge electrons and the number of electrons will depend on the energy of the photon which will depend on its wavelength [122]. These electrons are then converted to a voltage and then into a digital value using an on-chip Analog to Digital Converter (ADC) [123].

Each of these pixels are composed by a photodiode and three transistors. The photodiode is responsible for transforming the photons to electrons and the transistors reset and activate the pixel and amplify the charge [124].

The photodiode is a semiconductor device with a P-N junction. If the photon strikes the photodiode in either the P or N layers, the electron hole pairs will be recombined in the material. But if photons are absorbed in the depletion region then the electron hole pairs will travel to opposite ends of the photodiode due to the electric field. Electrons will move toward the positive potential on the Cathode, and the holes will move toward the negative potential on the Anode. These moving charge carriers form the current [125].

5.3.2 SPAD Sensors

Semiconductor-based single-photon avalanche diode (SPAD) detectors are avalanche photodiodes biased at fields above avalanche breakdown, in Geiger mode, where a self-sustaining avalanche current can be triggered by an incident single-photon [126].

In a pixel of a CMOS, usually made of a photodiode, the reverse bias voltage is low so the current changes linearly with absorption of photons. However, in a SPAD, the reverse bias is so high that a phenomenon called impact ionisation occurs which is able to cause an avalanche current to develop [127]. This property makes the SPAD sensitive to single photons, in contrast to photodiodes, which instead count the amount of light that has arrived in a certain amount of time.

5.4 Sound Retrieval from light: Phase Modulation

In order to recover sound from light's phase modulation one must build a Laser Doppler Vibrometer (LDV). An LDV is a non-contact measurement device which measures the Doppler frequency phase shift of a laser beam reflected from a moving target. An LDV consists of an interferometric setup. This consists of a coherent laser beam with frequency f_o which is then divided in two by a beam splitter. One beam collides with the vibrating surface, this will be called the probe beam, which oscillates at the sound frequency. Then the reflected light returns to a second beam splitter where it interferes with a reference beam. The reason why there is an intensity fluctuation is because when two coherent light beams combine (a reference beam and a probe, in the case of an interferometer), the resulting intensity has a component that is related to the difference in path lengths of the two beams. The resulting output beam, instead of being uniform will present a fringe pattern whose shape will depend on the number of half wavelengths that make up the difference in path length between the reference and measurement beams. These vibrations of the object produce intensity fluctuations once the the beams are recombined into one which is then collected by a detector. A detector is then used to convert the signal to voltage [128] according to the equation below.

$$v = 2fm \tag{5.1}$$

where v is the velocity of the vibrating object while f is the frequency and m is the magnitude of the vibration.

There were a few reason why we didn't employ this method to retrieve heart valve sound signal with although this method was shown to acquire heart sound frequencies up to 350 Hz from the chest (probe laser beam pointing at the bare chest of an individual). The major drawback of this method is that the projection laser and the detection interferometer module must be placed in very specific positions such that the reflected beam is directed towards the detection module, this reduces the applications to very specific regimes. As noted by [129], another problem of this method is the speckle noise. This is an interference phenomena which will be better described in later sections where a speckle pattern arises when a coherent laser light hits a scattering surface.

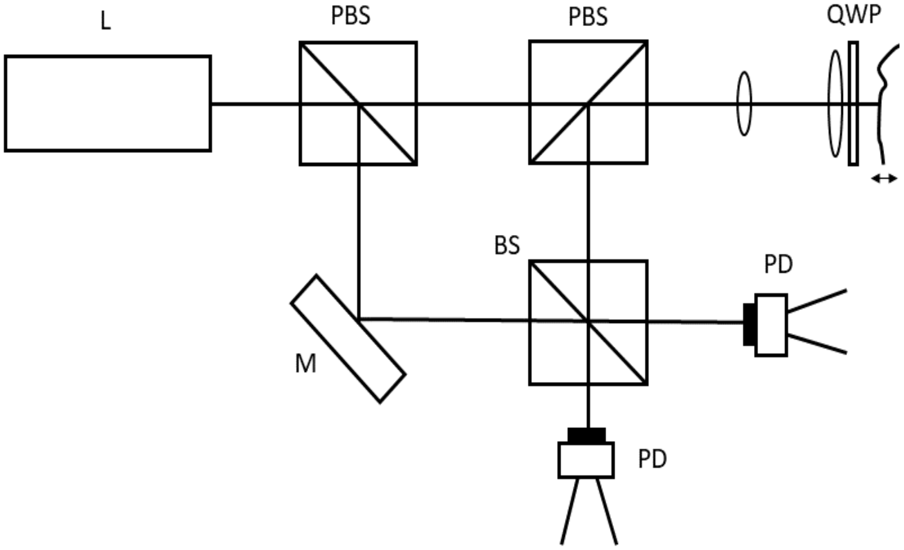


Figure 5.2: shows a Mach-Zehnder interferometer. M is the symbol for mirror, BS for beamsplitter, PBS is a polarizing beamsplitter, L is a laser, QWP is a quarter-wave plate, PD is a photo detector, and T is a telescopic lens array.

In Figure 5.2 it is shown the experimental set-up of an LDV system. The optical path of the reference beam is constant over time, the movement of the vibrating object (shown in the figure after the quarter wave plate) upon which the probe beam is incident will generates a pattern on the detector. One complete light / dark cycle on the detector corresponds to object being displaced exactly half of the wavelength of the probe beam. Since usually visible light is used, this corresponds to around 300 nm. If the object moves, there will be a Doppler frequency shift. This means that the modulation frequency of the interferometric pattern is directly proportional to the velocity of the sample. This interefometric pattern assumes the same shape whether the object moves away or towards towards the interferometer, therefore this set-up cannot determine the direction the object is moving.

Another configuration of the LDV uses a Bragg cell, which is an acousto-optic modulator that typically shifts the light frequency by 40 MHz. This is placed in the path of the reference beam. This generates a typical interference pattern modulation frequency of 40 MHz when the sample is at a standstill. If the object then moves towards the interferometer, this modulation frequency is increased, and if it moves away from the interferometer, the detector receives a frequency less than 40 MHz. This means that it is now possible to not only clearly detect the path length, but also the direction of movement too [132].

5.5 Sound Retrieval from light: Intensity Modulation

5.5.1 Passive Retrieval

An MIT group was the first to demonstrate a method to retrieve sound from light in a passive manner [131]. This method is described as "passive" because compared to the active methods of sound retrieval through light phase modulation or light amplitude modulation, where it is a requirement the laser beam must be incident upon the vibrating surface, that is not a requirement for the passive retrieval method. This method works on the principle that the vibrations that sound causes in an object manage to induce enough vibrations on the object itself to be visibly detectable through a high speed camera. After the video of the object has been acquired, the sound is extracted through post processing methods.

The MIT group which first devised this method, in order to retrieve the sound they extracted the local motion of the signal by applying a steerable pyramid algorithm. After the signal is decomposed and from each decomposition a signal is extracted, these local signals are aligned and averaged into a single 1D motion signal that captures global movement of the object over time.

It is to be noted that in comparison with active retrieval methods, this procedure yields a worse

SNR. This is due to the fact that, depending on the surface that is being recorded, the object in question might not move much due to sound. In fact, for certain kind of surfaces, sound doesn't perturb them enough for the discretised camera pixels to be able to see motion. However the advantages of this method are that it does not require active lighting from a laser and also it does not require that the vibrating surface is very reflective.

It is to be noted that not all objects are equally good for visual sound recovery, in fact, the objects which vibrate most are those with high compressibility or have a particular shape. Therefore the choice of object is important to determine whether it is possible to observe vibrations.

After the video has been acquired the post processing method used was a "steerable pyramid algorithm". This algorithm consists on filtering each frame of the video into sub-bands of different scales and orientations. This is achieved with filters, which can be used to extract the signal at different scales and orientations so that more sound signal can be extracted and added together to improve the SNR.

5.6 Sound Retrieval from Light: Speckles

When coherent light is reflected from a rough surface or propagates inside a medium with random refracting index fluctuations, the interference of multiple coherent spherical waves forms a speckle pattern in the far field [121]. This speckle pattern is sensitive to a nanometer-scale precision to surface movement. Thus, when shining a surface, which is vibrating due to sounds, with coherent laser light, the movement of the reflected speckle pattern will carry information both on the structure of the surface and on the sound carried by the surface.

In the next sections we will explore speckles statistics and various post processing methods which allow the retrieval of sound.

5.6.1 Sound Recovery from Speckle Tracking

Zalevsky's group has devised a method to retrieve sound from light by exploiting the movement of speckle patterns [134]. The method consists in imaging the reflected speckle patterns scattered off of a remote scattering surface with a fast camera. Because the surface does not change in shape as it vibrates but it only slightly tilts, the speckle pattern exhibits the memory effect, which means it does not change in shape but it only shifts. In order to measure the shift of the speckle pattern in time different methods can be used. Below there will be a brief physical descriptions of these methods since these are the methods that we used in our experiment.

- Normalized cross correlation (NCC):

In the NCC method, features between images are tracked by template matching. This approach involves shifting an image over another image until the best comparison is found. By looking at the cross correlation between two images, it will be possible to see that there will be a peak and everywhere else will fade to 0. The peak will represent the position where the two images would need to be at to be overlapped. It is by tracking this peak from it's original position at the middle of the image that the displacement of the two images can be inferred. And by tracking the displacement of this peak over time, sound can be retrieved [133].

The cross correlation equation is

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h(\vec{u},\vec{v}) F[i+\vec{u}, j+\vec{v}] \tag{5.2}$$

This equation just shows that one of the images will be slided over the other. At each location all the products of all the pixels of the two images will be summed together and then the result will be the new result for that pixel.

Zalevsky used this technique in his paper to retrieve sound [134].

- Optical Flow Algorithm:
  The flow algorithm uses the information between adjacent pixels to calculate the motion information between adjacent pixels. It allows to estimate the velocity of objects in a video and estimate their velocity in the next frames. Optical Flow is the motion of objects between consecutive frames.

  In the first attempt at building the optical flow algorithm, Horn and Schunck proposed an assumption that stated that the intensity of an object would not change between two consecutive frames [136, 137]. So that for a given pixel of an object, its intensity value in a successive frame would be

$$f(x,y,t) = f(x+\Delta x, y+\Delta y, z+\Delta z) \tag{5.3}$$

  This is called the Brightness Constant Assumption. This constraint, since far from reality, is only true for very small intervals of t, for which it can be assumed to be reasonable. This equation has 2 unknowns. So more constraints must be found.

  The second constraint is that the motion speed of the pixels must be similar and cannot be abrupt, meaning with sudden changes.

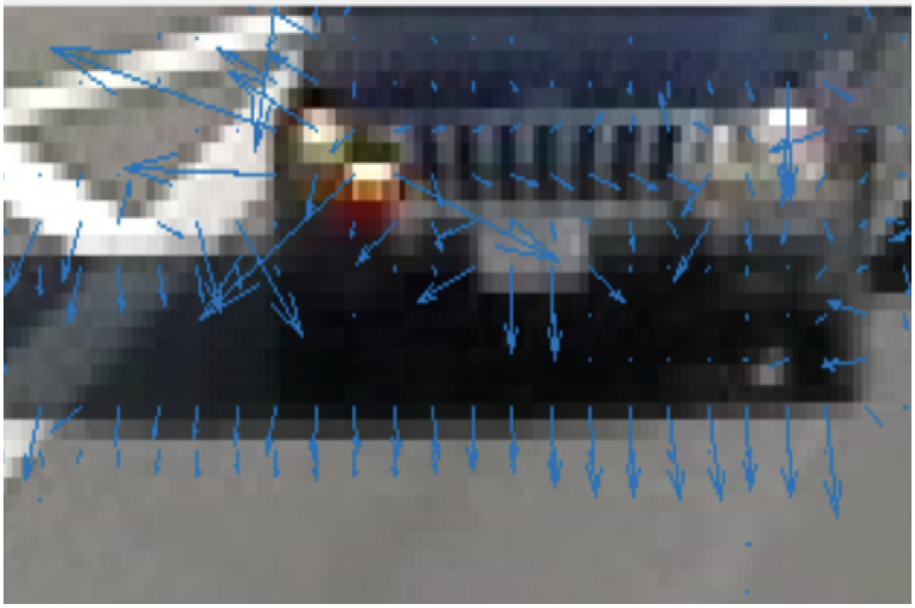


Figure 5.3: shows the concept of the Optical Flow algorithm's purpose. The Optical Flow allows to track objects between frames, estimate their current velocity to detect the position in consecutive frames so to track their movement. An object at position (x,y) will undergo a displacement and be at position (x+dx,y+dy) in the next frame at some time later. The constriction will be to say the intensity of the object will be constant.

Now, if we Taylor expand the above equation

$$f(x,y,t) = f(x,y,t) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y + \frac{\partial f}{\partial t}\delta t + \ldots \tag{5.4}$$

We ignore the higher order terms and simplify the equation to obtain

$$0 = \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y + \frac{\partial f}{\partial t}\delta t \tag{5.5}$$

And by diving by dt we obtain the equation and cancelling terms that equal 1 we obtain

$$0 = \frac{\partial f}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial f}{\partial y}\frac{\delta x}{\delta t} + \frac{\partial f}{\partial t} \tag{5.6}$$

Every pixel moves with velocity $\frac{dx}{dt}$ in the x direction, expression to which we can give the notation u and with velocity $\frac{dy}{dt}$ in the y direction which we will denote as v. We will also denote $\frac{\delta f}{\delta x}$ as f_x and $\frac{\delta f}{\delta y}$ as f_y and $\frac{\delta f}{\delta t}$ as f_t. So that we can rewrite the above equation as:

$$0 = f_x\vec{u} + f_y\vec{v} + f_t \tag{5.7}$$

This is the Optical Flow Equation. The two unknowns of this equation are u and v which respectively are the velocity of x in the x and y direction.

This expression can be rearranged to be in the form

$$\vec{v} = -\frac{f_x}{f_y}\vec{u} + \frac{f_t}{f_y} \tag{5.8}$$

which is the equation of a line which can be drawn in the x-y plane as shown in Figure 5.4

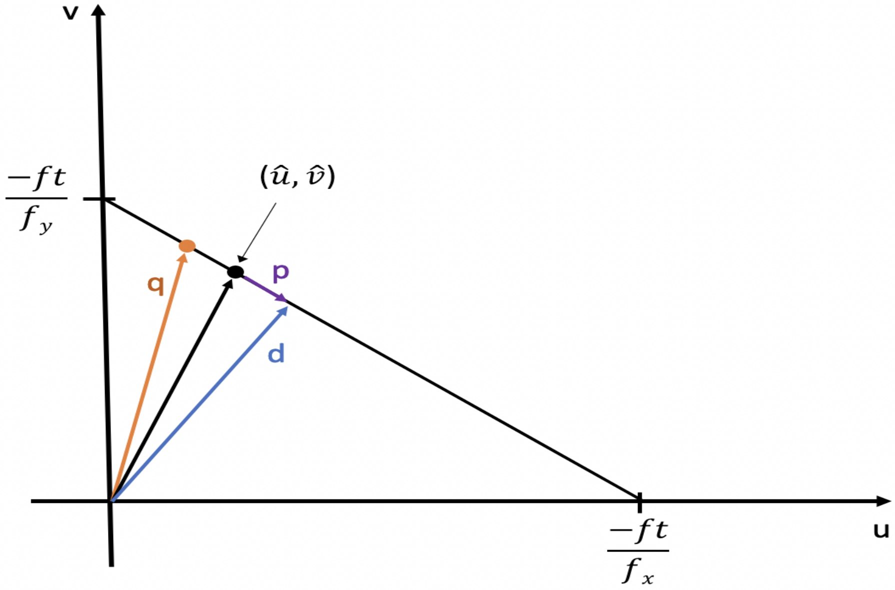


Figure 5.4: shows the graph obtained through equation 5.8. The line intercepts the y-axis at -$\frac{f_t}{f_y}$ and it intercepts the x-axis at- $\frac{f_t}{f_x}$. The distance d from the origin to the line can be inferred through trigonometry. However the distance p cannot be inferred.

So, equation 5.7 is an equation of a straight line. The optical flow vector could be anywhere along this line, this is why the problem is unconstrained. So the optical flow vector can be split up into its normal and parallel components. The perpedindicular component can be found through the formula $(I_x, I_y)/(sqrt((I_x)^2 + (I_y)^2))$. On the other hand, the parallel component cannot be computed. This is referred to as the aperture problem. Thus another constraint needs to be added, this is, in the case of the Horn and Shunck method, the Smoothness Constraint Assumption. This constraint states that all pixels within a small neighborhood move in the same direction, thus have similar optical flow vectors. This constraint works for example, for the majority of the pixels belonging to an object, but it does not work for pixels between the object and background. This concept can be best visualized from Figure 5.5. In this figure the green pixels represent optical flow vectors which do not change much within their own neighborhood while the blue and yellow contours of the object are those pixels which lie between the object and the background and whose optical flow vectors do change compared to the optical flow vectors of the pixels in their neighborhood [138].

From the constraints of smoothness and brightness we can derive a cost function

$$E = \iint_{x,y} (f_x u + f_y v + f_t)^2 + \gamma((u_x)^2 + (u_y)^2 + (v_x)^2 + (v_y)^2)\delta x \delta y \tag{5.9}$$

where the first term, which we have seen is the brightness constrains (see Eq. 5.7) and the second term is the smoothness constraint (see Eq. 5.8) multiplied by a factor λ. The λ term is a regularizer term which dictates how important either terms are. As before, u is the derivative of the velocity in the x direction and y is the derivative in the y direction. The goal is to minimize this cost function equation, in order to do this the Euler-Lagrange equation will be used, so the derivative with respect of u and v will be taken and put to equal 0 and then it will be subtracted to a derivative of x and y

$$\frac{\delta E}{\delta u} - \frac{\delta u_x}{\delta x} - \frac{\delta u_y}{\delta y} = 0 \tag{5.10}$$

This equation becomes

$$2(f_x u + f_y v + f_t) f_x - 2\gamma((u_x)^2 + (u_y)^2 + (v_x)^2 + (v_y)^2) dxdy \tag{5.11}$$

and it allows to find u. The same equation can be solved to find v by taking the derivative of E with respect to v.
So what we are trying to do is to find

$$e = e - s + \lambda e_c \tag{5.12}$$

the u and v values at every pixel that minimises the total error.

Another Optical Flow algorithm is the Lucas-Kanade algorithm [139]. This algorithm works similarly to the Horn and Schunck algorithm. The assumption is that for a small neighborhood of pixels the Optical Flow vectors, (u,v), will be similar. This implies that we can write an equation that states that the derivative of the intensity in a neighborhood of pixels over time is equal to.

$$0 = I_x(x,y)u + I_y(x,y) + I_t(x,y) \tag{5.13}$$

There will be such an equation for each pixel position (x,y). The reason why this system works is because these equations are not linearly dependent. That is because images usually have a lot of different colors thus the intensity I_x and I_y will be different from pixel to

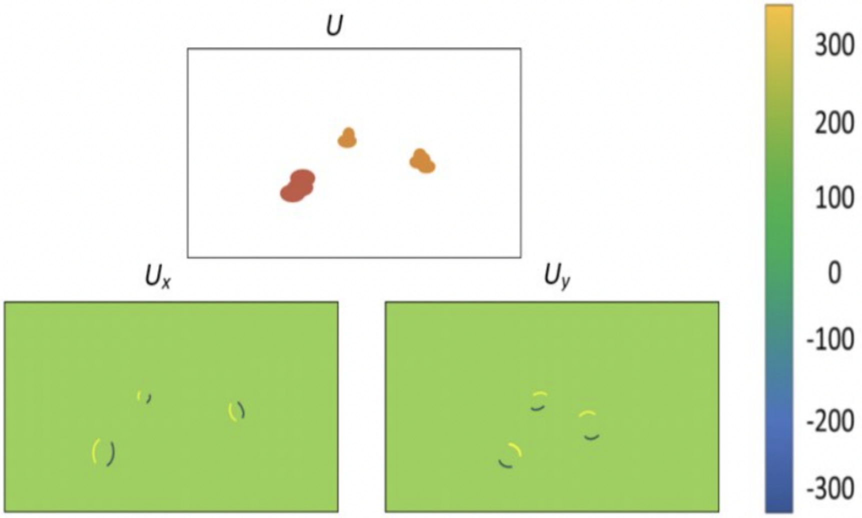


Figure 5.5: The green pixels represent the point in an image which have similar optical flow vectors with each other. As mentioned, almost all background pixels will have similar optical flow vector and the same will be true for almost all object optical flow vectors and the only pixels for which this statement is not true is for those pixels between the object and the background.

pixel.

These system of equations can be written in matrix form as Au=B.

$$\begin{pmatrix} I(x,y) & I(x,y) \\ I(x+1,y+1) & I(x+1,y+1)\ldots \\ \ldots & \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} I_t(x,y) \\ I_t(x+1,y+1)\ldots \end{pmatrix}$$

And this equation can be solved as $u = (A^T A)^{-1} A^T B$ and similarly for v for each u and v vectors of each pixel.

The assumption for both the Lucas-Kanade and Horn-Shunck algorithms is that dt in these equations is very small, meaning we are comparing consecutive frames. Furthermore, the Lucas-Kanade algorithm will not work when the image has no texture and all the derivatives of intensities will be 0. Another region in the image where the Lucas-Kanade algorithm has trouble detecting the real direction of motion are the edges, where the gradient in one direction is much greater than the gradient in the other direction, an image of this concept is shown in Figure 5.6.

So far we have have seen two examples of sparse optical flow algorithms, which only compute the flow of a few pixels. These selected pixels are edge pixels selected through

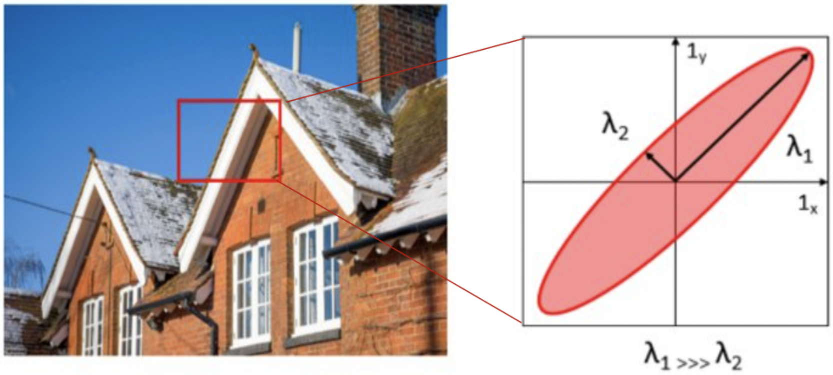


Figure 5.6: shows the gradient at the edge of an object in an image. Lucas-Kanade does work well in places where the gradient changes in all direction, but it has more trouble detecting the direction of change of an object when the object has a high gradient of change in one direction (in the case of this image in the x-y direction), but not in the other (-x-y direction).

another algorithm, the Shi-Thomasi algorithm [140, 142].

The Gunnar Farneback Optical Flow algorithm is a dense optical flow algorithm, which, contrary to the sparse optical flow algorithms (Lucas-Kanede and Horn-Shunck), computes the optical flow of all pixels and is thus slower in computation. This algorithm starts by approximating the neighborhood of each pixel with a polynomial

$$f(x) = x^T Ax + b^T x + c \tag{5.14}$$

and if the image is translated in the second frame the new equation will be

$$f(x) = (x-d)^T A(x-d) + b^T(x-d) + c \tag{5.15}$$

and the displacement d is thus

$$d = -\frac{1}{2}A^{-1}(b_2 - b_1) \tag{5.16}$$

So far the algorithm has been based on the assumption that the signal of the image is a polynomial and that from image 1 to image 2 some dt time later, the signal has only undergone a global translation. These assumptions are non intuitive. Thus we need to check that the errors related to these assumptions can be kept small. We start this check by performing a polynomial expansion on equations 5.14 and 5.15. The goal is to make sure that As and bs values vary with location. The result of this will yield our first constraint:

$$A(x)d(x) = \Delta b(x) \tag{5.17}$$

Then we make another assumption: that A and b don't change too quickly, this equation is written as

$$G[i, j] = \sum w(\Delta x) \mid\mid A(x + \Delta x)d(x) - \Delta b(x + x) \mid\mid^2 \tag{5.18}$$

From this equation we want to find the value of x which minimises the equation.

This method of sound extraction by tracking the speckle motion with the Farneback optical flow algorithm was used by [141].

5.6.2 Sound Recovery from Intensity Variation

We have seen that Bell retrieved sound from light thanks to the amplitude variation of the total intensity which changed with the same frequency of sound. It is also possible to recover sound from the variation of intensity of light given by the movement of the reflected speckle pattern in the direction perpendicular to the camera and it is even possible to recover sound from light that has travelled through scattering material by measuring the change of energy of a small region of the reflected light.

In this section we will mention all of these methods.

Photodiode and mask for sound retrieval

Veber [143] proposed a method in which he used a single photo-diode and a mask in order to retrieve sound. In their work they showed that they used two methods to convert light to sound: in the first method they used a lens to increase the size of the speckles so that a single speckle would hit the photodiode.

The second method consisted, as shown in Figure 5.7 of inserting a mask before the camera. Veber found that by using a lens to enlarge the speckles' size, given the photodiode was supposed to collect only one speckle, the irradiance wasn't sufficient to get signal. Furthermore, the amount of enlargement the speckle needed to be the only speckle the photodiode captured depended on the specific parameters of the experiment thus it wasn't general enough for all applications. The next method they tried was to use a mask in front of the photodiode. According to Veber, he claims that the mask allowed to measure the movement of several speckles instead of one and thus the SNR improved. This method allows for fast time of acquisition and it does not require post processing to obtain the sound [143]. The major limitation of Veber's work is that the pattern of the mask needs to be well designed, fabricated and adjusted according to the shapes and the sizes of the speckle patterns and the photodetector used. This severely limits the application range as for each distance and scattering surface a new mask needs to be fabricated.

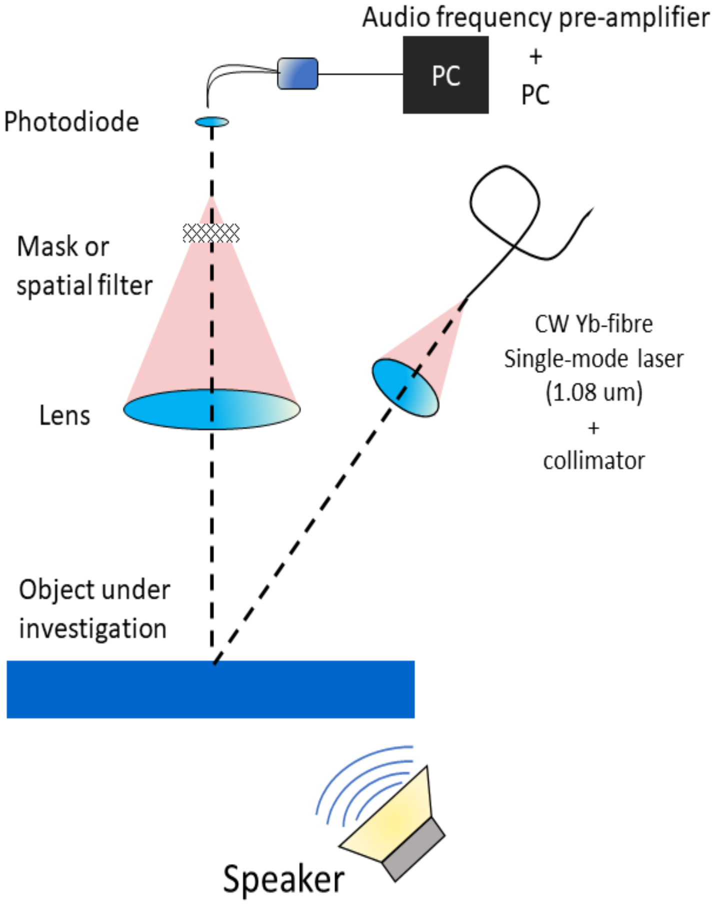


Figure 5.7: shows Veber's experimental design. The light is directed towards a rough surface. The refllected speckle pattern is enlarged through a lens. A mask is placed before the photodiode to split up the single speckle into multiple speckle.

Sound retrieval through grey value variation

Chen's group proposed a method of sound reconstruction based on grey value variation of the speckle pattern which moves perpendicularly to the camera detector. The method works by taking a fixed pixel in the speckle pattern collected by a multi-pixel camera and extract sound from its gray value variation as the speckle pattern shifts in time without changing shape thanks to the memory effect. This method relies on the fact that if the speckles are large enough and are several times bigger than the pixel size, since their intensity is Gaussian shaped, as the speckle moves perpendicularly to the pixel, the pixel will see a different intensity value of the Gaussian shape at each time frame.

Chen's group explained that, although compared to other techniques such as speckle tracking, this method provides worse SNR, it is possible to improve the SNR by selecting pixels based on these criteria:

- Along the displacement direction the selected pixel should situate at the middle position of neighbor minimum and maximum;
- The distance between neighbor minimum and maximum is twice as large as the maximum displacement during the movements;
- The gray value variation between the neighbor minimum and maximum is as linear as possible;
- The gray difference of neighbor minimum and maximum is as big as possible in order to ensure a large gray variation while translation takes place [144].

Finally, after selecting various pixels with these characteristics, the grey values of the pixels of each image are added together so to obtain a 1D signal in the time dimension.

The problem with this method is that once again it requires post processing as the retrieved sound is much noisier than the signal obtained with the Zalevsky's method.

The most recent form of improvement in sound extraction through active retrieval with amplitude modulation was carried out by Zhu's group [145]. They used a variance-based method to select pixels that have large variances of the gray-value variations over time. Then the gray-value variations are summed together. The limitations of the similar methods of Zhu and Chen's groups are that they state that when choosing a random pixel, they cannot retrieve sound as it is just too noisy.

Sound retrieval from light amplitude variation

Bianchi's group was the first to show that it is possible to retrieve sound from speckle patterns simply by collecting the reflected speckle patterns moving due to a surface and then integrating the total intensity values collected by the camera at each time frame [146]. Bianchi's assumption is that the vibration amplitude a(t) is small enough that the read out power P(t) can be approximated as a truncated Taylor's expansion.

$$P(a(t)) \cong P_0 + P_1 a(t) + \frac{1}{2} P_2 a(t)^2 + \ldots \frac{1}{n} P_n a(t)^n \tag{5.19}$$

Bianchi then considers the case in which there is a local tilt of the surface causes a translation

θL of the speckle pattern. Importantly, since there is an equal chance of the value of P incident upon the detector to increase or to decrease as the speckle pattern tilts due to vibration, the statistical average is

$$\frac{\partial_n P}{\partial_n a} = 0 \tag{5.20}$$

Measurements were acquired with different frame size D in order to establish the relationship of SNR to D. The results of the relationships of various parameters are shown in Figure 5.8.

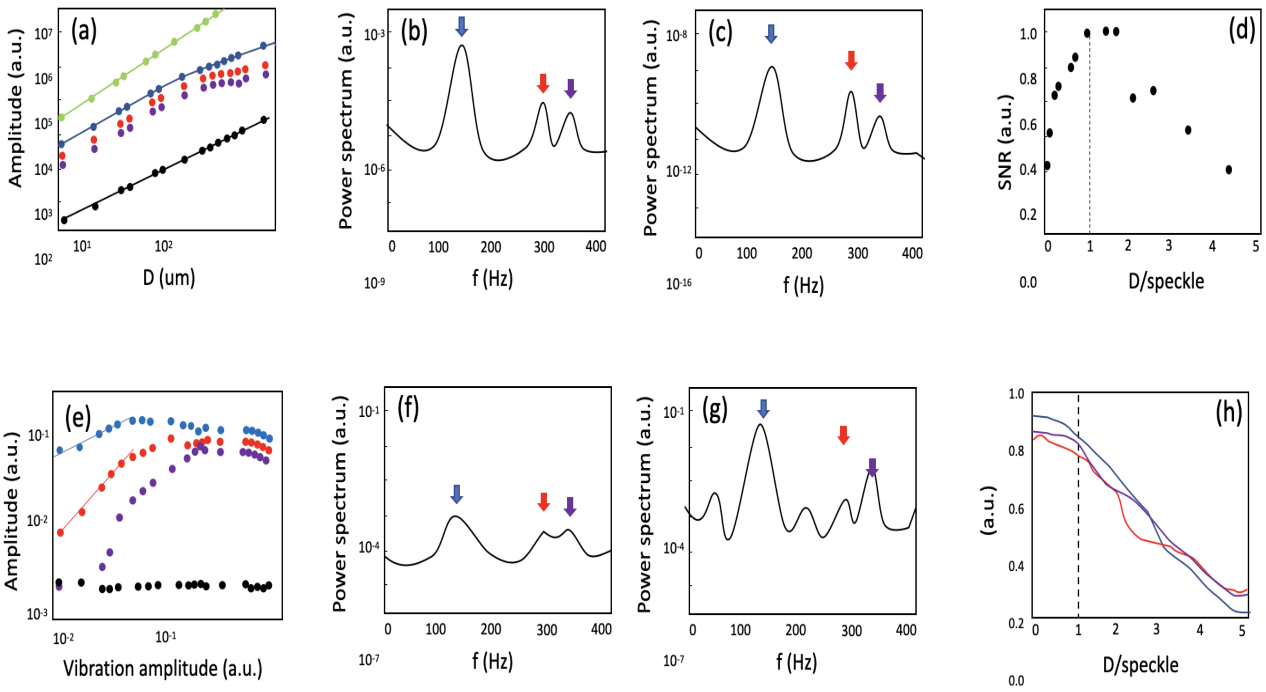


Figure 5.8: The figure shows the relationship between various parameters when a laser beam at 532 nm strikes a membrane which is vibrating due to a speaker playing behind it. The scattered light forms a speckle pattern which oscillates in time and which is collected by a detector (camera or photodiode). W is the size of the laser spot incident on the membrane; L is the distance between the detector and the target; D is the size of the aperture of the detector; ϕ is the local tilt of the membrane.

Figure 5.8 (a) shows the amplitude of the various terms in equation 5.19 as a function of the aperture of the camera: P0 (green), P1 (blue), P2 (red), P3 (purple), and error term in (black). Image (b) shows the power spectrum when D (aperture of camera) is 5 times larger than the speckle size and is 700 μ m. Image (c) shows the power spectrum when D = 7 μ m. It can be seen that in these regimes the higher order terms of equation 5.19 start to be as large as the first order term. Image (d) shows the SNR as a function of D calculated as the squared ratio between P1 and the error term. Image (e) shows the amplitudes of the first three harmonics and of the error as a function of the displacement of the vibrating surface. Then in images (f) and (g) it is shown the power spectrum of Pt for the lowest and the highest value of vibration amplitude respectively. Image (h) shows the standard deviation of P1, P2, and P3 divided their average values as a function of D. The blue, red, and purple arrows indicate the peaks corresponding to first, second, and third harmonic, respectively.

According to these results Bianchi claims that the best aperture size of the camera should be about 5 times as large as the speckle size if the desire post processing method is integration because this assures that there won't be higher order terms as is the case if the detector size is too small.

5.7 Sound Retrieval from Light Amplitude Integration

The next sections will show the results of our experiments for sound retrieval from secondary speckle patterns.

Firstly we will show experiments and complementary simulations of sound retrieval from the integration of all amplitude values of each frame. As we first recovered sound from simple integration of speckle patterns moving due to sound, we were quite surprised to see that we could recover sound as our initial thoughts were that since when the speckle patterns moves with respect to the camera, as some speckles leave the field of view, others enter the field of view and thus the total intensity over time should be 0. Furthermore, we thought that if there was such an easy solution to sound retrieval, many more papers should be using it rather than finding intricate solutions such as speckle tracking and masks.

It was only after some more research that we found Bianchi's paper [146] in which he used a photodiode to retrieve multiple speckles to retrieve sound.

We managed to go a few steps farther and retrieve sound from light scattering off different diffuse media both in line of sight and out of line of sight of the camera, bringing us in a regime where we could not rely on memory effect, thus the speckle pattern, as it was the case in all the methods we have mentioned so far, changed configuration from frame to frame.

The first more complex situation consisted in including an additional scattering layer which sits between the vibrating system and the detection system. Such situations may arise when for example there are additional translucent layers between the observer and the surface or the surface is beyond the direct line-of-sight and is accessed by scattering from an additional surface. We also consider a system with three scattering layers, where the vibrating surface is out of the line of sight of the camera. We purposely chose therefore to study a regime in which the additional scattering layer is placed far away from the vibrating surface. This in turn implies that there is no speckle 'memory effect ', i.e. the speckle pattern is not shape-invariant as there is no relation between the various speckle images as the surface vibrates. Thus, we cannot rely on the linear correlation between the speckle pattern shift and the amplitude of sound.

We then simulated these experiments through MATLAB to make sure these results were not due to mistakes committed in the lab and to try to understand where the sound came from given that intuitively, if vibration brought some speckles inside the field of view of the camera as other speckles got out of the field of view, as stated by Bianchi in equation 5.20, the intensity over time should be 0.

Then we will show the results of SNR vs distance for the two different methods of sound retrieval from light: speckle tracking and intensity integration.

5.7.1 In line of sight amplitude modulation retrieval

The first experiment was very simple and served as proof of concept. A focused laser beam was directed to an aluminium surface which was given freedom of movement. A speaker was placed behind the aluminium surface and the sound coming from the speaker vibrated the surface. Part of the reflected speckle pattern scattering off the surface was then collected by the SPAD camera.

There is an inversely proportional relationship between the size of the focused laser beam and the mean average size of the speckles of the speckle pattern. In our first attempts we decided to focus the laser beam as much as possible so to have large mean speckles sizes. The speckles' size is inversely proportionally related to the roughness of the scattering surface. Therefore we have chosen an aluminium foil as the scattering surface, its smoothness allows speckles' size with average size of the order of mm at around 10 cm from the scattering surface and our initial goal was to have speckle's seizes that allowed one single speckle to occupy more than a quarter of the detector.

In addition, the aluminium foil was chosen because it is a reflecting surface. Furthermore, the low weight of an aluminium sheet allows it to move substantially when hit by sound waves.

After the amplitude modulated speckle pattern is collected by the SPAD, the values of each frame are read as amplitude versus time so that at the end, from a 3D signal of frames vs time, we are left with a 1D time signal.

Experimental set-up

The experimental set up is shown in Figure and the components are: a laser beam (Laser Quantum Gem, wavelength = 532 nm, max.power = 2 W) is focused onto an aluminium foil which is free to move. A 32x32 PhotonPhorce SPAD camera is placed in front of the aluminium foil, at a distance of 1 meter and collects part of the reflected speckle pattern. In the first experiment, a speaker was placed behind the aluminium foil and it was playing heartbeat sound. Subsequently, the aluminium foil was replaced with my wrist, which is placed 50 cm from the camera. The laser power had to be increased from 50 mW to 2 W when my wrist was used as the vibrating surface. From my wrist we managed to retrieve my heart beat sound 5.10.

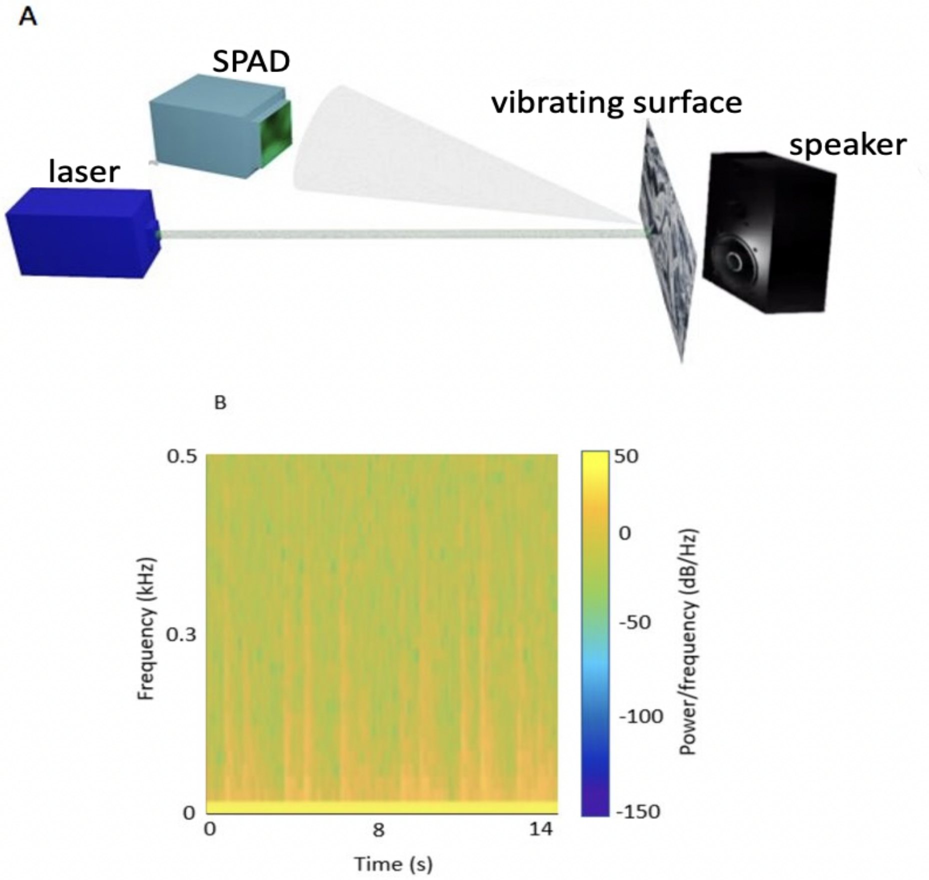


Figure 5.9: A shows the experimental set-up: a CW laser is focused into an aluminium foil. This results in a diverging speckle pattern directed towards the SPAD, which collects part of it. A speaker is placed behind the aluminium surface and as it play it vibrates the surface slightly which in turn tilts. This tilt produces a shift in the position of he speckle pattern.
Figure 5.9 B shows the spectrogram (frequency vs time) of the sound retrieved: heartbeat sound played on the aluminium foil with the speaker.

Results

For the first proof of concept, we collected speckle patterns which were shifting because of the vibrations of the speaker placed behind the aluminium foil. The successive step consisted in summing up all values of each frame acquired with the SPAD camera (integration). This amplitude vs time signal was the recovered sound. The spectrogram of this time signal is shown in Figure 5.9 B.

When instead of collecting the heartbeat sound from the speaker, I collected my heart sound from my wrist, a few considerations need to be discussed. While the aluminium foil is fairly stationary, and only tilts at a small angle while oscillating around a central point, it was more difficult to collect sound from my bare wrist. Small movement of my arm caused the laser beam to land on sightly different positions on my wrist and thus output differently shaped speckle patterns. Speckle patterns are nanometer-scale-variations sensitive as we have already shown. In addition, the speckle pattern changed in shape due to the internal movement of the blood which caused decorrelation. This happens because when the speckle pattern is being produced by a mixture of moving and stationary scatterers, or of scatterers with varying velocities, the speckle pattern is not stationary but experiences a depth of modulation of the speckle intensity fluctuations [147]. When retrieving sound from the wrist it was in fact only possible to retrieve a few heartbeat before the signal was lost due to speckle pattern decorrelation and movement of the arm.

The spectrogram of the sound retrieved by the wrist by integration of all amplitude values of all pixels is shown in Figure 5.10 A. For comparison it is shown the spectrogram of the sound retrieved when the signal is extracted by 1 single pixel in Figure 5.10 B. The SNR decreases so much no sound can be retrieved.

5.7.2 In-light-of-sight sound retrieval through diffusive media

Here we consider the more complex situation in which an additional scattering layers sits between the vibrating system and the detection system. Such situations may arise when for example there are additional translucent layers between the observer and the surface or the surface is beyond the direct line-of-sight and is accessed by scattering from an additional surface.

The additional surface brings us in a regime where, contrary to all previous studies, there is no speckle “memory effect”, i.e. the speckle pattern is not shape-invariant as there is no relation between the various speckle images as the surface vibrates. Thus, we cannot rely on the linear correlation between the speckle pattern shift and the amplitude of sound, as in previous techniques.

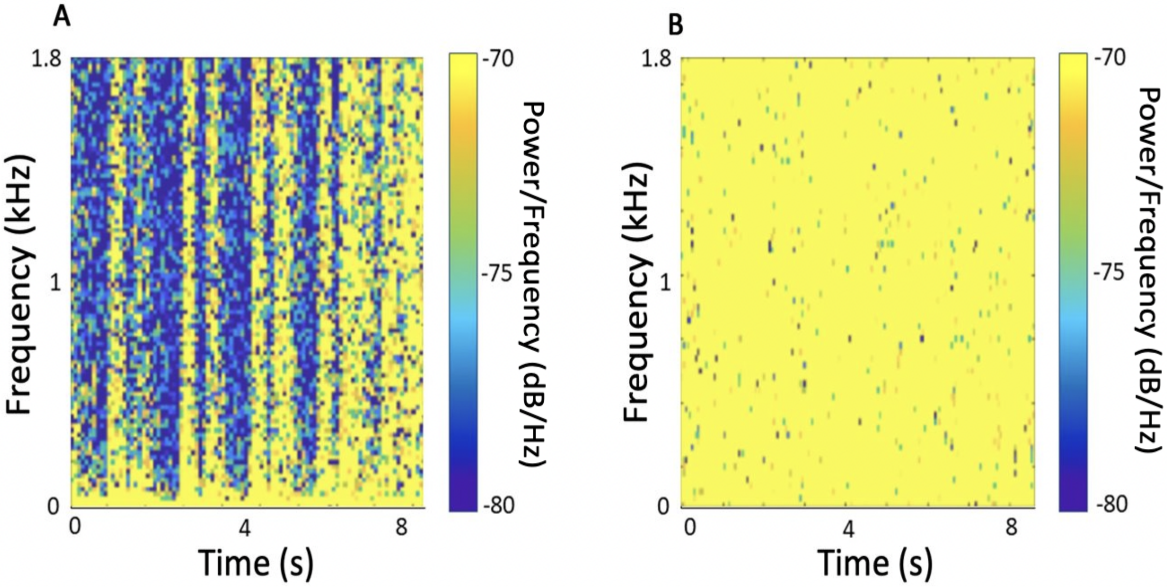


Figure 5.10: A shows the spectrogram of the frequencies retrieved at the wrist with our device and post processing method.

Experimental Set-up

As shown in Figure 5.11, the laser beam (Laser Quantum Gem, $\lambda = 532nm$, max. power = 2 W) is focused onto a glass diffuser. The diverging scattered speckle pattern is then incident upon a roughened aluminium foil placed at 10 cm distance. We then considered two different scenarios: one in which the reflected speckle pattern is directly collected by the camera and a more complex one in which the speckle pattern goes back through the scattering surface a second time on the way back before being detected by the camera.

The camera - used in its photon counting mode - is a single-photon avalanche diode (SPAD) array (Photon Force PF32, 32 X 32 pixels, pixel pitch = 50 μm and fill factor = 1.5 %). We used an acquisition frame rate of 3 KHz, chosen as a good compromise between exposure time (to increase collected light intensity) and frame rate (with sufficient bandwidth to reproduce audible sound frequencies). We measured the return signal resulting from various sound signals that are played back by a small speaker placed behind the foil, including for example a recording of a heart beat.

The sound from the vibrating surface is retrieved by integrating the intensities recovered in each frame into a single value per frame and then using the matlab sound function to play back the resulting values. The spectrograms of the ground truth and of the un-processed (raw data) sound recovered with our set-up are shown in Figure 5.12 a. The main features of the heartbeat

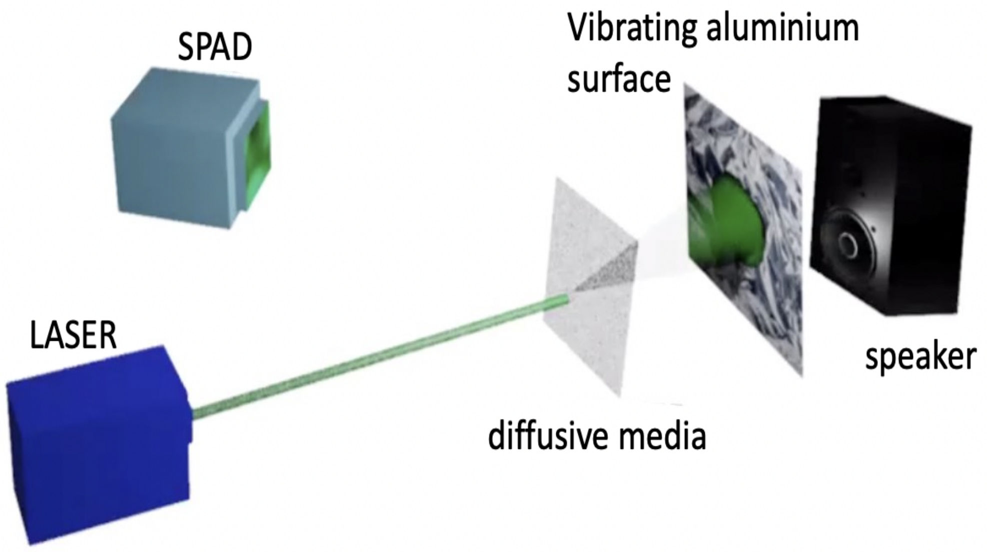


Figure 5.11: Experimental set-up: a CW laser illuminates the vibrating surface after passing through a diffuser. The reflected light, which diverges as it spreads, is detected by the SPAD. We tested the experiment in the case where the reflected light was directly collected by the SPAD with the same configuration as shown in the figure and we also tested the regime where the laser and the SPAD are one on top of the other and the light goes through the diffuser twice before being collected by the SPAD.

are clearly discernible in the measurements and can clearly be heard when simply playing back the spatially integrated camera recording intensity.

Results

The spectrograms in Figure 5.12 show in Figure 5.12 A the original heart sound spectrogram retrieved from the speaker while Figure 5.12 B shows the spectrogram of the retrieved sound that we extracted from the collected light which has passed through 2 scattering media. As it can be seen, the main features of the heartbeat are clearly discernible in the measurements and can clearly be heard when simply playing back the spatially integrated camera recording intensity.

5.7.3 Non-light-of-sight sound retrieval

In this final set-up we placed the vibrating surface outside the line of sight of the camera. The speckle pattern accesses the vibrating surface by bouncing off an intermediate surface.

In the real world, a practical example of a situation where this set-up might be useful would be when trying to assess if a person inside a house that is on fire is dead or alive. It will also allow to listen to remote cellphone conversations without being seen.

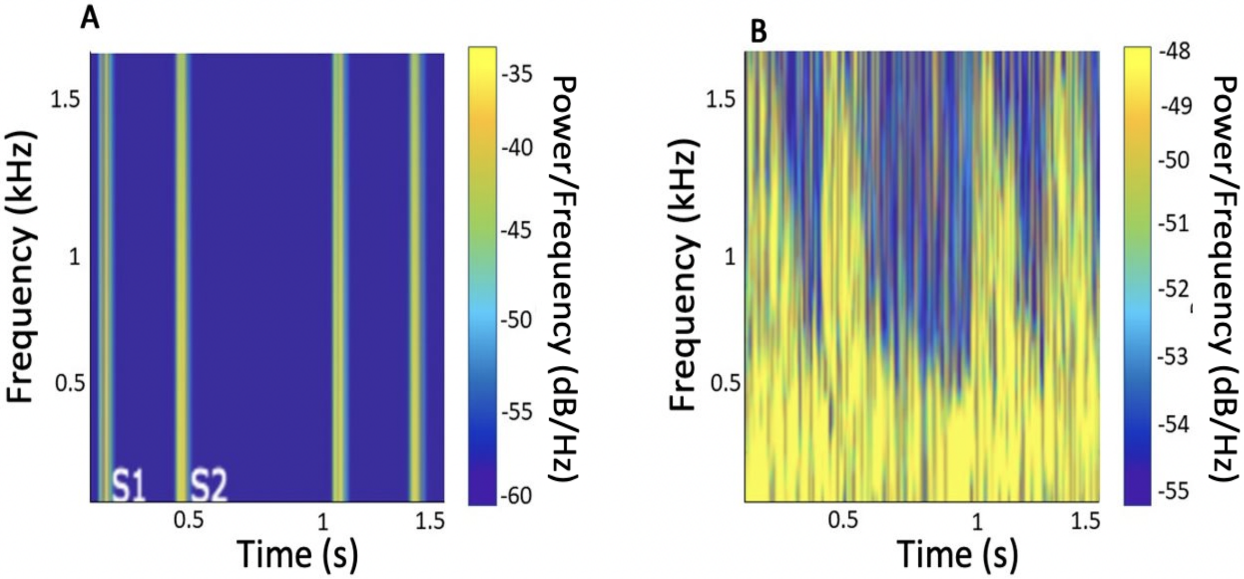


Figure 5.12: (a) Spectrogram of the ground truth of a heartbeat. The spectrogram shows the typical double 'beat 'or first (S1) and second (S2) heart sound, which are repeated periodically (only two heartbeat events are shown). (b) Spectrogram of the same sound signal recovered with the SPAD camera.Although significantly noisier than the original file, both first (S1) and second (S2) heart-sounds are clearly recognisable [148]

Experimental set-up

As shown in, the laser beam (Laser Quantum Gem, $\lambda = 532nm$, max. power = 2 W) is focused onto a stationary roughened aluminium wall, chosen for its reflectance. The speckle pattern scattered from the aluminium wall is reflected back to hit upon a vibrating aluminium foil placed outside the field of view of the camera. Behind the foil is placed a speaker which plays heart valve sound. The reflected light is then collected by the camera sensor. The camera - used in its photon counting mode - is a single-photon avalanche diode (SPAD) array (Photon Force PF32, 32 X 32 pixels, pixel pitch = 50 μm and fill factor = 1.5 %).

Results

Figure 5.14 shows the heartbeat sound retrieved when the vibrating surface was a cellphone which was playing a heartbeat noise with a volume setting so low that it was impossible to hear unless one placed their ear on the phone. The set up to acquire this data was the same used in Figure 5.13 except that instead of the aluminium vibrating surface and the speaker which vibrated it, a cellphone was used as vibrating surface which vibrated on its own.

5.7.4 Energy distributions simulations

In order to use speckle patterns to retrieve sound, both Zalevsky, Zhu, Chen and Veber's groups, which use amplitude modulation and speckle tracking, need to either post process their signal

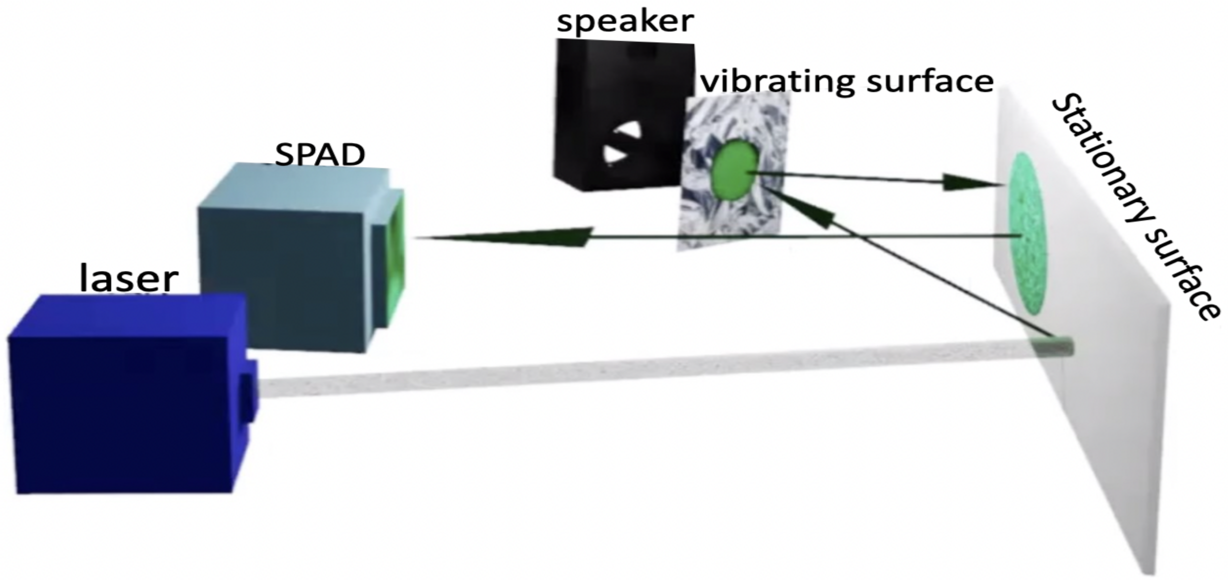


Figure 5.13: shows the experimental set-up used to retrieve sound from a location which was not in the direct line of sight of the camera but from which light had to bounce off multiple surfaces before being detected. A laser points towards a stationary aluminium surface, the arrow that starts at the position where the laser hits the surface and ends on the circle on the vibrating surface shows the size of the speckle pattern as it has diverged in its path. The large speckle pattern is then directed to hit back the stationary surface. After that it lands on the SPAD.

after the acquisition or need special masks which need to be fabricated depending on the conditions of the experiment thus could not use the device spontaneously when needed without knowing the parameters of the experiment before hand.

Using post processing means that these methods to retrieve sound from light cannot be defined as microphones but rather they can only be described as recorders as they do not produce sound in real time.

Veber's work also suffers from some limitations that make his device a microphone only for very specific situations. That is because he needs to prefabricate a mask to be the size of the mean average size of the speckle pattern. That means he needs to know in advance how big the mean average size of the speckle pattern is going to be. In order to do that, one needs to calculate the distance between the vibrating object and receiver and also know in advance the roughness of the material and how much the material is going to vibrate in mm in order to make the spacings in the mask accurate. This severely limits the application of Veber's set-up.

While by integrating we show that we can use speckles as an external diaphragm for our laser microphone to function as an actual real time sound recorder without post processing and at any distance or roughness of surface with simple frame intensity integration over time.

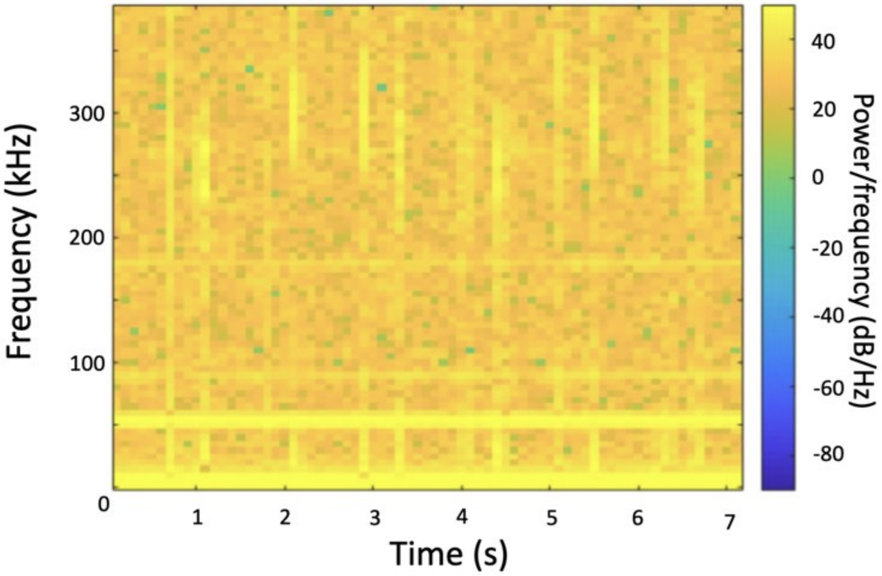


Figure 5.14: shows the spectrogram obtained from a cellphone playing heart valves sound which is playing behind a corner outside the field of view of the camera.

But why does such a simple technique work? When we integrate all the pixels from a single frame while collecting a speckle pattern in line of sight of the camera, the speckle pattern doesn't change shape it just shifts in space in the transverse plane. Thus the total amplitude that falls inside the camera is modulated by this lateral shift. But when we are collecting light from a speckle pattern that has bounced off three scattering surfaces and from a vibrating surface outside the field of view of the camera, therefore no memory effect is retained, how is it possible that sound is still heard simply by integrating over all pixels of the detector?

Our primary idea was that yes, there is no memory effect left, so the shape of the speckle pattern changes from moment to moment, however the energy redistribution is what now gives us the sound. To illustrate this better, one just needs to think about the case where we had a SPAD detector so big as to collect the full speckle pattern. In that case if we integrated over time we wouldn't hear anything. However if we selected a small area, because the speckle pattern changes shape, the energy is redistributing over that space, then we would hear sound due to the local redistribution of energy.

In the next section we will show experimental simulations of the experiments described in the previous sections that we performed using Matlab.

The code used to create these simulations is shown in Appendix C.

5.8 The Memory Effect

This section will describe the speckle memory effect, which has been exploited to retrieve sound from light.

Speckle patterns are created when coherent light hits a scattering surface and the reflected waves interfere constructively and destructively giving shape to patterns of darkness and lights.

In order to retrieve sound from light we measured both the change in total intensity of a region of the speckle pattern formed by shining a laser light towards a vibrating surface or alternatively, depending on the situation, we also tracked the movement of the speckle pattern which under certain conditions remained fixed in shape while shifting laterally in time. This phenomena, where the speckle patterns remains fixed in shape but shifts laterally is due to the memory effect.

In order to capture this reflected speckle pattern the camera is not focused directly on the vibrating object but rather it can be focused in field with respect to the object so that the speckle pattern falls upon the camera rather than the object itself which we are not interested in. As we have seen in Chapter 5, in the close field the speckle pattern changes shape with distance but in the far field it is stationary if the vibrating object only slightly tilts but does not change shape. This can be proven by looking at the equation of the diffraction of light in the far field regime which can be approximated through Fraunhofer equation. Fraunhofer is just what the speckle pattern would look like at infinity, so the speckle pattern does not change in shape in the Fraunhofer regime because it is just a Fourier Transform between the phase mask and incoming light.

$$U(x,y) \propto FT(lightpattern \circledast (Aperture(u,v))) \tag{5.21}$$

Therefore a tilt in the transverse plane, which only changed the phase of the aperture with respect to the light pattern, but does not change the roughness as seen by the light pattern, does not cause a change in the speckle pattern but only a later shift due to the reflection of the light at a slightly different angle.

5.9 Fresnel and Fraunhofer approximations

In order to simulate the above experiments, we need to simulate what happens to a light beam as it hits scattering surfaces. The Huygens-Fresnel principles determines what the Intensity of the light will be after it goes through an aperture and the result is given by the formula [149]

$$U(x,y) = \frac{z}{(\lambda \times j)} \times \int\int \frac{(\xi,\eta)^{jkr}}{r^2 d\eta d\xi} \tag{5.22}$$

This equation expresses the field as a superposition of spherical waves originating from secondary sources at every point of the aperture. In this equation, λ is the optical wavelength; k is the wavenumber, z is the distance between the centers of the source and observation coordinate systems; and r is the distance between a position on the source plane and a position in the observation plane. When the source and observation planes are parallel planes this expression can be expressed as a convolution

$$U(x,y) = U(\eta,\xi)h(x-\eta,y-\xi)d\eta d\xi \tag{5.23}$$

where h is

$$h = \frac{z(j \times \lambda)^{(jkr)}}{(r^2)z} \tag{5.24}$$

and

$$r = \sqrt{x^2+y^2+z^2} \tag{5.25}$$

In the near field the diffraction pattern will follow the Fresnel approximation. In the far field it will follow the Fraunhofer approximation.

Fraunhofer is just what the speckle pattern would look like at infinity, so the speckle pattern does not change in shape in the Fraunhofer regime because it is just a Fourier Transform between the phase mask and incoming light.

$$U(x,y) \propto F(Aperture(u,v)) \tag{5.26}$$

Fresnel diffraction is valid everywhere from the aperture itself, out to an infinite distance beyond it. In the special case of an infinite distance, the Fresnel pattern is the same as the Fraunhofer pattern. Right next to the aperture, the pattern is essentially the illumination pattern masked by the aperture. As your observation plane moves farther and farther away from the aperture, it starts to look less like the aperture and more like the Fraunhofer pattern. This means that in the Fresnel regime you will have a distance dependence while the Fraunhofer formula does not provide it because it's just a Fourier Transform. The larger the aperture is with respect to the wavelength of light, the closer the Fraunhofer zone will be. In fact in the special case where the amplitude was gigantic the transmitted light pattern would very quickly be just essentially the shadow of the hole, no matter how far you were from the hole (for collimated light of course).

Modelling of the propagation

In order to prove that the sound can be heard because of energy redistribution we wanted to firstly simulate the propagation of a virtual speckle pattern through the same experimental setup we had in the lab. The reason for this is because we just wanted to make sure that we did not hear the sound because of some mistake in the lab that we could not trace but we were actually supposed to hear the sound.

So we created with Matlab the same experimental setup that we had in the lab. The stationary surface that is shown in the experimental set up in Figure 5.13 and the skin were modelled according to parameters DCC = 40 μm and DSC = 20 μm [150]. A visual representation of these parameters is given in Figure 5.15.

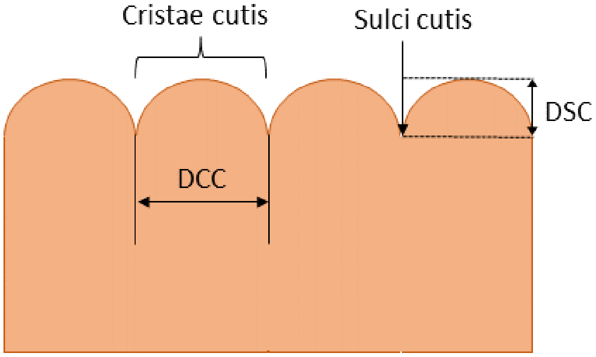


Figure 5.15: shows the parameters DCC and DSC of the skin. These parameters can be used to simulate the roughness of the skin.

A coherent green laser beam is focused to a region of 1 mm in the first stationary aluminium surface (the representation in the simulation of this is shown in Figure 5.16 where "Laser Beam", is simulated by a gaussian function, the stationary aluminium surface is labelled "Stationary Surface"), it then diffracts following Fresnel model in the near field and then Fraunhofer in the far field, which is the region where the speckle pattern's shape is stationary.

To model the speckle pattern created by the interaction of a laser beam on skin-like surface Fraunhofer propagation will be used because we are interested in the speckle pattern created in the far field. In this regime we can approximate Fraunhofer equation as the Fourier Transform of the intensity (proportional to the electric field squared) of the laser beam multiplied by the skin-simulated surface, which is a phase mask. This results in a speckle pattern in the far field which is incident on a second surface (now the speckle pattern will be larger because it diverges with distance) modelled by the skin parameters, however the specific surface configuration is not shown (in Figure 5.16 the visual of the speckle pattern's shape at the second surface is labelled as "First speckle pattern seen at the stationary surface" and the second surface is labelled as "Vibrating Surface". This is the vibrating surface).

In order to simulate the speckle pattern resulting from this first speckle pattern hitting the vibrating surface once again an approximation to Fraunhofer diffraction will be used and it will just be a Fourier Transform of the vibrating surface times the speckle pattern.

Since the surface is vibrating in time, and it is the vibration in which the sound is encoded, the simulation will need to perform this Fraunhofer approximation for all displacements of the surface. The surface is made to vibrate (shifting left and right - compared to the camera - without loosing shape) at the same frequency as a chirp sound, which is the sound we are trying to recover in this instance.

An example of a sound retrieved is shown in Figure 5.19 B. This is the amplitude vs time plot of the sound recording played. For every time frame (moment in time) there is in the graph an amplitude value which corresponds to a sound amplitude. The vibrating surface was shifted laterally in space left and right from the origin with an amplitude variation proportional to the amplitude of the sound. The vibrating surface was made to shift as many times as the number of time frames in Figure 5.19 B to simulate the response of a surface to sound. This second speckle pattern upon hitting the vibrating skin, scatters again into a more complex speckle pattern because each previous speckle of the speckle pattern is giving rise to secondary speckle patterns which adds and subtracts destructively and constructively with all the neighbouring speckle patterns. Now our second speckle pattern reflected from the skin (which is not actually just one speckle pattern but there are as many speckle patterns as time frames of amplitudes of sound) is incident upon the aluminium wall once again (labelled as "Stationary Surface"). After a third Fraunhofer approximation of the propagation of light, the speckle pattern is finally incident upon the SPAD. Once again, after the vibrating surface, these new speckle patterns (one for every time frame), are shifted laterally the one with respect to the other so the simulation with the encounter with the third surface requires a Fraunhofer approximation between the third surface and all the lateral shifts of the speckle pattern.

As previously hypothesized, when the full speckle pattern is incident upon the collecting camera (in the simulation we assumed all the intensity of the initial laser beam reaches the collecting camera) and the speckle pattern is integrated through time, no sound can be discerned as the energy is constant over time when all the light is collected. However, when only a smaller window of the full speckle pattern is captured, the sound can be heard. This demonstrates that it is the local redistribution of energy to give rise to the sound.

Another important observation is that there is a divergence of the speckle pattern through space.

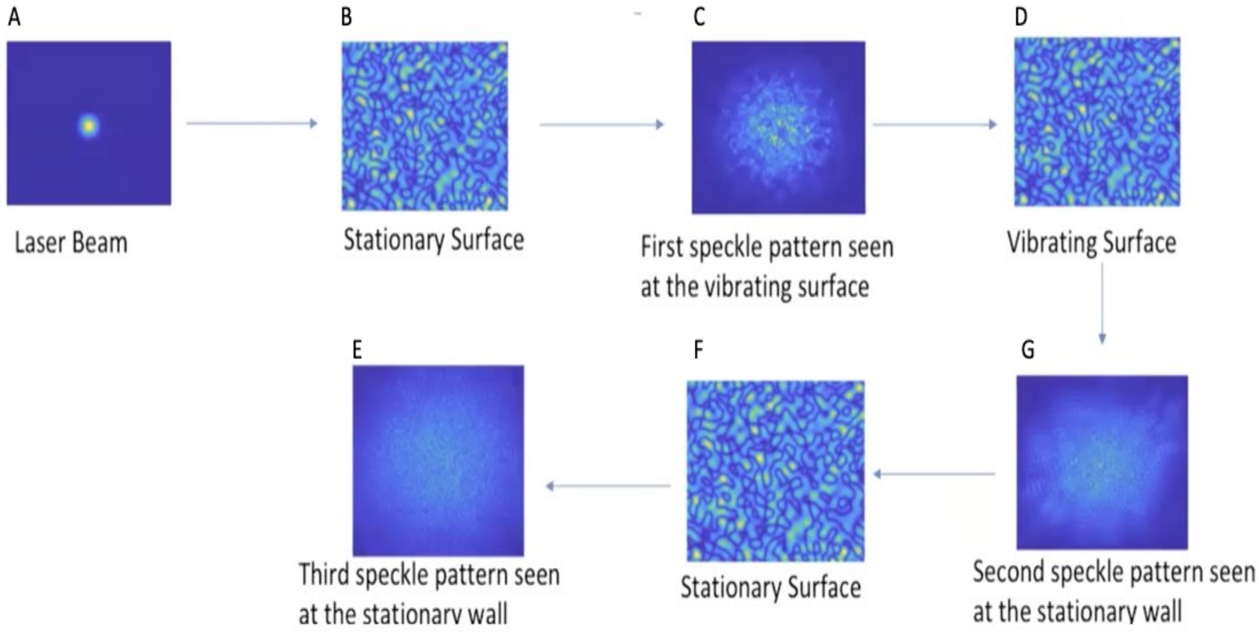

A
Laser Beam
B
Stationary Surface
C
First speckle pattern seen at the vibrating surface
D
Vibrating Surface
G
Second speckle pattern seen at the stationary wall
F
Stationary Surface
E
Third speckle pattern seen at the stationary wall

Figure 5.16: shows the various final stages of the simulation of the experimental set up shows in Figure 5.13. a laser beam which is approximated by a Gaussian function (Figure A) is incident on a stationary surface which has been approximated as a random distributions of roughness in the x and y direction (Figure B). After propagation the speckle patterns will hit the vibrating surface (Figure C). Then speckle pattern will then hit the third stationary surface (Figure D and E). Then it will be collected by the camera (Figure G).

This is shown in Figure 5.17.

Figure 5.17 A shows what the simulated speckle pattern looks like in the Fraunhofer regime after it has been scattered off the first surface. Figure 5.17 B shows what the speckle pattern looks like in the Fraunhofer regime after it has been scattered off the second surface and Fig-

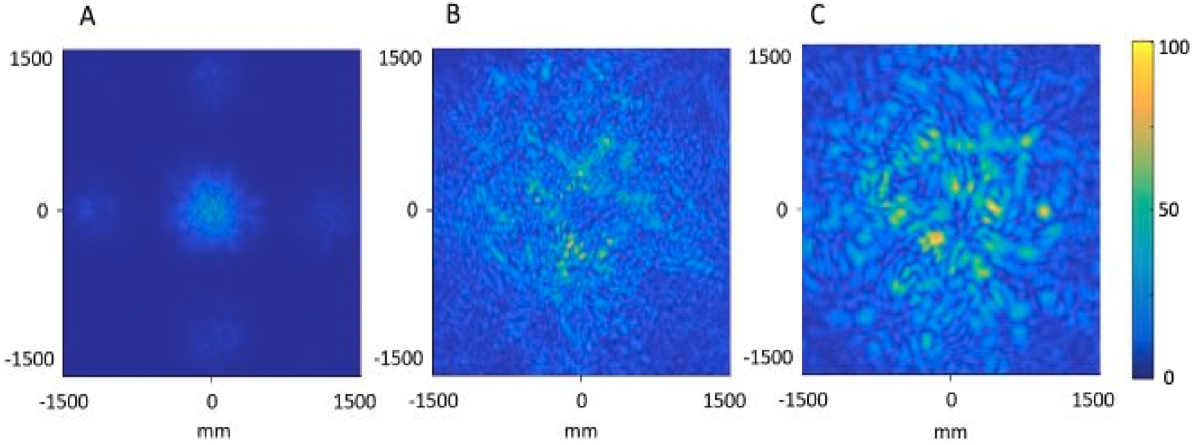


Figure 5.17: shows a speckle pattern which increases in size as it diverges.

ure 5.17 C shows what the speckle pattern looks like in the Fraunhofer regime after it has been scattered off the third surface. In the Fraunhofer regime, as shown in the previous section about the energy simulations, the speckle pattern does not change shape anymore however it diverges. The speckle pattern changes shape once again as it hits another rough surface. This is the reason why in the figure the speckle sizes become increasingly smaller and the structure more complex although after propagating the speckle pattern's total size increases.

We also included in the simulation how the vibrating surface moved with respect to the speckle pattern at the frequency of various sounds. Then the speckle pattern moved perpendicularly with respect to the third scattering surface and finally it moved at the same frequency as the sound with respect to the sensor's aperture.

We performed two simulations, one in which the speckles size were smaller than the pixels sizes and one in which they were bigger. In both cases sound was retrieved. Even if the shape of the speckle pattern completely changes from time frame to time frame, as shown in Figure 5.18, by then using our simple integrating method we managed to retrieve the sound. It was only when we integrated over the full size of the speckle pattern that the we did not retrieve any sound. This is because we included the total energy of the speckle pattern, with the assumption that we were not loosing scattered or absorbed photons anywhere. Thus, since we were collecting the total energy of the system and integrating it through time, we could not detect the energy redistribution. This confirmed our initial guess that we were able to retrieve sound by detecting the energy redistribution in a small part of the speckle pattern.

Figure 5.19 B shows what the retrieved sound looks like when it is portrayed in its simplest form as amplitude versus time for a small window of acquisition of the speckle pattern, while Figure 5.19 B shows again the retrieved sound as amplitude versus time when the all speckle pattern is considered. The amplitude over time of the latter stays constant as expected because energy is conserved.

We have tried various methods of how the vibrating surface could have been affected by the sound, with lateral and transversal shifts and even the case of when the vibrating surface changes completely between frame to frame. Through integration, no matter how the surface was affected due to sound, the sound could be retrieved.

5.9.1 Optimal Acquisition Method

Hitherto we have seen that it is possible to retrieve sound from light by either tracking the movement of the speckle pattern (in the cases where there is memory effect) or we can retrieve sound by simple integration which, not only works for the in line of sight regime, but it also works for

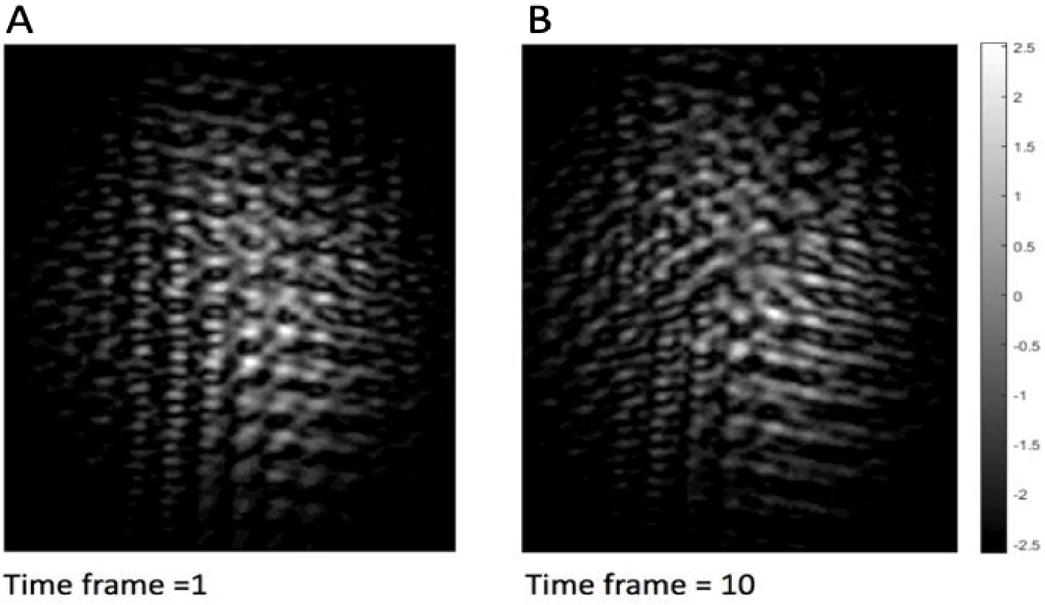


Figure 5.18: shows two speckle patterns obtained from the incidence of a laser beam simulated by a Gaussian with two mildly rough surfaces. The speckle pattern in the Fraunhofer regime, if the surface is subject to movement due to sound will produce a change in the shape of the speckle pattern from frame to frame.

the non line of sight regime and scattering media regime.

In the next two chapters we want to exploit this technique to acquire data from 10 subjects in a in line of sight configuration, thus we want to check what method amongst these two gives the best SNR versus distance for the in line of sight acquisition regime. In order to do so we set up the experiment as shown in Figure 5.19.

Experimental set-up

The experimental setup built to determine the best retrieval method and experimental conditions for recovering sound vibrations from the speckle dynamics is shown in Figure 5.20. A laser diode (DJ532-40 Thorlabs) is directed towards a thin PTFE membrane which was used to mimic the skin surface. A fast Basler camera collects the resulting dynamic speckle pattern at a sampling frequency of 900 Hz. The membrane was actuated by a loudspeaker, which played a pure tone of 300 Hz.

Method

The SNR was determined as the average amplitude of the f=300 Hz sound spectral peak divided by the variance of the signal across the whole sampling region $0-\frac{f}{2}$. This was done over a 1 second recording. In order to then check SNR vs distance we acquired different measurements where we changed the distance from the camera to the vibrating surface to allow to vary a num-

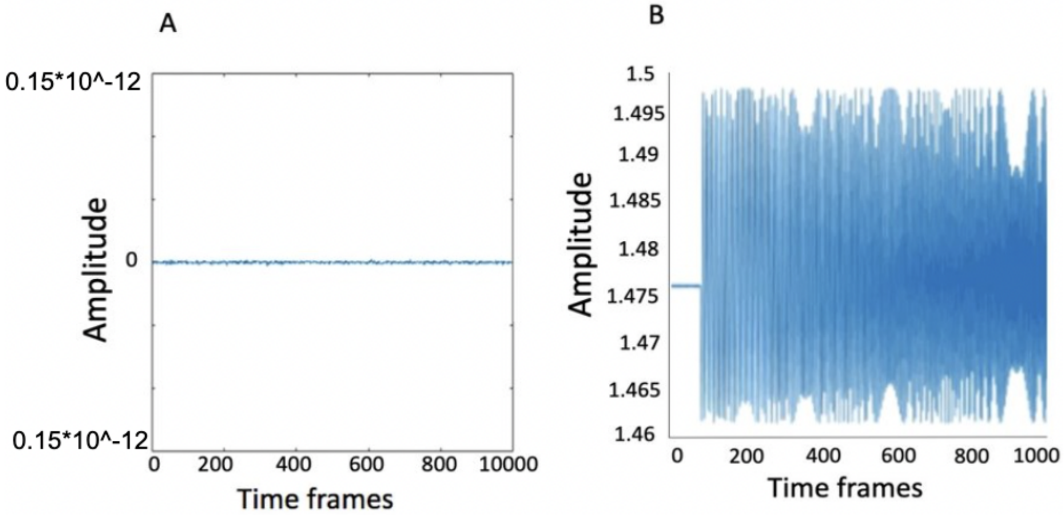


Figure 5.19: A shows the retrieved sound obtained in the simulation by integrating the speckle pattern moving due to sound. Because the full size of the speckle pattern has been integrated, as expected, no sound is retrieved because the total energy is conserved. Figure 5.19 B shows the sound retrieved from integrating a part of the speckle pattern obtained from the simulation. As it can be seen the sound has been retrieved because the energy gets redistributed proportional with the frequency of vibration of the surface which vibrates due to the sound.

ber of parameters that could affect the SNR of both retrieval methods.

First of all, since we have seen both by eye during experiments and through the simulations with results shown in Figure 5.17 that the speckle average grain size linearly increases with distance, also the speckle displacement amplitude due to surface vibration increases. On the other hand the average speckle intensity decreases with distance, which leads to noisier images. The dependence of the SNR for both methods on the camera-surface distance is shown in Figure 5.21.

Results

Figure 5.21 shows the comparison of the SNR obtained from the sound retrieval with the integration method versus the tracking method for 7 locations of camera-sensor. The error bars correspond to the variance across 10 different positions of where the laser hit the scattering surface which in turn generated different speckle patterns configurations (this has also been tested during the simulations, where, for a small change in position of the laser versus scattering surface, the shape of the speckle pattern changed).

The integration method (red square markers) shows a clear maximum at 60-80 cm. It is roughly the distance at which average speckle size becomes comparable to the field of view of the camera. At smaller distances the SNR decreases because the statistical fluctuations of the integrated

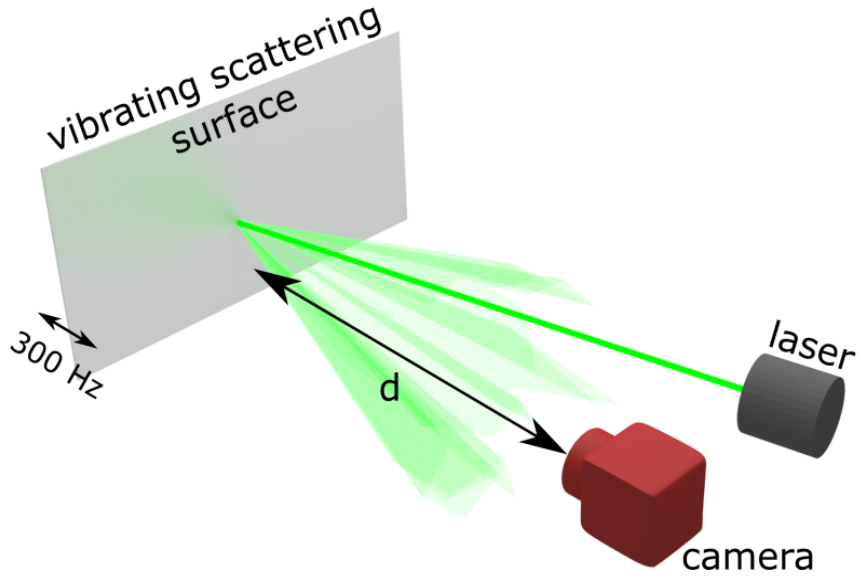


Figure 5.20: shows the setup of the auxiliary experiment to determine the relationship between various parameters to obtain the optimal SNR. The skin vibrations are modeled with a PTFE membrane actuated by a loudspeaker. The laser used is diode laser (DJ532-40 Thorlabs) and the camera is a Basler.

intensity tend to flatten out when more speckles get into the field of view, which is a conceptual limitation of the integration method. In fact, we have shown in the previous section, that if the full speckle pattern is integrated no sound can be retrieved. At larger distances the SNR becomes worse because of the drop in the average intensity.

The SNR of the speckle tracking method (blue round markers) almost monotonously drops upon the surface-camera distance because with bigger speckles there's less distinct features in the camera images and it becomes harder for the tracking algorithm to calculate the displacement.

As can be seen from Figure 5.21, we achieved the best average SNR with the speckle tracking method at a close distance from the surface. This method also showed overall less SNR variance depending on the speckle realizations. Based on that we have selected it for any further experiments.

5.10 Conclusion

In this chapter we have shown that we were able to acquire sound from light in a regime which has been previously unexplored: the non line of sight regime. As mentioned in the chapter, this is a regime where the vibrating surface is not in direct line of sight of the camera, thus there is

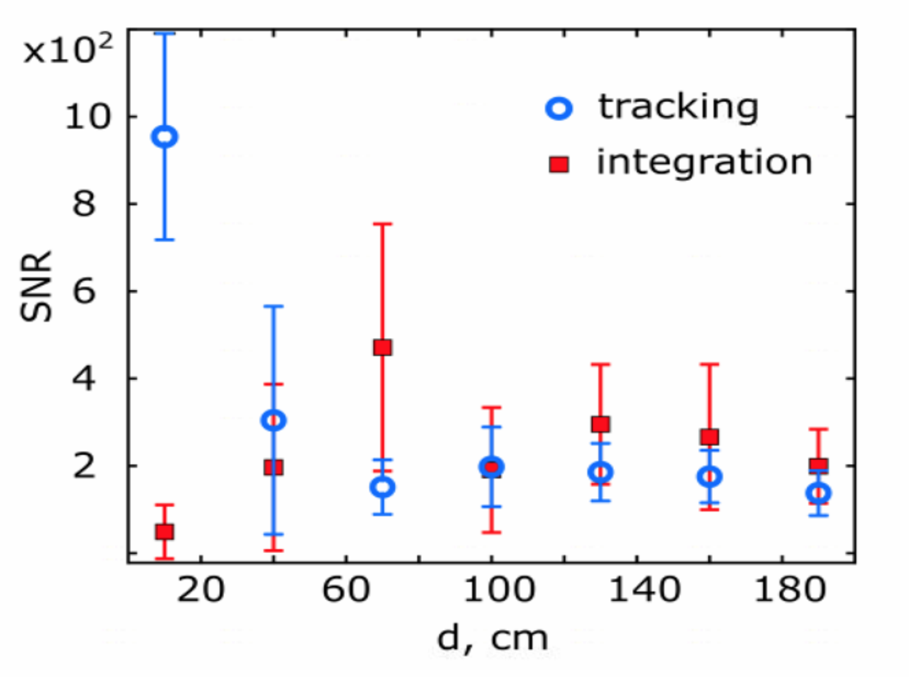


Figure 5.21: shows the comparison of the SNR vs distance of the sound retrieved with the tracking method vs the integration method. At each distance we acquired 10 measurements where in each measurement we changed the laser position with respect to the surface in order to obtain a different speckle pattern shape and get a statistical view of the results.

no retained memory effect from which to extrapolate the sound. Therefore, in order to recover the sound, we used a new(at the time we performed these experiments) way to recover sound from light: integration.

We then simulated the experiments to understand why the simple integration method worked, and confirmed that it was because of a redistribution of energy from frame to frame. This theory is proven from the fact that when the full speckle pattern is integrated, no sound is retrieved.

Successively we have carried out an experiment to find which post processing method between the optical flow algorithm and the integration method gave the best SNR versus distance. We found that the post processing method which gave the best SNR if many speckles fell upon the detector in a in line of sight configuration was the optical flow algorithm.

These results allowed us to pick the optical flow algorithm to use as post processing method for our next two experiments, which consisted of acquiring the reflected speckle patterns from people's necks in a in line of sight configuration.

Chapter 6

Valve Sounds Heart Assessment with Machine Learning

In this chapter the heart valve sound signal that we can retrieve with our device will be shown. We will also show a comparison of what sounds the stethoscope can retrieve from the neck compared to what our device can acquire. We chose the neck as acquisition position because that was the place the acquisition of heart sound with our device showed the less corruption by noise.

In this chapter we will show a comparison of data acquired with the digital stethoscope at the chest and neck compared with data acquired with our device from the chest and neck.

We will then train a model with heart sound data acquired from the digital stethoscope and test the model on a subset of that data to check that it gives a high accuracy. Then we will test if the model trained on stethoscope data can predict correctly on data acquired with our device. Therefore we then test the model on data acquired with our device.

6.1 Methods for Heart Sound Acquisition

As described in the previous chapter, we acquired heart sound data contactlessly from non specific distances by shining a very weak laser (power less than 4 mW) at the frontal region of the neck. This is done by manually pointing our device towards the person's neck and adjusting where the green visible laser dot falls. An interesting application for future use would be to implement an AI algorithm that automatically finds the neck. The back-reflected speckle patterns fall a CMOS camera that records at 1500 Hz.

In order for this device to be able to monitor people constantly, maybe as they go on their daily lives in their houses, data must be acquired from a body part which is free from clothing most of the time. Data must also be acquired from a location which is near the heart. Therefore, to satisfy those two conditions we shone the laser at people's necks. The reason why heart sound data must be collected from a region close to the heart is because the high frequencies of the heart sound are lost the farther away from their site of origin because they dissipate the fastest. In other words, this means, that at peripheral vessels' locations almost all the frequencies of the heart valve sounds will be lost [142]. We acquired heart valve sounds from the base of the neck, where the high frequencies of the heart valve sounds were not yet dissipated.

The 10 subjects were 10 volunteers whose age ranged between 20 to 30 years old and considered themselves as healthy. Since this was the first case study that we undertook and COVID limited movement, we had to work with a relatively small sample size. We asked and obtained Ethic Approval from the University Ethic Committee (application number 300200122). Volunteers signed a form of consent in which it was explained to them why they were taking part on this experiment, what part they would play in it and what would happen to them during the data taking process. The form they signed also stated that they gave their consent for this data to be used in published work and that the data would be anonymized.

The subjects were asked to sit down on a chair and breath normally as we pointed the laser at the base of their neck. It is possible to see the set up of our device from Figure 6.1. The volunteers were asked to wear eye protective goggles which stopped a range of wavelengths which included the one of the laser which was 532 nm. We acquired 4 minutes and 30 seconds of data. The laser power fell in the range of Class 2M, which is safe for the skin. Visible light laser power above 500 mW burns the skin.

As it has been described in the previous chapter, when a coherent laser beam is incident upon a scattering surface the interference of the reflected waves creates a pattern of bright and dark spots which is called "speckle pattern". The shape of this pattern will depend upon the roughness of the surface. When the skin vibrates due to blood flowing and carrying the frequencies of the heart valve sounds the speckle pattern will move in comparison to the skin location it is incident on. As we have seen, when the tilt of the vibrating surface is small enough and the surface isn't deformed by the vibrations, then the speckle pattern's behaviour follows the memory effect and it doesn't change shape but only shifts and such displacement can be tracked with optical flow algorithms. The optical flow algorithm used to retrieve the 1D signal corresponding to the heart valve sounds was the Lukas-Kanade algorithm implemented through Matlab.

In order to avoid overfitting, a feature selection algorithm based on the wavelet scattering transform (algorithm which has been covered in Chapter 3) was applied on the data to reduce its dimensionality. The scattering transform used feature selection to extract a small selection of highly predictive features which are stable, robust and highly informative. This method yielded better results then just applying a CNN, which would have found features tailored to the data. That is because the dataset was small and as described in Chapter 3, the wavelet scattering transform is tailored to extract sound signal. Then the data was passed to a ML algorithm which was used to classify it as either healthy or unhealthy.

The code used can be seen in Appendix D.

6.1.1 Experimental Set-Up

The experimental setup is shown in Figure 6.1 A and B. The figure shows our device's hardware: a diode laser attached to a Basler camera. The diode laser is a DJ532-40 from Thorlabs. The laser is directed at the test subject 's neck and the it is focused in an aera of about 5 mm. The basler camera collects the resulting dynamic speckle patterns at fsamp= 1.5 kHz frame rate. An objective is placed in front of the camera because otherwise not enough reflected light would make it to the sensor without having to use excessive laser power which would burn the skin. The objective has a focal length of 20 mm. The test subjects, while sitting down in a chair and normally breathing, are asked not to move.

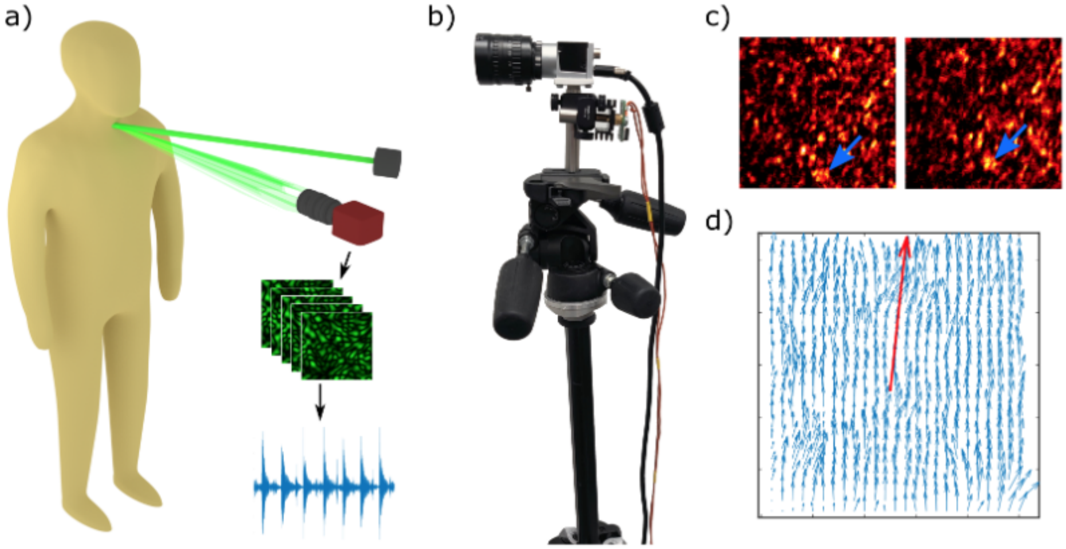


Figure 6.1: A shows the laser diode and the basler camera next to it. The laser points towards the base of the subject's neck and the camera collects the diverging speckle pattern reflected from it. Figure 6.1 B shows a actual picture of the set-up and Figure 6.1 C shows two frames acquired by the camera of two speckles patterns side by side. As it can be seen, the frames are almost identical the one to the other, the only difference is a minimal shift whose displacement is captured by the optical flow algorithm.

6.1.2 Results

In order to compare our results of the retrieved heart valve sound frequencies with the frequencies retrieved with the stethoscope which is the gold standard for auscultation, we took heart valve sounds acquired for the Physionet 2016 PCG challenge [155]. The Physionet PCG data contained heart valve sounds obtained with a digital stethoscope acquired from chest locations above the heart (locations shown in Figure 2.4). The stethoscope data was obtained from subjects which were either healthy or diseased. Heart sound recordings were sourced from several contributors around the world, collected at either a clinical or nonclinical environment, from both healthy subjects and pathological patients. The data was labelled as normal or abnormal and the frequency of acquisition of this data was 2000 Hz.

Figure 6.3 A shows the retrieved heart valve sounds from one of the Physionet data-sets which was labelled as belonging to a healthy individual and which we filtered with a bandpass (the type of bandpass chosen was a Butterworth) filter which allowed frequencies between 30-250 Hz to go through and the results are shown in Figure 6.3 C. As it is possible to see, our retrieved data appears less noisy. In Figure 6.3 B it is possible to see the same dataset as shown in Figure 6.3 A, but now filtered with a bandpass filter which allowed frequencies from 375 to 1000 Hz to go through. As it is possible to see, the quality of the heart valve sound signal if only a high frequency range is considered highly decreases. This agrees with the literature that says that S1 and S2 lie in a frequency range below 200Hz. However as it is possible to see from our data in Figure 6.3 D, with our proposed method we can retrieve heart valve sound frequencies at a frequency range which is much higher than what the stethoscope (comparison with stethoscope data taken from the Physionet 2016 challenge) can retrieve and with a better SNR. Furthermore, the cardiologists we have been working with have told us they do not acquire S1, S2, S3 and S4 sounds from the neck because the SNR is too low to make a diagnosis. These claims are sustained by the fact that the heart sound is attenuated via non linear process as it propagates towards peripheral vessels. Shi et al [153] has shown what happens to the heart sound recovered at various points around the body. Once the sound reaches the wrist, only the lower frequencies (below 30 Hz) are retained. In Figure 6.2 it is shown the frequency content acquired with our device from the wrist. As it can be seen, the frequency content at the wrist is lower than the content acquired at the neck. In contrast, our device acquires even the high frequencies of the heart sound from the neck.

From Figure 6.4 it is possible to see this concept. Figure 6.4 A shows the frequency content of heart sound data acquired with a digital stethoscope. It can be seen that there is no signal after around 250 Hz. Figure 6.4 B shows the acquired frequency range of heart valve sounds obtained from data obtained with our device. As it can be seen there is still a lot of signal past 250 Hz.

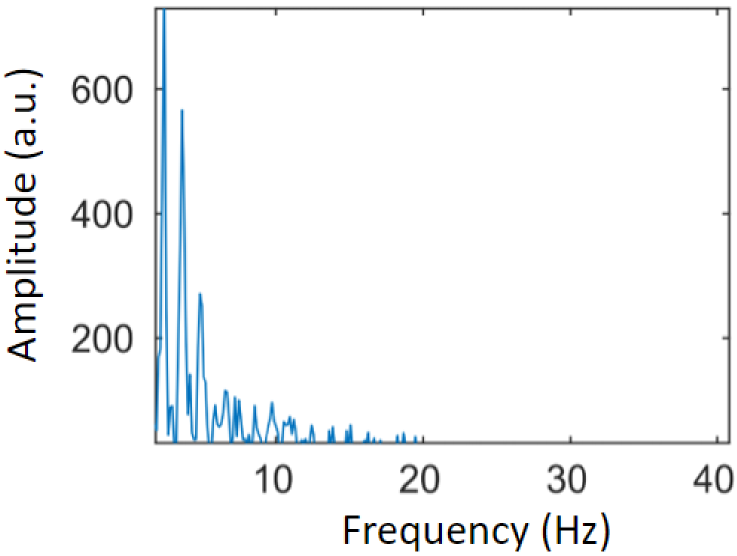


Figure 6.2: shows the frequency content of the heart sound acquired with our device from the wrist.

In order to compare our results to the results that can be obtained through a stethoscope, we bought a Thinklabs digital stethoscope. With the digital stethoscope we weren't able to record sound from the neck, but we managed to do so with our device. It is possible to see in Figure 6.5 the retrieved sound (shown as samples vs time) retrieved with the digital stethoscope at the neck. No heart valve sound is discernible.

In Figure 6.5 A we show the retrieved high frequencies of the heart valve sounds
The last thing to notice about the quality of the signal that we can acquire with our method is the comparison of the amplitude of the low versus the high frequencies of the retrieved heart valve sound. In Figure 6.3 it is possible to notice that the amplitude of the low frequency range of the heart sound, which is the orange signal, is an order of magnitude higher than the amplitude of the high frequency range of the heart valve sound (500-750 Hz) retrieved with our method.
So far we have shown that by collecting and analyzing the movement of the speckle patterns back reflected from subjects' necks it is possible to obtain heart valve sounds.
We also shown that our method can retrieve heart valve sounds from peripheral blood vessels while the digital stethoscope can only acquire heart valve sound signal from specific locations on the chest.

Finally, we have also shown that it was the first time that a very high frequency range of the heart valve sound was acquired from the neck.

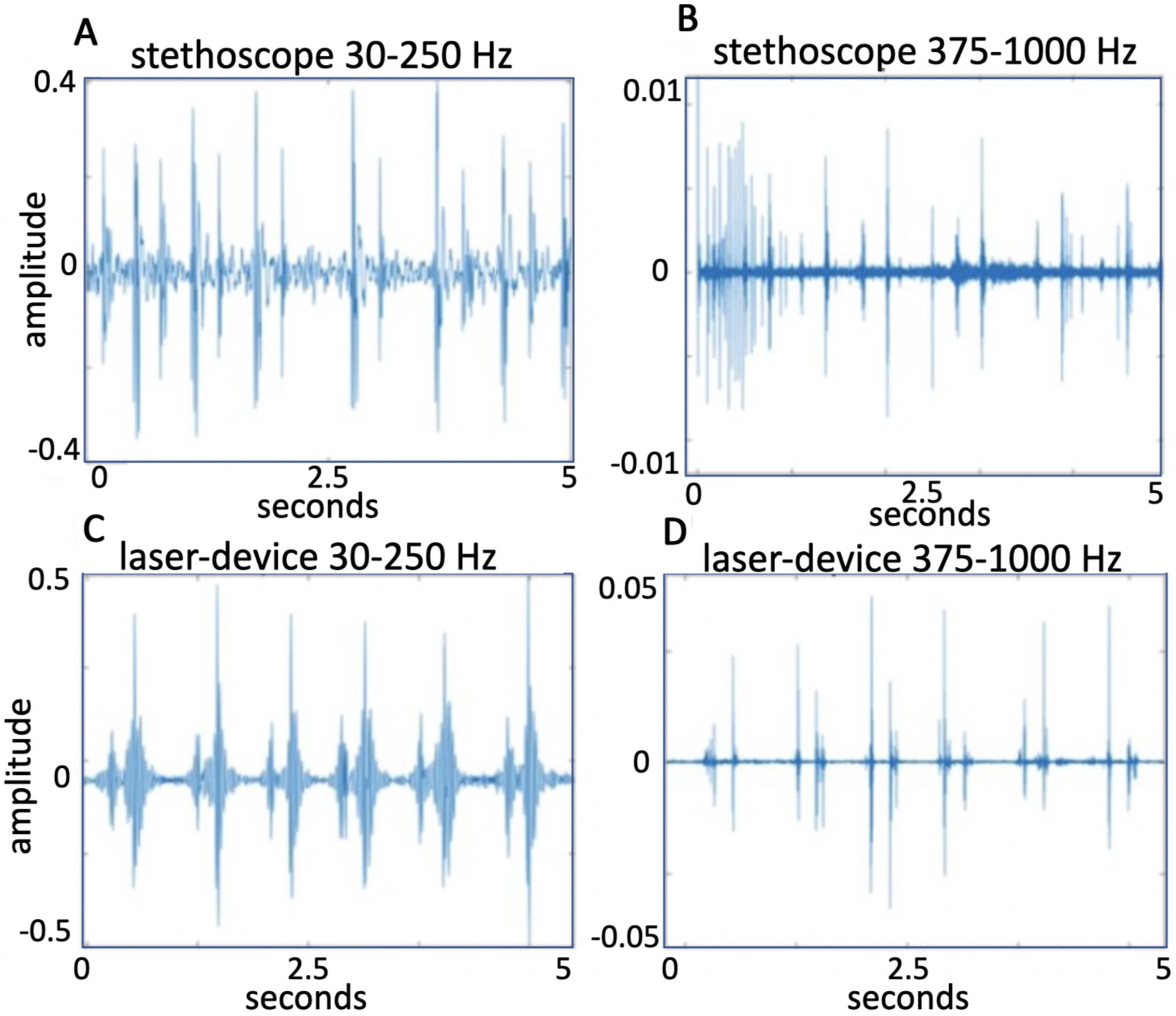


Figure 6.3: A shows the heart valve sounds acquired with the digital stethoscope from the chest location above the heart from a healthy individual. The data-set was taken from the heart valve sound data which was made available from Physionet 2016 Challenge [155]. The data, acquired at 2000 Hz, has been filtered with a bandpass that allowed frequencies from 30-250 Hz to go through. Figure 6.3 B shows the same data-set shown in Figure 6.3 A, but this time the bandpass filter applied on the original data-set lets only frequencies from 375-750 Hz to go through. Figure 6.3 C shows the heart valve sound signal acquired from one of our subjects and post processed with the Lukas-Kanade optical flow algorithm. The signal, acquired at 1500 Hz has been put through a bandpass filter which only allows frequencies from 30-250 Hz to go through. Figure 6.3 D shows the same signal set shown in Figure 6.3 C but this time the original signal set went through a bandpass filter which allowed only frequencies from 375-750 Hz to pass through.

Furthermore, from Figure 6.7 it can be seen that there is not much difference between the heart sounds acquired when the volunteers where asked to breathe normally during acquisition vs when they were asked to hold their breath. Therefore, for a matter of comfort we decided to acquire data while the volunteers breathed normally even if there was slightly more noise in the data. Furthermore, no motion adjustments to the signal is needed when the subjects breaths

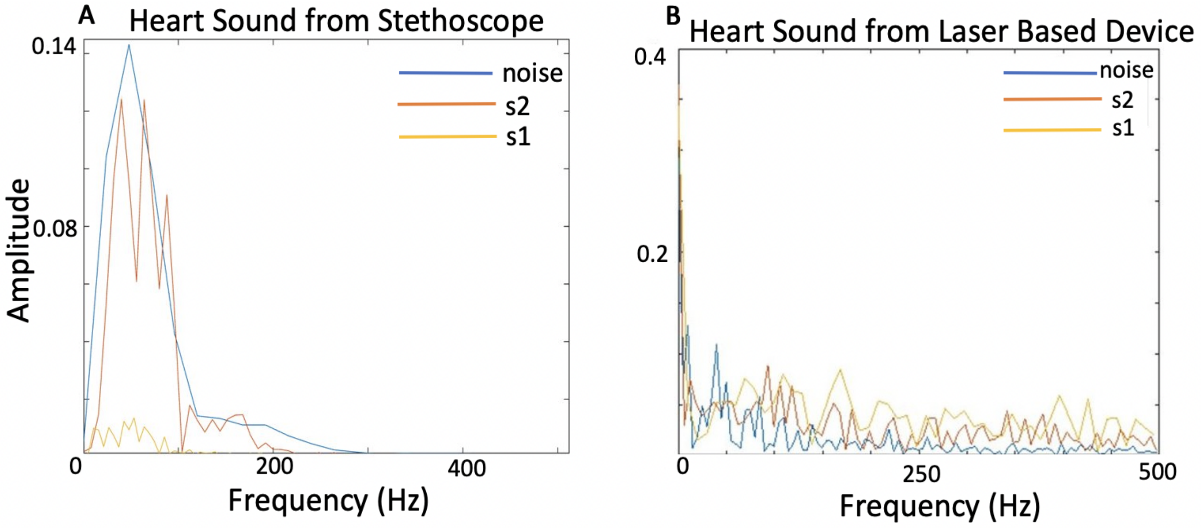


Figure 6.4: shows a comparison of the frequency content obtained from taking a Fourier Transform of data taken with a digital stethoscope (Figure 6.4 A) versus data taken with our device (Figure 6.4 B). As it can be seen, the frequency range of the heart valve sounds lie between 0-250 Hz for data acquired with the digital stethoscope, with no frequency content above that level. While for the data acquired with our device the frequency content extends beyond 250 Hz.

during acquisition.

In the next sections we will show the utility of this newly retrieved heart valve sound high frequency range signal in heart health assessment and biometric authentication.

6.2 Heart Health Assessment Through Heart Valve Sounds

As we have discussed, there is a newly renovated interest in diagnosis through heart valve sounds which was born with the development of the digital stethoscope and advanced sensing technologies. To diagnose patients'heart condition, cardiologists have access to electrocardiograms, chest X-rays, ultrasound imaging, MRI, Doppler techniques, angiography, and transesophageal echocardiography. These diagnostic techniques require a cardiologist’s visit and are expensive, the examination time is long and so are the waiting lists. Phonocardiography is a cost-effective method which records the sounds the heart makes. Many heart diseases cause changes in heart sounds before other symptoms appear.

However, auscultation is not widely used as a diagnostic technique because it requires considerable training and it relies on the hearing abilities of the clinician. Furthermore results are qualitative in nature and not reproducible [154], that is because through manual auscultation, the accuracy of the results is based on the experience of the doctor which makes a choice through analysis of the tone and intensity of the heart sounds. Furthermore, even experts in this discipline could only predict pathology correctly 70% of the time [157]. Therefore, the reason to

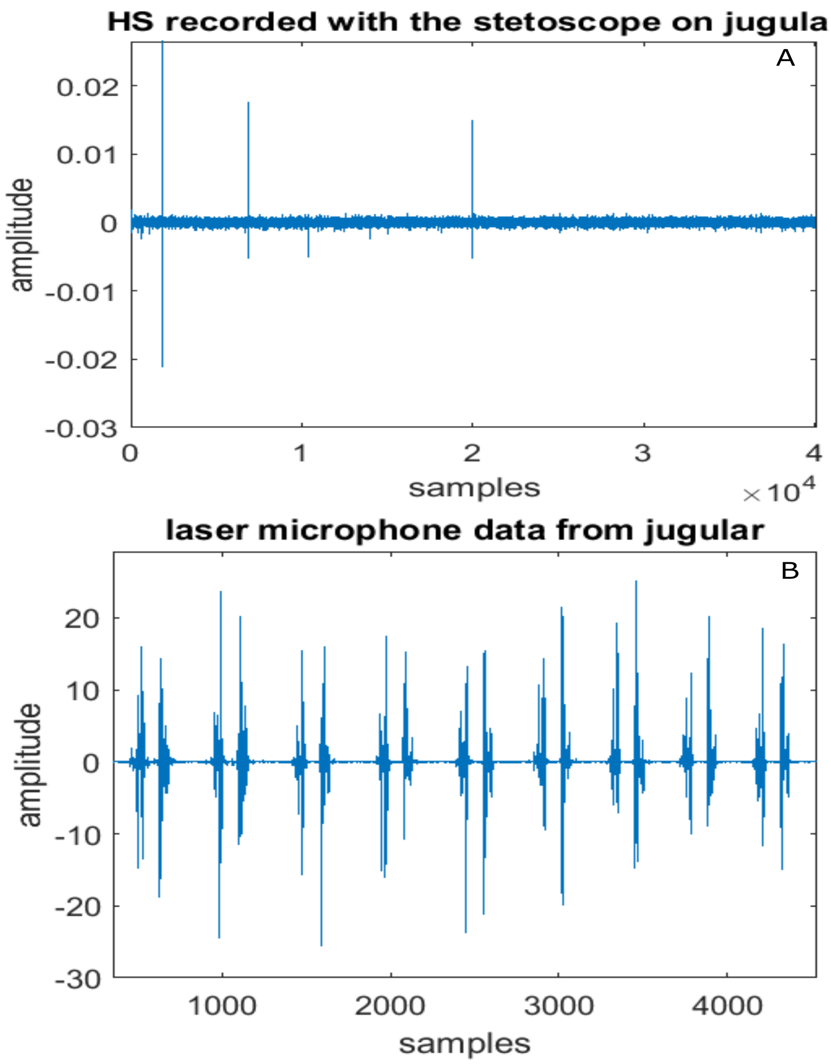


Figure 6.5: A shows the heart valve sound retrieved at the base of the neck with the Thinklabs digital stethoscope. Figure 6.5 B shows the heart valve sound acquired with our method and postprocessed with the Lukas-Kanade algorithm and fast wavelet transform.

move to computer-aided detection techniques for heart sound analysis and classification is to obtain quantized parameters out of the signal which will be a more reliant and stable approach to classify heart diseases.

In the next sections I will present a method to use the heart valve sounds acquired with our device from 10 volunteers to assess their heart condition. The classification will be binary where the output can be either normal if the person is healthy or abnormal if the person is unhealthy. Before feeding the data to a machine learning algorithm to perform the binary classification, the data went through some post processing methods to decrease the dimensionality of the problem.

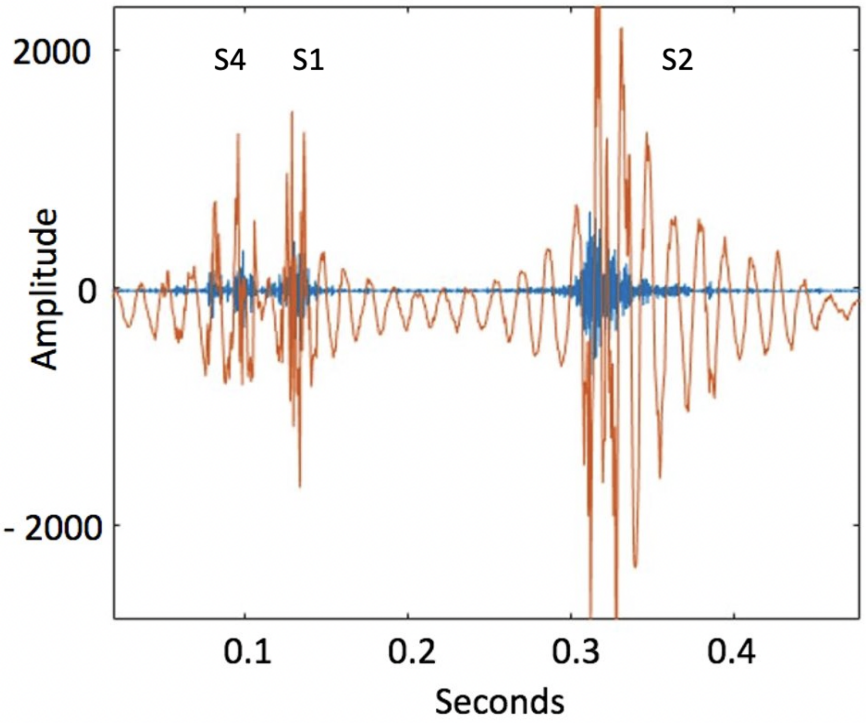


Figure 6.6: shows in orange the full frequency range of the heart valve sound retrieved with our method and in blue the high frequencies of the heart sound, a 500 Hz highpass filter was applied to the signal shown in orange.

6.2.1 Heart Sound Denoising

Heart valve sound signals are usually coupled with sources of noise such as background noise, power interference, breathing or lung sounds, and skin movements in the surrounding environment [158]. The most common methods for heart valve sound denoising are wavelet transform (WT) [159, 198], total variation (TV) [161–163] and empirical mode decomposition (EMD) [166, 171]. Because successful diagnostic accuracy is reliant on the quality of the signal, denoising is essential. Here I will give a brief description of how these three denoising methods work:

- Wavelet Denoising:
  Chapter 3 is dedicated to this item. But to briefly summarize, wavelet denoising consists of three steps: decomposition, thresholding and reconstruction.

  The signal is firstly decomposed with wavelets and then the desired decomposition level is chosen. Then coefficients can be tresholded and finally the inverse wavelet transform is used to retrieved the signal in the time domain.

  In order for this process to yield optimal results, careful consideration, which will depend on the characteristics of the signal at hand, care must be given to the parameters

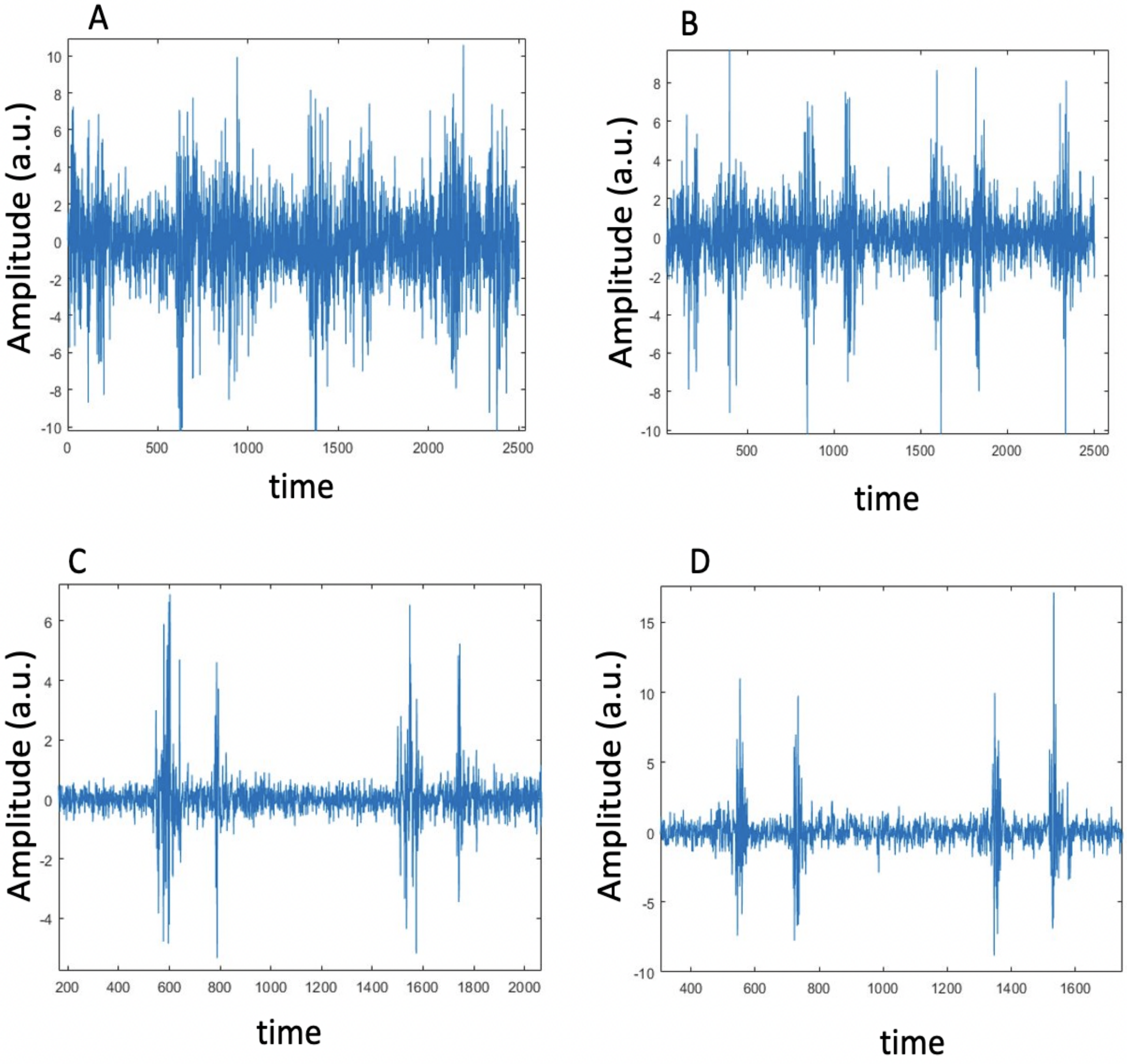


Figure 6.7: A shows subject 1 heart valve sounds acquired while the subject breathed normally. Figure B shows heart valve sounds acquired from the same subject while they weren't breathing. Figure C and D show the same process for a different subject.

chosen. One must choose the proper wavelet family, decomposition level and tresholding method. As we have exposed in Section 3.1.1, the choice of the wavelet family depends on what kind of decomposition is needed. Because we need to perform feature extraction, we need a wavelet family which extracts closely spaced features, and because we also want to denoise the signal, we want to pick an orthogonal family so that energy is conserved. Therefore the best wavelet family to use is the Daubachies.

- Total Variation Denosing:
  This approach aims at preserving the sharp edges of the signal by minimizing a cost function.

In this approach we consider the signal as x(t) perturbed by some noise so that the signal plus noise is $y = x(t) + \lambda$ and the goal is to minimize the function $(x(t) - y(t)^2 dt + \lambda(\dot{x}(t))^2 dt$. By forcing the derivative of the function x to be small, you impose smoothness [167].

- Empirical Mode Decomposition (EMD):
  EMD is a method to decompose data in order to obtain components which help understand features of the data. EMD allows to process non-linear and non-stationary data. EMD does not have a predefined basis like Fourier Transform and fast wavelet transform do, the basis system is dictated by the data.

  This method aims to represent the signal as a superposition of functions called Intrinsic Mode Functions (IMFs). These functions sample the signal at different scales/frequencies. The summation of all modes will return the energy of the original data.
  This IMFs are found by assuming that the characteristics of the signal are found by the time-lapse between the maxima points of the signal. Then an envelope is fitted to the maxima and minima points. Then the mean envelope is found by averaging the previously found envelope functions. Then this function is subtracted from the original data. This is the first IMF. EMD method decomposes a signal into a set of IMFs that are made of single-frequency components. These function that represent the signal at different scale are derived from the signal [168].

6.2.2 Heart Sound Segmentation

Segmentation of heart valve sound is performed either on the raw or denoised signal and it is usually done in order to capture only one full heart beat per segment. This means cutting the signal from the beginning of S1 to the end of S2.

6.2.3 Heart Sound Feature Extraction

We then perform feature extraction on the denoised signal. This is done so that only a small subsample of important features of the signal are passed to the classifier. Classification on features is much more effective than classification on the raw signal because even deep learning models don't know which part of the signal to look at if it is full of irrelevant data (for situations were the training set is much smaller than the number of degrees of freedom) and shallow learning methods such as SVMs would never be able to find the best hyperplane amongst disordered features. A good example of why feature extraction is crucial is shown in Figure 3.6 which shows that it is much easier to classify the signal once the features have been extracted.

Common methods for feature extraction of PCG signal are the wavelet scattering transform and Mel Frequency Cepstrum Coefficient (MFCC).

- Wavelet Scattering transform.
  The wavelet scattering transform has been discussed at length in chapter 3. This method relies on using an operator which is invariant to various translations and additive noise of the signal and extracts invariant features to these mutations of the signal which are intrinsic to the signal.

- Mel Frequency Cepstrum Coefficients.
  In order the acquire the MFCC coefficients the first step is to make an assumption that on short time scales the audio signal doesn't change much. Then we take the power spectrum of these short segments of the signal (the power spectrum is determined using fast Fourier Transform then taking the square of the magnitude component). Then we want to cluster this power spectrum in bins where we sum up all the components in a bin together so to check how much energy exists in various frequency regions. This is done with the Mel filterbank. The first filter is very narrow and gives an indication of how much energy exists near 0 Hz. And then all other bins enclosing higher frequency ranges are checked as well until an estimate of how much energy is present in each bin is obtained. The next step is then to take the logarithm of the energy in each bin. This is done because human hearing works in a non-linear fashion: we don't hear loudness on a linear scale. Then we take the Discrete Cosine Transform (DCT) to decorrelate the energy values of the bins [169].

6.2.4 Feature Selection

Feature selection is the process of reducing the number of variables when there are several input variables. This process can reduce the computational complexity of the problem and yield better results.

There are two main types of feature selection techniques: supervised and unsupervised.

- Unsupervised.
  These methods do not use the target variable to select features.

- Supervised.

These methods take in consideration the target variable.
There are different methods to do this [170, 171]:

- a) Wrapper.
  This method searched for the best performing subsets of features.

- b) Filter.
  This method makes a selection of the features based on their relationship with the target variable.
- c) Embedded.
  These method uses algorithms that perform automatic feature selection during training.

6.2.5 Classification

The goal of classification is to train a model to be able to classify unseen heart valve sound data as either belonging to an healthy or to an unhealthy individual. Classification algorithms that can be used are Naive Bayes, K-nearest neighbor, Support Vector Machine and deep learning methods.

6.3 Devised Proposed Method for Heart Health Assessment

As we first analyzed the data of acquired from our 10 volunteers we noticed that one subject's heart valve sound had different shapes than those of the other subjects. The comparison of this subject's heart sound to the heart sound of another random subject in the group can be seen in Figure 6.8. Compared to the heart sounds of all the other test subjects in our study, which presented an S1 and S2 sounds, as shown in Figure 6.8 A, the test subject whose data is shown in Figure 6.8 B presented also an S4 and S3 peaks before and after all S1 and S2 peaks in all data set obtained from this subject.

After referring to the subject the results of our experiment, the subject went to see a GP and a cardiologist that confirmed that the subject presented with a heart disease, specifically, the subject's heart valve sound also had an abnormal S4 sound.

Given that our group of subjects contained the heart sound of one person which presented a pathology we decided to see how our data-sets of each subject would be labelled (either normal or abnormal) when the training dataset was the one obtained from the 2016 PhysioNet/CinC Challenge. In 2016 PhysioNet/CinC launched a challenge aimed to develop algorithms for the classification of heart valve sounds acquired from different people with the aim to establish if the subject had a heart problem.

The Heart valve sound recordings which composed the 2016 PhysioNet/CinC dataset were sourced from several contributors around the world and collected at either a clinical or nonclinical environment from both healthy and unhealthy subjects. These heart valve sound recordings were collected from different precordial locations. In both training and test sets, heart valve

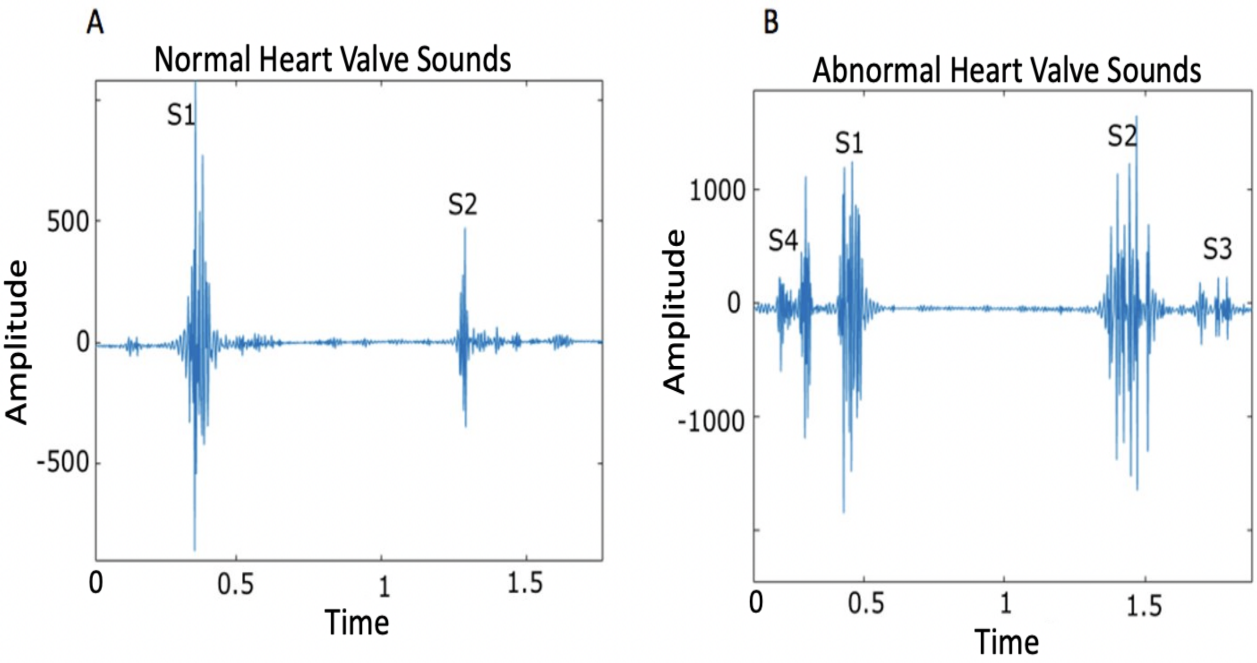


Figure 6.8: A shows the data acquired from one of our subject whose data only presented S1 and S2 heart sounds. Figure 6.8 B shows the data acquired from one of our subjects which also presented S3 and S4 peaks along all the S1 and S2 peaks for the data we acquired.

sound recordings were divided into two classes: normal and abnormal (which depended on whether the heart valve sound recording belonged to a healthy or to an unhealthy subject). The subjects whose heart valve sounds were labelled as unhealthy typically had heart valve defects or coronary artery disease (CAD). The number of normal vs abnormal heart sound recordings was unbalanced given that the number of normal recordings was greater than the number of abnormal recordings. The recordings were acquired both from adults and children and were sampled at 2,000 Hz. The recordings were corrupted by noise. This was due to the fact that these recordings were acquired in uncontrolled environments and noise sources were such as talking, stethoscope motion, breathing and intestinal sounds. These are all important considerations to take into account when building the algorithm.

In the next sections we will show the method that we devised to classify these heart valve sounds in the normal vs abnormal class and then we will show the accuracy of our devised method by testing on a previously unseen subsample of the dataset. Finally, we will how the algorithms classifies the data acquired with our method.

6.3.1 Data Acquisition 2016 PhysioNet/CinC dataset

The 2016 PhysioNet/CinC dataset consists of 3829 recordings, 2575 from healthy subjects and 1254 from unhealthy subjects. Each recording is 10,000 samples long and is sampled at 2 kHz, which corresponds to a segment of 5 seconds. The dataset was found on a GitHub repository and it had to be downloaded. There were two files, one which contained the heart valve sounds and the other which contained the respective class labels.

6.3.2 Post-processing and Feature Extraction for 2016 PhysioNet/CinC dataset

For classification problems it is possible at this stage to segment the signal in order for each dataset to contain only one heart valve sound. However this step wasn't taken as it wasn't necessary, the classification yielded good results with the 5 seconds long datasets.

The next step, since the datset contained more normal sets than abnormal sets was to duplicate some of the datasets so to obtain an equal number of signals in the normal and abnormal classes. This duplication, commonly called oversampling, is one form of data augmentation used in deep learning [173].

The next step consisted in passing the data through a high pass filter which removed all frequency content below 40 Hz and kept the rest. The reason for doing this is that the data acquired with our device contained noise in the frequency range up to 40 Hz, therefore, since the goal was to compare our data to the Physionet data, we needed to make the data sets equal in content. This meant that since we needed to remove low frequencies from the data acquired with our device, we needed to remove this frequency range from Physionet data as well.

The next process applied on the data was feature extraction. Through Matlab a wavelet scattering transform algorithm was applied on the data (structure and functionality of this operator are described in Chapter 3). The hyperparameters of the algorithm were tuned in order to obtain optimal results. It is to be noted that the most important hyperparameters to be tuned are the the number of wavelet transforms (filter banks) to be used. The filter banks used were [16 8 1] which means that the first wavelet filter bank has 16 wavelets per octave, the second 8 and the third 1. For the given scattering parameters, from our input of 3446 testing sets of 10000 samples each, a 375-by-5-by-12540 matrix. There are 375 non-zero scattering paths and five scattering windows for each of the 3446 signals. The actual number of paths for each layer of the scattering three are 138, 4308, and 7000 respectively. However as we have seen in Chapter 2, not all paths are non-zero. The wavelet scattering algorithm subdivides each signal in a determined number of scattering windows and then applies the wavelet scattering transform to each window, that is why from an input of 3446 x 10000 we obtain an output of 375x5x12540.

At this stage the set should be divided in a training a testing set, we partitioned the data 90% of the sets to train the algorithm and 10% to test it.

6.3.3 Classification for 2016 PhysioNet/CinC dataset

In order to pass the exracted features to the SVM classifier (SVM used to perform binary classification -normal, abnormal- on the Physionet data first), we had to reshape our output into a 62700-by-375 matrix where each row represents a single scattering window across the 375 scattering paths. We have just multiplied the number of samples times the number of scattering windows. Now, since we have 5 times more data sets per each sample, we need to replicate the labels.

It is now possible to find a good model to classify our data with. At first a few different models were tested on the training and testing data and the results are shown in Table 6.1

Algorithm	Testing Accuracy
STM	99 %
LSTM	93 %
K nearest neighbours	78 %
Naive Bayes	68 %

Table 6.1: shows the accuracy results obtained on stethoscope test data with different classificaiton algorithms.

Since the SVM and LSTM algorithms were the ones that yielded the best results they were the ones we chose to use going forward.

At first we tried a few algorithms (results shown in Table 6.1) to check which one gave the best results, we tried with an SVM and an LSTM algorithm. The SVM algorithm obtained an accuracy of 99% while the LSTM algorithm obtained 92 %. Furthermore the LSTM algorithm took much longer to train than the SVM. The former took more than 2 hours while the latter just a few minutes so that was the motivation which led to the choice of the SVM.

For the SVM algorithm we used a quadratic polynomial kernel. After fitting the SVM to the training data, we performed a 2-fold cross-validation to estimate the generalization error on the training data. The loss of the algorithm is 0.9 %. Since the loss is very low, we can determine that the model performs well enough. The confusion matrix of predicted versus true labels is shown in Figure 6.9.

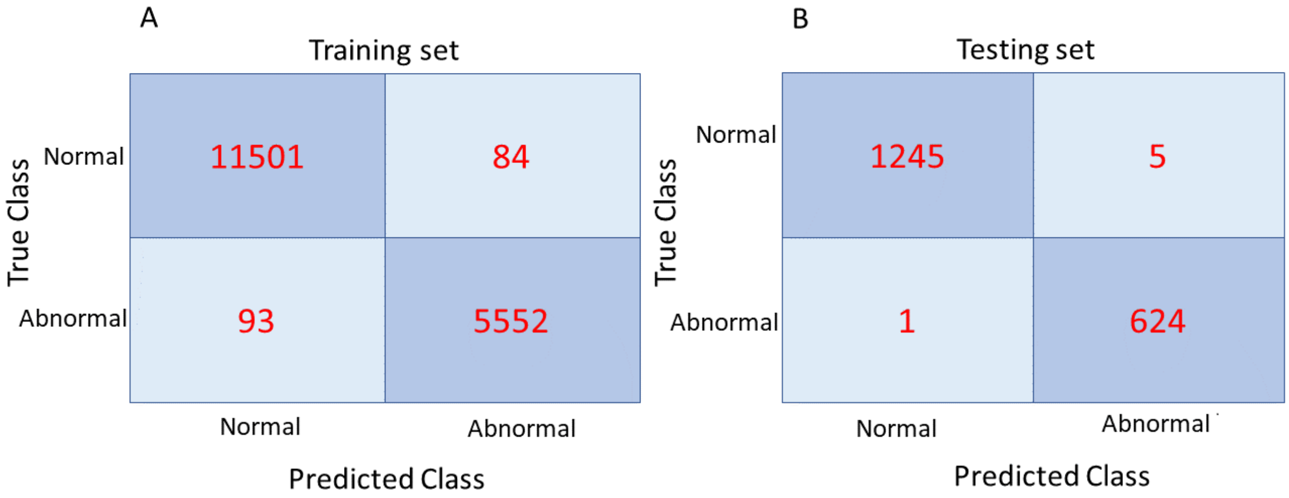


Figure 6.9: A shows the confusion matrix of the predicted vs true class of the classification of the 2016 PhysioNet/CinC training set. Figure 6.9 B shows the confusion matrix of predicted vs true labels of the classification of the 2016 PhysioNet/CinC dataset testing set.

6.3.4 Testing with the Data Set Acquired with our Device

Now that we have tested on the training data, we want to test the accuracy of the algorithm on the data acquired with our device, specifically, we want to test it on the data acquired from our 10 test subjects. This type of classification will mean the algorithm is robust because we are comparing two different types of data. One type is acquired with a digital stethoscope placed on the chest and the other with our device which relies on a laser, a fast camera, post processing and the data is acquired at the base of the neck of the subjects.

If the classification works this would also show that our acquisition method produces data which is very similar to the kind of data that is acquired with the stethoscope on the chest area.

For proof of concept, initially we played recordings of normal and abnormal heart sounds taken from the Thinklabs digital repertoire in an experimental set-up whose layout is shown in Figure 5.9 A. Before passing the data to the normal/abnormal classification algorithm we performed some tasks:

- Rescale the data to -1 to 1 to match the amplitude range of the training data.
- Resample the data, which was acquired at 1500 Hz to 2000 Hz to match the sampling frequency of the training data.
- Segment the signal to 10000 samples, which is the same size of the sets of the training data.
- Pass the data through a high pass filter at 40 Hz.

- pass it through the wavelet scattering transform.

At this point the data was fed into the algorithm to predict the class.

An example of what normal and abnormal recordings look like is shown in Figure 6.10. These recordings were acquired by playing heart valve sounds from a speaker to a vibrating surface and post processing the reflected speckle patterns acquired with a Basler. This was done as proof of concept that the classifier worked properly. The algorithm correctly classified both the normal and the abnormal sets correctly. This gave us confidence that the algorithm was equally well suited to classify data acquired with our device from the neck although it had been trained on data acquired with a digital stethoscope from the chest. This fact is quite remarkable because it meant our algorithm could be used to classify data that was slightly dissimilar to what it was trained with. In machine learning, the goal is to have an algorithm which is trained on data and then can be used on data which is slightly different otherwise it would be useless. The fact that the algorithm could correctly classify heart valve sound data acquired from two different methods was a very good sign that the algorithm worked well.

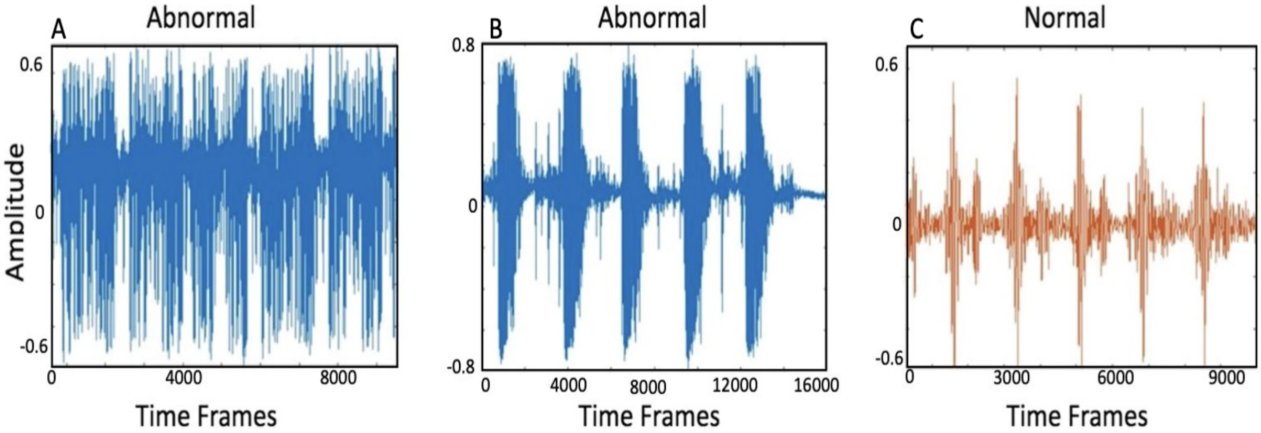


Figure 6.10: shows the amplitude vs time of three data sets acquired with our device with the set-up as shown in Figure 5.9. From the speaker heart valve sounds from healthy and unhealthy patients were played, which were acquired from the ThinkLabs digital repertoire. Figure A and B shows the unhealthy heart valve sounds. Figure C shows healthy heart sounds.

The next step consisted in testing the algorithms on data that we had acquired with our device from subjects' necks. In Table 6.2 it is possible to see the results of the binary classification obtained when we tested the algorithm with the data acquired with our device from the neck of 10 test subjects.

For each subject we tested 10 data sets of 5 seconds each. As it can be seen from the results, for

Subject	Normal	Abnormal
1	20 %	80 %
2	100 %	0 %
3	90 %	10 %
4	80 %	20 %
5	100 %	0 %
6	100 %	0 %
7	90 %	10 %
8	100 %	0 %
9	100 %	0 %
10	100 %	0 %

Table 6.2: shows the results of the binary classification of heart valve sounds obtained with our device from 10 test subjects. Each subject had 10 data sets of 5 seconds each being tested. The total for all subjects were 100 data sets. The table shows how many sets out of the 10 for each subject were classified as normal vs how many as abnormal.

9 subjects the algorithm classified the data sets as being normal. But for 1 subject the algorithm classified the data as being abnormal. The data from this subject has been shown in Figure 6.8 where the abnormal heart valve sound acquired with our device is shown. The test subject went to be tested and it was confirmed that his heart valve sound was abnormal because he presented with an S4 sound. As shown in Figure 6.8 B, it is in fact possible to see the subject's S4 sound, confirming that our device's results matched the medical expert's.

6.4 Conclusion

In this chapter we have shown that we can retrieve heart valve sounds remotely and that our device allows us to acquire the high frequencies ranges of the heart sound from people's neck with a better SNR than the stethoscope can. We have also shown that the high frequencies of the heart sound allow to diagnose heart conditions in a cheap and reliable manner.

We have also shown that machine learning algorithms, aided with feature extraction tools can be used to use this acquired heart valve sound data to make predictions of cardiovascular health.

A tool that can acquire heart valve sounds remotely and use the data to make predictions on people's health can be used, for example, in intensive care units, where the time of action is limited and critical. Such a device could be used for constant non contact and more precise monitoring of the heart condition and then make predictions of when the patient's cardiovascular status is deteriorating through classification algorithms, such as those that have been shown in this chapter and even provide clinicians with relevant information such as how much time they have to make a decision before the situation brings to a fatality with regression algorithms.

Chapter 7

Valve Sounds Authentication with Machine Learning

Person authentication is used in the security domain where one needs to prove their identity. Traditional authentication methods range from inserting a password or showing an ID document, (knowing a piece of information or possessing a token) however these methods are not perfect as a password can be forged or forgotten and an item stolen. Other authentication methods rely on behavioural traits of people such as gait and signature. These methods too are not always accurate as these traits can change over time.

Thus the rise in interest in authentication through biometric traits. Biometric systems are more reliable because they cannot be lost or forgotten, they are much more difficult to forge, share, and distribute. Some examples of biometric traits that can be used for authentication include facial and iris structure [175]. However, such traits can be modified through contact lenses and by wearing make up. Thus the reason for using the heart valve sounds as biometric trait for authentication: it cannot be forged. It has been also proved that heart sounds have highly distinctive characteristics from person to person, therefore it is suitable for human recognition [192].

In this chapter we will discuss common traits used for biometric authentication, compare the performance of these traits against heart valve sounds and then show our results obtained by using heart valve sounds as a biometric trait for authentication. We will demonstrate that by retrieving by including the high frequencies of the heart valve sounds, one can obtain better results compared to using only the low frequency range of the heart valve sound.

7.1 Biometric Authentication Review

Biometric authentication is a sub-field of people identification which deals with identifying individuals based on their physiological parameters. Biometric authentication systems are employed for two different purposes: verification and identification of individuals. In the former case a user's provided identification data is compared to stored data of that person. In the latter, called identification mode, the system validates an individual by comparing their data to that of all other users in the database for a match [177].

The first recorded evidence of biometric authentication comes from 14th century China. Chinese merchants used to acquire fingerprints through ink for identification purposes.

Authentication methods are needed to give access to specific services only to the right individuals who should access those services. Authentication methods are a way to not allow hackers to gain unauthorized access into systems [178]. The use of human characteristics for biometrics provides authentication for different kind of systems. Thanks to the incessant technological advances, nowadays biometric authentication can be divided in several classes [179]:

- Physiological:
  These methods rely on some form or another on the shape of the body. People can be identified through their face, hand, iris or fingerprint structure, etc.
- Behavioral:
  These methods of identification rely on the behavior of a person. People can be identified through their hand writing, dynamics of voice or emotions.
- Cognitive:
  Cognitive biometrics relies on identifying people through the response of the brain to stimuli.

We will now give a brief overview of physiological biometrics.

7.1.1 Fingerprinting

Optical Fingerprints Sensors

The most commonly used fingerprint scanners are optical and solid state sensors. Figure 7.1 shows a schematic of an optical sensor that acquires fingerprints. This device works by frustrated refraction over a glass prism. As it can be seen from Figure 7.1, there is a transparent prism on which a light source is shone through. The configuration of the prism is such that total internal reflection of the light takes place. When a finger is placed or the prism, some of the light is not reflected but absorbed, this is called frustrated total internal reflection [180].

Optical devices for fingerprints acquisition do not work under sunlight, certain types of fingers and consume a lot of power. That is why other methods have been developed. Below we will discuss them.

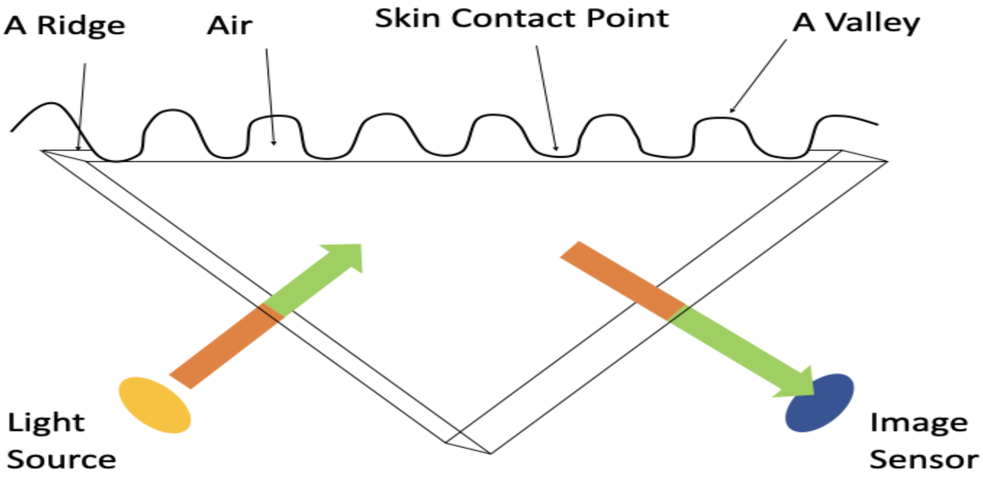


Figure 7.1: shows the set up of an optical fingerprint device. A light source goes through a prism and is totally reflected to fall upon an image sensor which can be a CCD or CMOS. When a finger is placed on the top of the prism some of the light is absorbed due to frustrated total internal reflection and this change in intensity produces the image.

Capacitance Fingerprints Sensors

The human body is filled with conductive electrolytes (minerals that carry an electric charge). The capacitance fingerprints sensor is made up of many parallel plate capacitors which store charge. The capacitance formula is

$$C = \varepsilon_0 \varepsilon_r \frac{A}{d} \tag{7.1}$$

where C is the capacitance, ε_0 is the permittivity of free space, ε_r is the dielectric constant, d is the separation between the electrodes, and A is the area of each electrode [181]. Figure 7.2 shows the schematic layout of a capacitance fingerprint sensor device. The image shows that the valleys and ridges of the finger have different distances from the capacitor sensors, which are placed next to each other. The different distance changes the capacitance. This happens because the finger has a different dialectric constant than the medium between the electrodes. The finger can change the dialectric constant although it is not in the middle of the capacitor's plates because of the evanescent fields emanating from the capacitor.

The advantages of the capacitance fingerprint sensor technologies is that they have low power consumption and work for almost everyone. The drawbacks are vulnerability to strong external electrical fields [181].

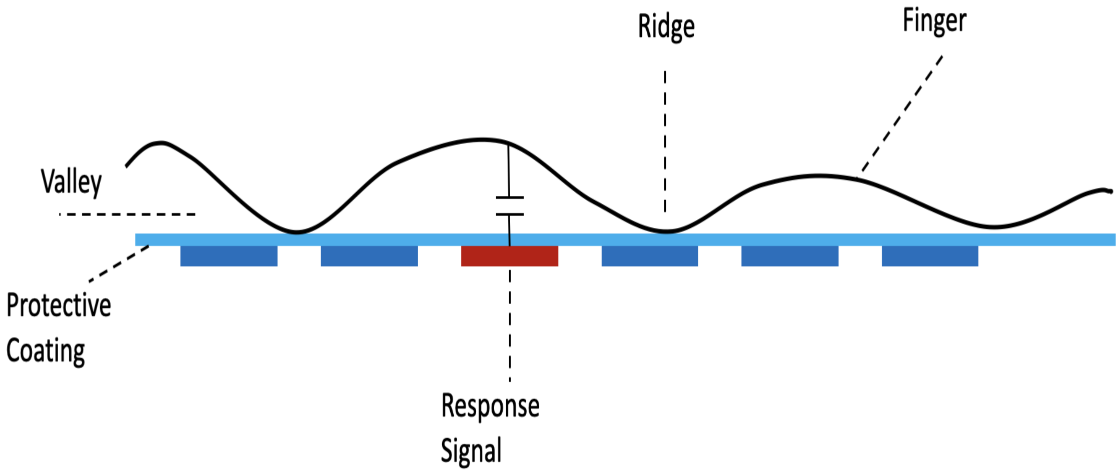


Figure 7.2: shows a capacitance fingerprint sensor. The ridges and valleys of the skin, which touch the small sensors, change the capacitance of the sensors.

7.1.2 Face Recognition

Another biometric trait that can be used for authentication is the shape of the face. Using faces as biometric authentication method yields less precise results than using other biometric traits such as iris or fingerprint. Part of the reason is that faces can be modified with cosmetics, surgeries, disguises, lighting, etc [182].

Face recognition is performed with machine learning algorithms. In 2016 Facebook developed Deepface, an algorithm which can automatically recognize faces almost as well as the human eye can [183]. When using machine learning algorithms, which is the preferred method for authentication systems, the standard procedure is to first detect whether there is a face or not. The following steps consist in feature extraction and classification. These are the steps that Facebook and showed in their paper. They used 4000 different people's faces to train the algorithm and a nine layer deep neural network for classification.

7.1.3 Iris Recognition

This authentication method relies on the iris as biometric trait. The iris is the colored part which surrounds the pupil in the eye. Iris patterns are unique and can be obtained through video. In each iris is embedded a structure featuring a combination of characteristics known as corona, crypts, filaments, freckles, pits, furrows, striations and rings [184]. One of the challenges of using the iris as a means to authenticate people identity is that the iris must first be found inside the image. Popular methods for iris recognition inside an image include taking the derivative of the image to check the location of the edges between the iris and the sclera (white part of the eye) [185]. The next step is to align the acquired iris pattern with other iris patterns in the pre-existing database so to make a comparison of similarity.

7.1.4 Other Physiological Biometric Identification Techniques

Hand Geometry Recognition

These techniques rely on the geometry of hands as a biometric trait. In [186] they proposed a method consisting of the use of an office scanner for the acquisition of digital images followed by post-processing of these images by contouring the outline of the hand and then the use of a multilayer neural network based classifier for the authentication of individuals.

Retina Geometry Recognition

The blood vessels' pattern of the retina has been used as a biometric trait for authentication. The configuration of the vasculature remains unchanged throughout one's lifetime and the pattern is unique for each person. There are some clinical conditions which may produce vasculature changes, however these affect the pattern only at terminal stages of the disease [187]. The benefit of using the retina's vasculature pattern is that since it lies behind the eye it is very difficult to forge compared to other traits which are much easier to forge, such as fingerprints. [188].

Ear Shape

Ear shape is a very reliable biometric trait for authentication because of its stable structure, which varies little with age [190]. In order to segment ears in acquired videos one common technique is called edge orientation pattern matching. This technique relies on using a prepared pattern template to search for the ear in the image. This allows to find the edges and the orientation of the ear [191]. The next steps are those commonly used for biometric authentication techniques in general: feature extraction and classification.

7.1.5 Behavioural Biometric Traits

Now we will discuss behavioural biometric traits.

Voice Recognition

Although the voice is a physiological trait because every person has a different pitch which is dependent on sizes of lungs and oral cavity and nasal cavity etc., voice is classified as a behavioral trait for authentication because these are traits acquired from an action that a person has to take. For voice recognition a person has to speak to a sensor, thus taking an action, and the person can change the tone or cadence of their voice at will [192].

7.1.6 Signature Recognition

The way a person signs their name is a characteristic of the individual and can be used for authentication.

Other behavioural traits include gait and keystroke.

7.1.7 Cognitive Biometric Traits

Brain response to visual stimuli acquired through the ElectroEncephaloGram (EEG) signals have been used as a mean of biometric authentication. The authors of the study [193] claim that this is a very robust method of authentication because the individual's brain brain response to visual stimuli cannot be forged.

7.1.8 Comparison of chosen physiological and behavioural metrics

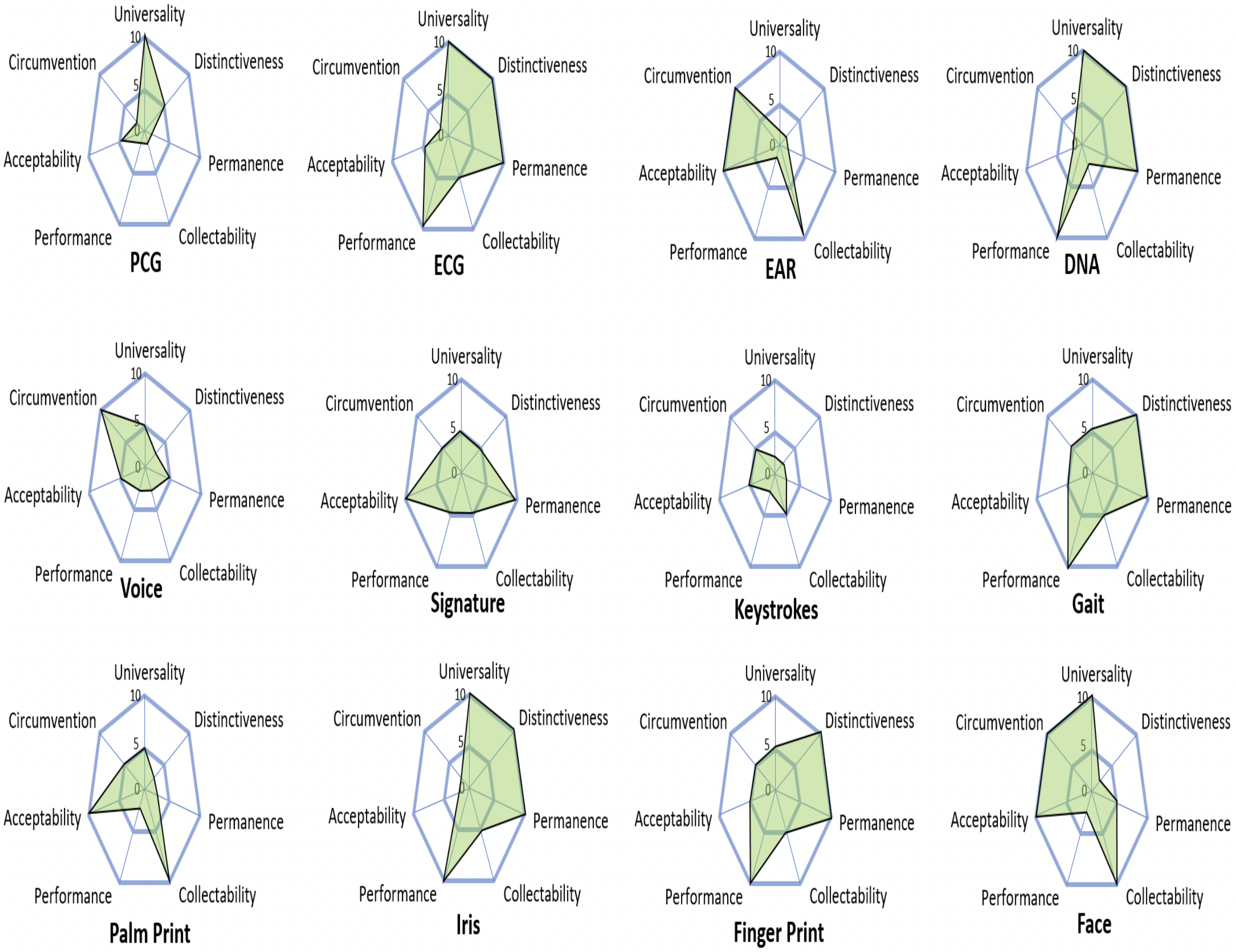


Figure 7.3: Radar plots showing the comparison of how different traits work for the purpose of authentication.

In the field of authentication, in order to assess whether a given trait is suited for the task of authentication, the trait is rated upon 7 metrics. Some traits and their respective scores based on these metrics are shown in Figure 7.3. This is a radar plot where the score of each metric is represented by the colored area of the metric. The seven metrics are universality, uniqueness, permanence, measurability, performance, acceptability, and circumvention. Universality means that each person should possess it. Distinctiveness means that it should aid in the distinction between any two people. Permanence means that it should not change over time. Collectability means that it should be quantitatively measurable. Performance means that the identification should yield efficient results with respect to speed, accuracy and computational requirements. Acceptability refers to the willingness of people to have trait used. Circumvention means the system should be robust to malicious identification attempts [194].

As it can be seen from the graph, ECG, face and iris yield the highest score for a few metrics, making them the best biometric candidates. PCG as a biometric does not fare particularly well compared to other biometrics, but it is deeemed acceptable for these reasons: it can be collected non-invasively, it can only be acquired from a living human body and most importantly, PCG is praised because it is impossible to forge or steal as other authentication methods can.

7.2 Comparison of Low vs High Frequencies of Heart Valve Sound for Biometric Authentication

As we have seen from the previous two chapters, we have shown our device can acquire an higher frequency range of the heart valve sound than what the digital stethoscope can acquire. We have also discussed how, traditionally, biometric authentication has been carried out with heart valve sound data acquired with the stethoscope. However biometric authentication through PCG acquired with stethoscopes, generally has not given good results in terms of performance mostly due to collectability problems. This is because the quality of the sound depends on subjective factors such as how much pressure a clinician applies on the stethoscope and the size of the patients along with other factors which can introduce an error in the acquisition of heart valve sounds.

We will now test heart valve sounds acquired with our device as a method for biometric classification, because not only we can acquire a higher frequency range than the stethoscope can, but our method also gives better results in terms of SNR.

The first test that we carried out was to see which frequency range of the heart valve sound is better suited for biometric authentication since we can retrieve higher frequencies than the stethoscope can, we wanted to test if this was an advantage for people authentication purposes.

In order to test this we needed to filter the heart valve sounds frequencies in low and high frequency ranges to test which range would give the best classification results. The choice of filter for this task was a Butterworth filter. In the next section it will be explained why this is the best choice of filter for this task.

7.2.1 Filter Comparison

There are different linear filters designs which can be chosen according to the requirements of the problem. The elliptic filter is used in applications where a very fast transition between the passband and stopband frequencies is required. This filter in fact provides the fastest transition of any type of filter, but at the cost of having gain ripples in both passband and stopband.

Since in order to separate the high frequencies from the low frequencies we want to use a highpass and lowpass filter, a filter which has ripples in the passband is not good to use because it would give skewed results given that when using for example a low pass elliptic filter the ripples in the high frequency band would still let some (although lower in amplitude) of the high frequencies through thus it would not be possible to make a fair comparison as to whether the people authentication problem is best to be solved with the low vs with the high frequencies of the heart valve sound [195].

Chebyshev filters have a less steep passband and stopband than an elliptic filter which means that they have less ripples. The Butterworth filter has the less steep descent at stopband and passband however it has no ripples. It has a flat transition band [196]. This can be seen from Figure 7.4.

The goal now is to try to classify 10 test subjects from their heart valve sounds acquired with our device. Since our device can acquire frequencies of the heart valve sound which lie in a higher range than the stethoscope can, we want to test if this range of frequencies yields better authentication results. In order to do so we need to properly separate the high from the low frequencies, and the fairest way to do this is to use a Butterworth filter because as we have just seen, this filter does not contain ripples in the stopband.

7.2.2 Data Acquisition

For the experiment of people classification through their heart sounds, we used the data which was acquired for the classification of healthy/unhealthy individuals. So, as described in the previous Chapter, we have acquired data from 10 test subjects, where 4 minutes and 30 seconds of data was acquired from each subject during one session. This data's purpose was to train

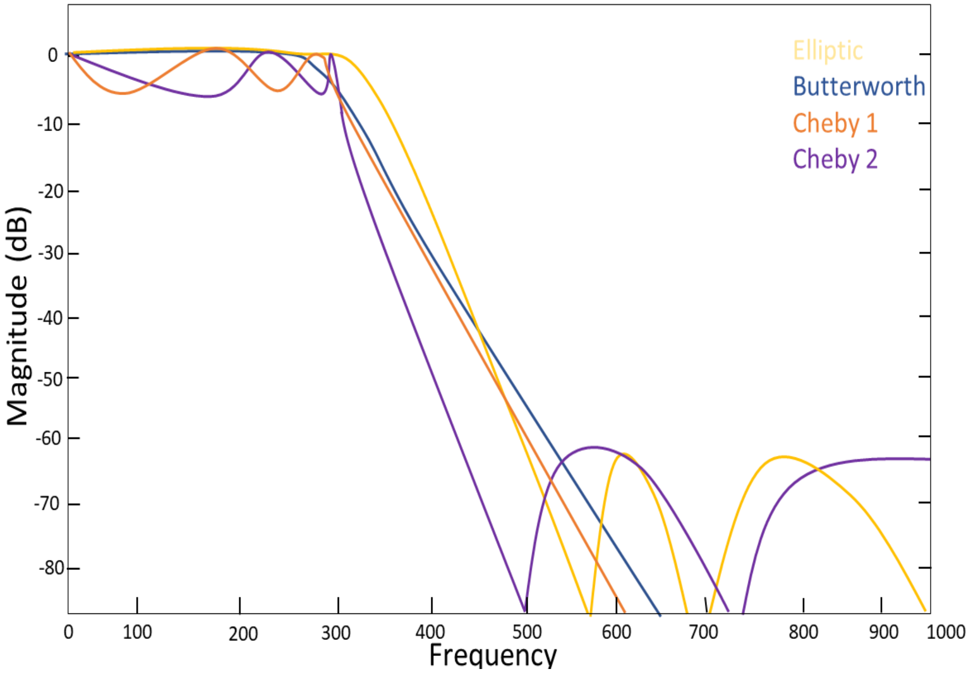


Figure 7.4: shows the shape of various linear filters. As it can be seen, the Butterworth filter has the least rapid descent but then it does not contain ripples in the stopband as compared to all other filters which have a steeper descent but then have ripples (although of very low amplitude) in the stopband.

the algorithm. Then during another session a few days later, 30 seconds of data were acquired from each subject. Once again the laser was pointing towards the neck of the subjects and the experimental set up for data acquisition was the same as the one described in section 6.1.

7.2.3 Post Processing

The data from the 10 test subjects was acquired at a frequency of 1500 Hz. The first post processing step was to apply a level 1 bandpass Butterworth filter to filter our data into different frequency ranges. Because the acquisition frequency was 1500 Hz, the maximum acquired frequency bin which contained frequency information was 749 Hz. We then used the Butterworth bandpass filter to test the classification accuracy which different frequency ranges would yield. One training set contained frequencies in the range 30-740 Hz, another 100-740 Hz, 200-740 Hz, 400-740 Hz and finally 5-250 Hz, the frequency range which is acquired with the stethoscope. This implied we had five training sets, one with the low frequency content of the heart valve sound (5-250 Hz), range that the stethoscope acquires, and 4 others with the high frequency content of the heart sounds, which, to the best of our knowledge, we are the first ones to retrieve.

The frequency range between 5-250 Hz is the range of frequencies which can be acquired with a stethoscope belonging to the sounds the heart valves make as they close. The stethoscope cannot acquire heart valve sounds in higher frequency ranges than about 300 Hz, although it can acquire heart murmurs of higher frequency ranges from the neck or chest, it cannot retrieve heart valve sounds from the neck. Therefore the frequency range containing frequencies between 400-750 Hz is a new range of heart valve sound frequencies which has never been used for biometric authentication as it is the first time, to the best of our knowledge, that it has been retrieved from subjects' necks.

The next steps in post processing consisted of signal normalisation and segmentation. Since we didn't want to keep our test subjects for too long while we acquired data from them, we only acquired 4 minutes of data. Subsequently we divided our 4 minutes of acquisition for each subject into time intervals of 2.5 seconds so that we ended up with 108 training sets for each subject. This was used as training data. The 30 seconds of data acquisitions obtained the following day went through the same pre-processing and we ended up with 12 testing sets for each of the 10 subjects. We used this data as our testing data.

Successively, since the reflected speckle pattern increases in size proportionally to the distance it travels, the displacement of individual speckles caused by the heart mechanical vibrations also increases. Thus the distance between the subject and the camera plays a significant role in the amplitude of the sound retrieved from the optical flow algorithm. Since the subjects were placed at a distance range between 90 and 110 cm from the camera, in order to remove the amplitude bias (which would mean that if a subject was closer than another to the camera, the amplitude of the sound of that first subject would be much lower than the amplitude of the subject farther away from the camera) given by the distance, we rescaled the signal between -1 to 1.

7.2.4 Feature Extraction

In the feature extraction step we used a wavelet scattering network (working principles of this are described in Chapter 3) to extract features of our audio signals which are stable to variabilities caused by time-warping deformations. We did this because in classification problems the goal is to group similar signals in the same class, but the problem is that real life signals, even when belonging to the same class, have dissimilarities between each other, thus making the signals impossible to classify through a simple Euclidean norm. In contrast, by extracting stable features it is possible to disregard signal deformations and only obtain coefficients which are intrinsically representative of the signal even when the signal is deformed by additive noise, translations, dilations, rotations, etc (as properly explained throughout Chapter 3). At the end, the extracted coefficients will allow to group signals belonging to the same class closer together [74].

Furthermore, although as described in Chapter 3, a CNN's first layers are the equivalent of the wavelet scattering transform, the dataset was too small for a CNN to appropriately choose the correct filters. This concept is best illustrated in Figure 7.5. The architecture of our wavelet scattering transform, which resembles the physiological processing method used by the cochlea (thus proving its efficiency in dealing with audio signals) uses three filterbanks. The first one contains 8 wavelets per octave. The second one contains 4 wavelets per octave and the third one contains 1. The number of total nodes at each layer are then shown in the architecture structure next to each layer. Figure 7.5 shows the structure of the scattering transform we used. This figure is also included in my published work [197].

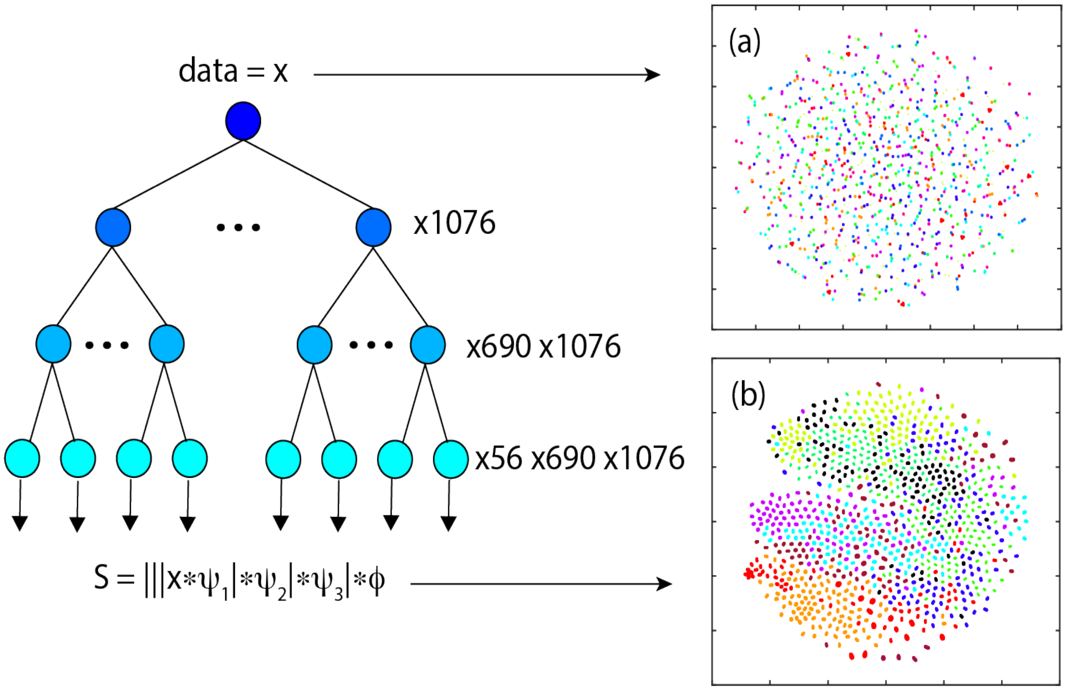


Figure 7.5: shows on the left the architecture of the scattering transform. The signal is convolved with 1076 filters in the first layer, 690 in the second and 56 in the third. We train the SVM with the outputs of the scattering transform of the third layer. Figure 7.5 a shows the heart sound data from the different data sets of the 10 test subjects. As it can be seen from the t-SNE, the sets belonging to different individuals are not grouped together. After the data goes through the scattering transform the data sets within the same class are clustered together (Figure 7.5 b). This figure has been included in my published work [197].

We then used the outputs of the third layer to train our SVM model. In Figure 7.5 (a) it is shown the raw data represented through a t-SNE. As it can be seen, that there is no real distinction between the classes (each class represents data from a different subject). But the coefficients obtained after the signal goes through the network are very well grouped into classes, as shown in Figure 7.5 (b). By looking at Figure 7.5 (a) and (b), it is possible to see that there are more data points in Figure 7.5 (b) than in (a). This is because for each initial input signal, there will be 4 new signals containing invariant coefficients.

7.2.5 Classification

Once we extracted features with the wavelet scattering transform which grouped signals within the same classes close together, we were able to obtain high classification accuracy by using an SVM. The hyperparameters used were a 3th degree polynomial kernel, which was found to give optimal results. The model was trained using 2-fold cross validation. 3-fold up to 10-fold cross validation was also tested, but the accuracy improved only by 3% total and it took from 3 to 30 minutes longer to train the algorithm, therefore a 2-fold cross validation was chosen.

The code that has been used to filter the signal, perform feature extraction and classification in included in Appendix E.

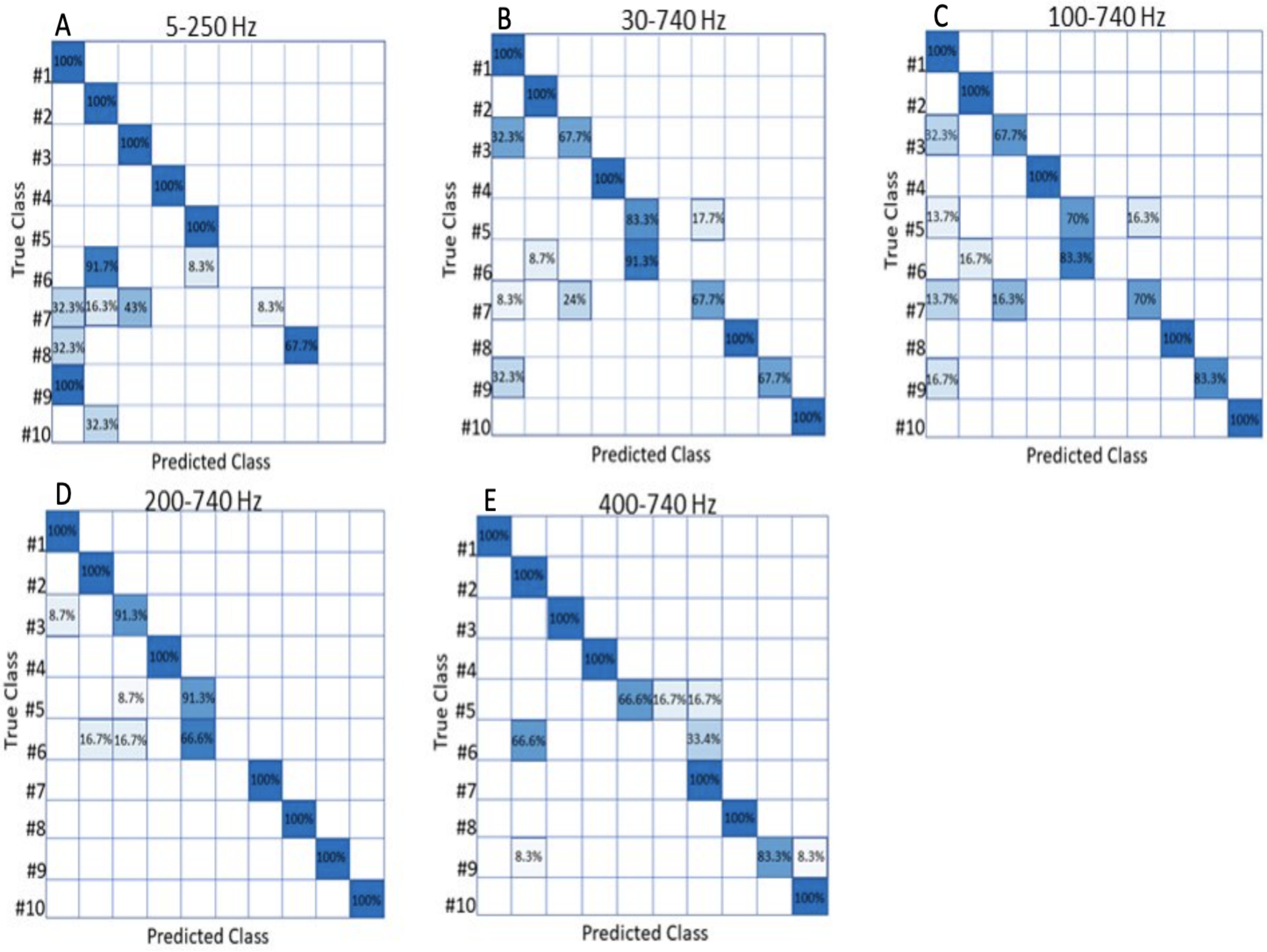
7.2.6 Results

To test this model, we used the testing data, whose acquisition and pre-processing has been described in the previous sections. It is important to note, that since, as described in the feature extraction section, we obtain 4 outputs signals of coefficients for each input, in order to get a true classification output, we take the mode of the 4 outputs labels for each signal. To say this in other words, since from the last layer of the scattering network we obtain 4 sets of coefficients for each input data set, in order to classify an original now that 4 sets of coefficients have been extracted from it, the set will be classified as belonging to a particular person if 3 sets out of the 4 have been classified to that person.

Since our goal in this section was to check whether the low or high frequency range of the heart sound yielded the best results, we trained the same algorithm and performed the same pre-processing on the data as described in the previous sections. The only change was the frequency range that the Butterworth filter let through.

As mentioned, we trained the SVM with data which has been filtered at different frequency bands. The classification accuracy of the frequency band that the stethoscope can acquire is shown in Figure 7.6 5-250 Hz. This frequency range yields an accuracy of 59%. The confusion matrix shown shows the results of the test sets of the 10 subjects labelled 1-10. At the intersection between the predicted class and true class lies the number of sets (out of 12 testing sets per each subject) that the algorithm classified correctly. The testing sets for which the predicted class does not match the true class, thus those which have been incorrectly classified, lie outside the diagonal.

The frequency range 30-740 Hz yields 48% accuracy. The frequency range between 100-740 Hz gives an accuracy of 77%. The frequency range between 200-740 Hz yields an accuracy of



A 5-250 Hz

#1	100%									
#2		100%								
#3			100%							
#4				100%						
#5					100%					
#6		91.7%			8.3%					
#7	32.3%	16.3%	43%				8.3%			
#8	32.3%							67.7%		
#9	100%									
#10		32.3%								

B 30-740 Hz

#1	100%									
#2		100%								
#3	32.3%		67.7%							
#4				100%						
#5					83.3%		17.7%			
#6		8.7%			91.3%					
#7	8.3%		24%				67.7%			
#8								100%		
#9	32.3%								67.7%	
#10										100%

C 100-740 Hz

#1	100%									
#2		100%								
#3	32.3%		67.7%							
#4				100%						
#5	13.7%				70%		16.3%			
#6		16.7%			83.3%					
#7	13.7%		16.3%				70%			
#8								100%		
#9	16.7%								83.3%	
#10										100%

D 200-740 Hz

#1	100%									
#2		100%								
#3	8.7%		91.3%							
#4				100%						
#5			8.7%		91.3%					
#6		16.7%	16.7%		66.6%					
#7							100%			
#8								100%		
#9									100%	
#10										100%

E 400-740 Hz

#1	100%									
#2		100%								
#3			100%							
#4				100%						
#5					66.6%	16.7%	16.7%			
#6		66.6%					33.4%			
#7							100%			
#8								100%		
#9		8.3%							83.3%	8.3%
#10										100%

Figure 7.6: shows the confusion matrix results obtained when the SVM was trained with the frequency range of the heart valve sounds which can be acquired with a stethoscope above the chest (5-250 Hz). The numbers on the diagonal represent the percentage of sets that were correctly classified while off diagonal the ones that were misclassified. The legend next to each column and row, P_1 to $P_{1}0$ is an abbreviation to signify results from person 1 to person 10 and for each row there are the results of the specified subject. The other confusion matrices show the results when different frequency ranges are used. As it can be seen, the classification done with the frequency range that the stethoscope can acquire is compared with the frequency ranges 30-740, 100-740, 200-740,400-740 Hz. The frequency range which correctly classifies the most sets is the one between 200-740 Hz.

89%. The frequency range between 400-740 Hz gives an accuracy of 84%.

It is important to reiterate that, to the best of our knowledge, this was the first time that classification of heart valve sounds was carried out by using this kind of high frequency range of the heart valve sounds to train a machine learning algorithm.

The confusion matrices have entries where there is a correctly or incorrectly predicted value for a given person. The values range between 0 and 12 on the rows because 12 is the number of total testing sets available from each subject. The accuracy results show that the best frequency range to perform authentication with is the one containing frequencies between 200-740 Hz. This results show that the high frequencies should be used in people classification because they allow to obtain better accuracy results.

Although these results are good, we wanted to increase the accuracy of people identification through heart valve sound through better feature extraction and post processing methods.

In the next sections we will show that we can improve and show the techniques which allow to do so.

7.3 Improving the accuracy of biometric authentication through wavelet filters

Once again the steps taken to perform biometric authentication were: database acquisition, post processing, feature extraction and finally classification. In the next sections we will go into depth into each one of these stages. The main difference compared to the results shown in the previous section will be to use wavelet filters in the post processing step to improve the classification accuracy.

7.3.1 Database Acquisition

The data used for this task was the same used in the previous section. The data was acquired from 10 test subjects where 4 minutes and 30 seconds of data were acquired from each subject during one session. This data's purpose was to train the algorithm. Then during another session a few days later 30 seconds of data were acquired from each subject, this data was used for testing. Once again the laser was pointing towards the neck of the subjects and the experimental set up for data acquisition was the same as the one described in section 6.1.

7.3.2 Post Processing

After acquiring the signal, the first step was to use an optical flow algorithm so to obtain the heart valve sound from the videos of the reflected speckle patterns. In order to achieve this we used the optical flow algorithm. We used an objective in front of the camera to retrieve as much light as possible with the least amount of laser power so as to not irritate the skin of the test subjects but also so that it would be eye safe. The laser power was 0.4 mW.

The objective also allowed us to capture many speckles, which made the optical algorithm the

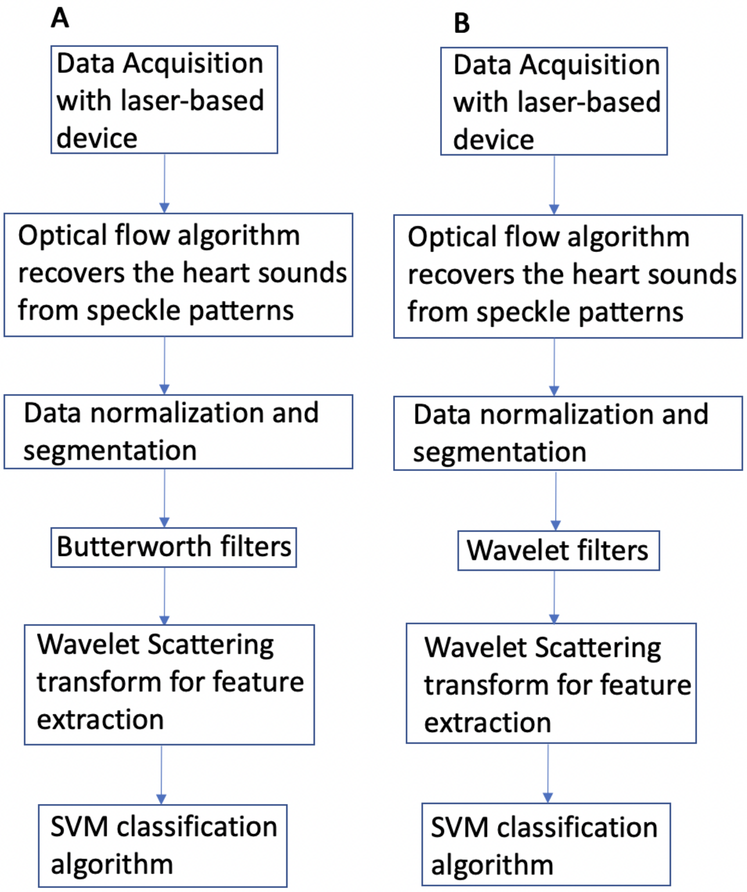

A
Data Acquisition with laser-based device
Optical flow algorithm recovers the heart sounds from speckle patterns
Data normalization and segmentation
Butterworth filters
Wavelet Scattering transform for feature extraction
SVM classification algorithm
B
Data Acquisition with laser-based device
Optical flow algorithm recovers the heart sounds from speckle patterns
Data normalization and segmentation
Wavelet filters
Wavelet Scattering transform for feature extraction
SVM classification algorithm

Figure 7.7: A is a diagram of the steps of the algorithm for biometric authentication described in the previous sections. One of the steps requires filtering with a bitterworth filter. Figure 7.7 B is the same as 7.7 A but now instead of using a Butterworth filter, wavelets filters are used.

best post-processing method, as compared to when only one speckle falls on the camera, where the best post processing method consists in simple frame integration.

The next few steps in post processing are as follows:

- Mallat Algorithm (Fast Wavelet Transform).
  The signal was denoised using wavelets. It has been shown wavelets are well suited to denoise PCG signal [198].

  Compared to the short time Fourier Transform, which returns the time-frequency content of a signal with a constant frequency and time resolution due to the fixed window length, wavelets allow for a multi-resolution analysis because they can be scaled and dilated (see full explanation in Chapter 3).

  The first level detail coefficients are retrieved using Mallat's Algorithm, also known as the fast wavelet transform, which in the first stage, divides the signal into high frequencies and low frequencies from f to $\frac{f}{2}$ and from $\frac{f}{2}$ to 0. In the next stage it will divide the low frequencies into high frequencies and low frequencies again and so on. This concept has been explained in Chapter 3 in depth and a representation of what this signal decomposition looks like is shown in Figure 7.8.

The first level D1, shows the high frequencies of the heart valve sounds (amplitude vs time), from 750 to 375 Hz (f to $\frac{f}{2}$). As it can be seen from Figure 7.8, at this level the signal is very well located in time.

In the second level, D2, there are the frequencies ($\frac{f}{2}$ to $\frac{f}{4}$) from 375 Hz to 187 Hz. In this band the frequencies are not as well localised in time as they were in D1, it can in fact be seen that they span a larger part of time.

The other levels follow the same logic, with D3 being $\frac{f}{4}$ to $\frac{f}{8}$ etc and the heart valve sounds gets less and less localised in time.

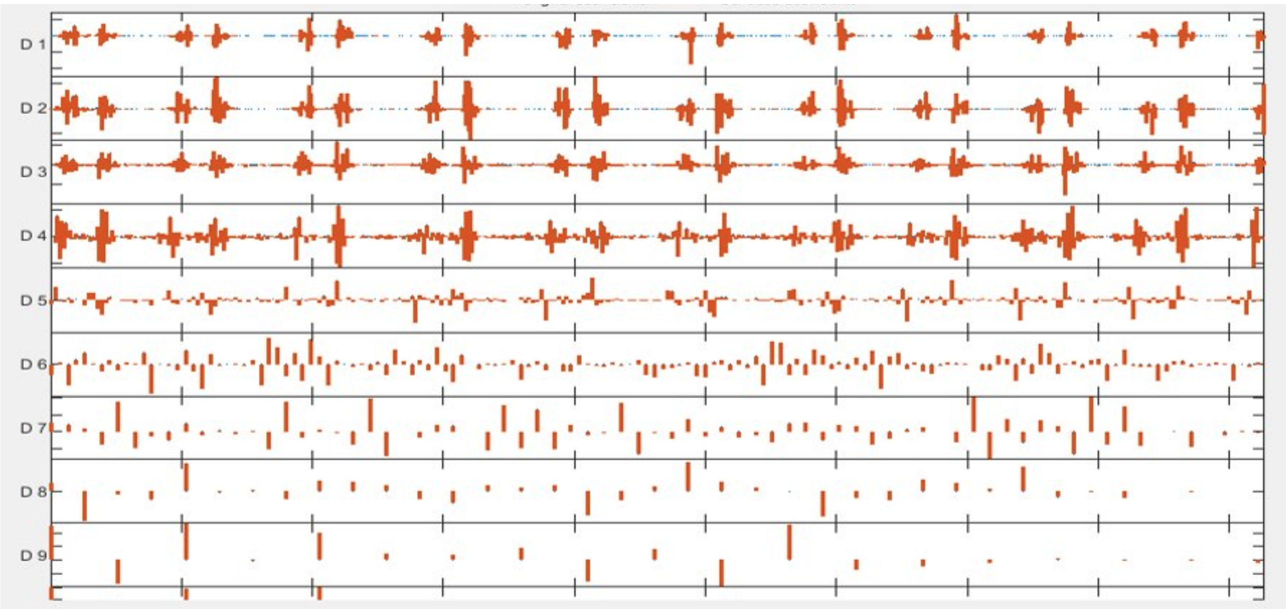


Figure 7.8: Decomposition of the signal with the Mallat algorithm (fast wavelet transform). The signal is decomposed in various levels with wavelet filters. The first level contains the frequencies from $\frac{f}{t}o\frac{f}{2}$, the second level contains the frequencies of the signal from $\frac{f}{2}$ to $\frac{f}{4}$ and so on for all other levels.

The number of decomposition levels obtained through this passage can be chosen as required. Choosing the decomposition level which contains the most signal and the best SNR depends on the sensor acquisition frequency and the parameters of the experiment.

We will perform biometric authentication using the signal decomposed in various levels of the fast wavelet transform. We will provide the results of the classification algorithm when this is trained with the first decomposition level versus when we train it with the second and third decomposition level.

- Choice of Wavelet.
  Because the continuous Wavelet Transfrom is too computationally intensive, we used the discrete wavelet transform (DWT) to extract the first level detail coefficients. In order to denoise the signal we used a Daubechies 2 wavelet because for feature extraction we want a wavelet with low vanishing moments and we want to choose an orthogonal wavelet (see Chapter 3 for more details). A Daubechies wavelet is also similar to the filter used by the cochlea in our ears, this shows it's a type of filter well suited for sound signals. The more a wavelet resembles the signal, the better it can denoise it. This is because the wavelet is convolved with a section of the signal and the degree of correlation between the wavelet and signal section is calculated. Then the wavelet is shifted and the process repeated. Then the wavelet is scaled, placed back at the beginning of the signal and the a process repeated.

- Tresholding Method.
  Wavelets are not coherent compared to the noise, thus the convolution with noise will result is small coefficients while the convolution with the signal will result in big coefficients. After retrieving these coefficients, the low ones, attributed to noise, can be removed. A suitable threshold can separate the noise, low coefficients, from the signal, high coefficients. Then the signal is reconstructed through inverse reconstruction (IDWT).

  There are different thresholding techniques, such as hard or soft thresholding. We found no significant difference in the thresholding method used.

- Amplitude rescaling and signal Segmentation.
  After a coherent laser light hits a scattering surface, the resulting speckle pattern diverges as it spreads. This means that the individual speckles increase in size with distance and the distance between speckles also increases. Accordingly, the later displacement of individual speckles caused by the heart mechanical vibrations also increases, thus the distance between the subject and the camera plays a significant role in the amplitude of the sound. Since the subjects were placed at a distance range between 90 and 110 cm from the camera, in order to remove the amplitude bias given by the distance, we rescaled the signal between -1 to 1. Then we divided our 4 minutes of acquisition for each subject into time intervals of 2.5 seconds therefore we ended up with 108 training sets for each subject. The 30 seconds of data acquisitions obtained the following day went through the same pre-processing and we ended up with 12 testing sets for each of the 10 subjects. The signal is finally sub-sampled by 2 so as to reduce the amount of data given to the classifi-

cation algorithm.

7.3.3 Feature Extraction

In order to reduce the dimensionality of the problem and extract invariant features we used the wavelet scattering transform again.

7.3.4 classification

We used an SVM for classification. The hyperparameters used were a 3th degree polynomial kernel, which was found to give optimal results. The model was trained using 2-fold cross validation.

7.3.5 Results

As it can be seen from Figure, the best results are achieved when the first level detail coefficients are used to train the algorithm. These are frequencies ranging from 375-750 Hz. The accuracy decreases when the second level detail coefficients are used to train the algorithm, these frequencies go from 187 to 375 Hz. The third level detail coefficients yield the worst results. This frequency band contains frequencies between 93 Hz to 187 Hz.

As it can be seen, the results obtained when the algorithm is trained with the first level detail coefficients of the wavelet transform, the accuracy is better than when a simple Butterworth filter was used.

It would be interesting to test what the accuracy would be if the first and second level detail coefficients were to be summed so to span a frequency range from 187 Hz to 750 Hz which would make a good comparison to the frequency range tested with the Butterworth filter (200-750 Hz) which yielded the best results with that filter. This could be a continuation of this work.

7.4 Conclusion

Since our method acquires the high frequencies of the heart valve sound from the neck with a better SNR than the stethoscope can we have tested which frequency range gives the best people classification accuracy. In order to do this we used a Butterworth filter, which is appropriate for the task because it doesn't have ripples in the stopband, thus undesired frequencies are completely removed.

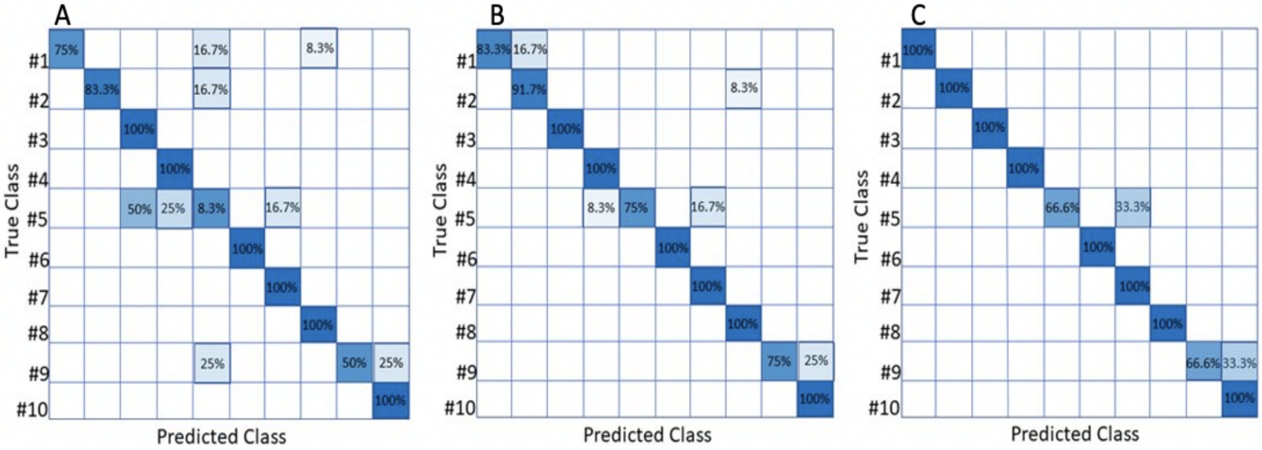


Figure 7.9: shows the confusion matrices obtained by using respectively CD1 (750-375 Hz, Daubechies 2 wavelet filters), CD2 (375-187 Hz) and CD3 (187-953 Hz).

We found out that the best frequency range to perform authentication is the range from 200-750 Hz.

We have then tried to improve the accuracy by using wavelet filters at the post processing stage to clean the signal. We decomposed the signal into separate levels and tested which level gave the best accuracy. The first level detail coefficients (375-749 Hz) gave the best classification accuracy, which was better than the accuracy obtained with the Butterworth filter.

The overall conclusion to be drawn from these results is that our devised laser method for heart sound acquisition gives better SNR than the stethoscope can, especially at frequencies above 300 Hz.

Chapter 8

Conclusion and Future Directions

In this thesis I have presented a non-contactless device which can acquire heart valve sounds remotely and a complimentary software which, with computational methods, feature extraction, analytical methods and machine learning can be trained on the acquired heart valve sound data to asses cardiovascular health and perform biometric authentication.

Part of the inspiration for this thesis has been the work of Dr. Eric Topol, who has pioneered individualised medicine, which has been described by the National Academy of Sciences as âĂIJtailoring of medical treatment to the individual characteristics of each patient". There is a digital transformation taking place in the NHS and in healthcare in general which aims to constantly collect, as Dr. Topol sustains, " a panoramic of biologic data and relevant medical information" from the individual in order to provide the most targeted treatment possible [199]. Fundamental to individualized medicine is analysing an immense amount of diverse biological data sets and extract only the relevant information to make diagnosis. This is the job of the software. The software I devised uses the work of the French mathematician Stéphane Georges Mallat to extract only the relevant bits of information from a large amount of heart sound data. Then machine learning algorithms are used to make decisions on the patient's health and to perform biometric authentication.

The work of this thesis aims to take a small step forward towards the implementation of individualised medicine with remote monitoring for data collection and diagnosis through artificial intelligence.

The second chapter lays down the information about the anatomy of the heart, the mechanical processes which produces the heart sounds, the illnesses that can affect it and what mechanical changes these illnesses would produce. Each illness induces a different sound, and it is upon this premise that the remote sensing device and complimentary software were devised.

The third chapter goes into depth about describing the wavelet scattering transform, an algorithm developed by Dr. Stéphane Mallat, which has been used in the software to extract the invariant features of the heart sounds and reduced the dimensionality of the problem. This procedure allowed the machine learning algorithm to perform classification with high accuracy despite the dataset being small compared to the number of degrees of freedom. Without this algorithm the ML classifier obtained only 20 % accuracy, while the accuracy reached over 99% once the wavelet scattering transform was applied to the data before this was fed to the ML classifier. This implies that for small datasets with a high number of degrees of freedom, computational methods can be used to manipulate the data before feeding it to ML algorithms in order to increase the accuracy of the classifier. It is also of interest to note that the wavelet scattering transform served a dual role in this thesis. For the classification problem of distinguishing healthy from unhealthy individuals, the wavelet scattering transform served to pick out the most important features of the signal, while for the problem of biometric authentication the wavelet scattering transform most important purpose was to reduce the dimensionality of the problem. The problem of successfully building a ML classifier which obtains high accuracy with small datasets is of utter importance in my current field of work in the NHS where, despite the immense amount of data collected, there are still many reasons why in many instances there is not enough data to use to train ML models. Instances of this can be clinical trials with new techniques are tested on a small cohort of patients or a hospital cures a diseases which is rare in occurrence and datasets need to be individually labelled by the clinicians. When gathering more data is not a viable option, the ML is likely to incur is over-fitting the test data and thus not being a reliable predictor of unseen data. In this thesis some methods to work with small datasets have been used: from feature engineering to feature extraction to using models which perform better with small datasets.

The fourth chapter provides a brief introduction to machine learning. It delivers an overview of the principles, algorithms, and applications of machine learning from the point of view of modeling and prediction. It includes concepts such as classification, linear regression and support vector machines by supplying the basic ideas and intuition behind modern machine learning methods and laying out the basics as to how, why, and when these models work. Most importantly, this chapter provides the physical interpretation of the Support Vector Machine model, an algorithm which works well with small datasets, which has been used to perform both assessment of healthy vs unhealthy individuals and for biometric authenticaiton.

The fifth chapter has given an introduction to Laser Doppler Vibrometry techniques, describing how sound data has been recovered from light. Our main personal contribution to this topic is the relationship between what post processing technique to use depending on the distance between the camera and the reflecting surface. Our research delves into different post processing techniques that can be used on the reflected light data to extract the sound. In this chapter we showed that by simply summing all the pixels in every frame of the collected video of the scattered light it is possible to recover sound with a high SNR if the camera is close to the reflecting surface, while if the camera is farther away from the reflective surface, a machine learning tracking algorithm is preferable. This concept can also be summarized in a different way: if the speckle size is similar or larger than the sensor size, then integration will yield the best SNR, while if the speckle size is much smaller than the sensor size, then a tracking algorithm will produce a better SNR. These findings have served to set up the experiments described in the sixth and seventh chapters. In fact, since the subjects were placed at one meter from the camera, and we used very low laser power to avoid possible damage to eyes and skin, an objective to collect light had to be used. The objective in front of the camera brings us in the regime where the speckles' size is much smaller than the detector size. Our findings then directed us to choose to post process the data with a tracking algorithm.

The sixth chapter describes how we used our laser-based device to collect data from our cohort of tests subjects, the nature of the data, how we post processed it and finally the machine learning algorithm used to classify the data into an healthy and unhealthy category. A major problem for small datasets in overfitting. In this chapter we describe the implementation of the feature extraction algorithm presented in chapter 3. The wavelet scattering transform is used on the data to extract the stable and most representative features of the signal, thus vastly reducing the dimensionality of the problem and allowing to avoid overfitting. An SVM classifier is then applied to the data to make the healthy vs unhealthy prediction. What is worth noting from our results is that we first trained the Machine Learning model on heart sound data acquired with a digital stethoscope applied on the chest of the patients. The model was then tested on a previously unseen subset of this same data, providing accuracy results above 99%. Then the algorithm was also tested on the data acquired with our laser-based device from the neck of the subjects. Surprisingly, the algorithm could still recognize healthy from unhealthy individuals. Furthermore the SNR of the data acquired with our device is higher than the SNR obtained with the digital stethoscope.

In the final chapter we use the data collected with our device to perform biometric authentication. We also compare which frequency ranges of the heart sounds yield the best accuracy.

Now I work in the Clinical Scientific Computing team at Guy's and St Thomas' hospital, which is dedicated to bring the NHS into this digital transformation era to develop people, policies and platforms for digital health. With the support and permission of my new employer I have applied to the NHS Topol Fellowship and the Wellcome/EPSRC Centre for Medical Engineering funding. The goal is to test the limitations and potential of this device, which allows the detection of cardiac dysfunction long before the heart begins to fail. Therefore there could be opportunities to transform care (delivered by clinicians or enabling remote patient self-management) by:-

- Screening people with risk factors (eg: hypertension or diabetes) to detect cardiac dysfunction.
- Monitoring people with CV disease for early detection and prevention of heart failure.
- Guiding management of congestion (key to wellbeing prognosis) in patients with heart failure.

Thanks to the connections of the Clinical Scientific Computing team it will be possible to implement collaborations between different cardiology clinics to:-

- Further develop laser diagnostic technology (LDT), to create a miniaturised, portable device that can be easily deployed (10x10 cm2 box containing a 0.5 mW eye-safe laser and a camera).
- Determine the feasibility of identifying, or excluding, cardiac dysfunction and congestion, by LDT compared to echocardiography and cardiac biomarkers, in a cohort of patients with a broad range of cardiovascular risk factors (eg: diabetes and hypertension) and established cardiovascular disease (eg: prior myocardial infarction, heart failure and valve disease).
- Verify the diagnostic between-visit reproducibility, sensitivity and specificity of LDT.
- Use machine learning and neural networks trained on existing stethoscope data, to verify the robustness of heart sound detection and classification.
- Identify other potential areas for clinical research using LDT, for instance arterial rigidity and arterial flow turbulence.

Once enough data will be collected it will be easy to implement the software thanks to platform such as AIDE and XNAT which will, respectively, facilitate deployment of AI software and allow the easy anonimization of patient data. If the results of the clinical trials will be promising, the software and complimentary device might be implemented in the NHS and eventually for home remote monitoring.

Appendix A

Interpretability Study on the LSTM Algorithm Used in Chapter 6 to Classify Healthy Vs Unhealthy Heart Valve Sounds

Our collaborator, Ms Yola Jones, explored the interpretability of the LSTM algorithm which was used to predict healthy vs unhealthy heart valve sounds acquired with a digital stethoscope. As explained in Chapter 6, in the feature extraction step, from the PCG sound some features which are invariant to various translations are obtained. These relevant features make it so that from a signal of around 8000 samples we are left with an order of magnitude of features of 350 features per sample. However, Ms Yola Jones discovered that the LSTM was not looking at all these 350 features per signal, rather it was focusing on the first few and last few features of each signal. In fact she discovered that when cutting out the middle features for each sample in the test set (basically setting the middle values to 0 to) this is what happened to the model's accuracy:

- Cutting out the middle 60 % reduced model accuracy by less than 1 %

  Cutting out the middle 80 % reduced model accuracy by less than 5 %

- Cutting out the middle 85 % reduced model accuracy by less than 5.5 %Ãź

Now, to illustrate this concept, one can look at figure 1 which shows the variance of the features of the training set for normal and abnormal subjects and the variance of the features in the testing set for normal and abnormal subjects. In the graph it is also plotted the importance given to the features by the LSTM algorithm. From this graph the fact the algorithm gives more importance to the first few and last few features makes sense because these are the points where there is more variance between normal and abnormal features sets, and since there is the most variability, then the algorithm can pick these points to make predictions.
This test proved that the algorithm is making choices by looking at the right features.
This is an important test which could be shown to clinicians to prove that the algorithm, although

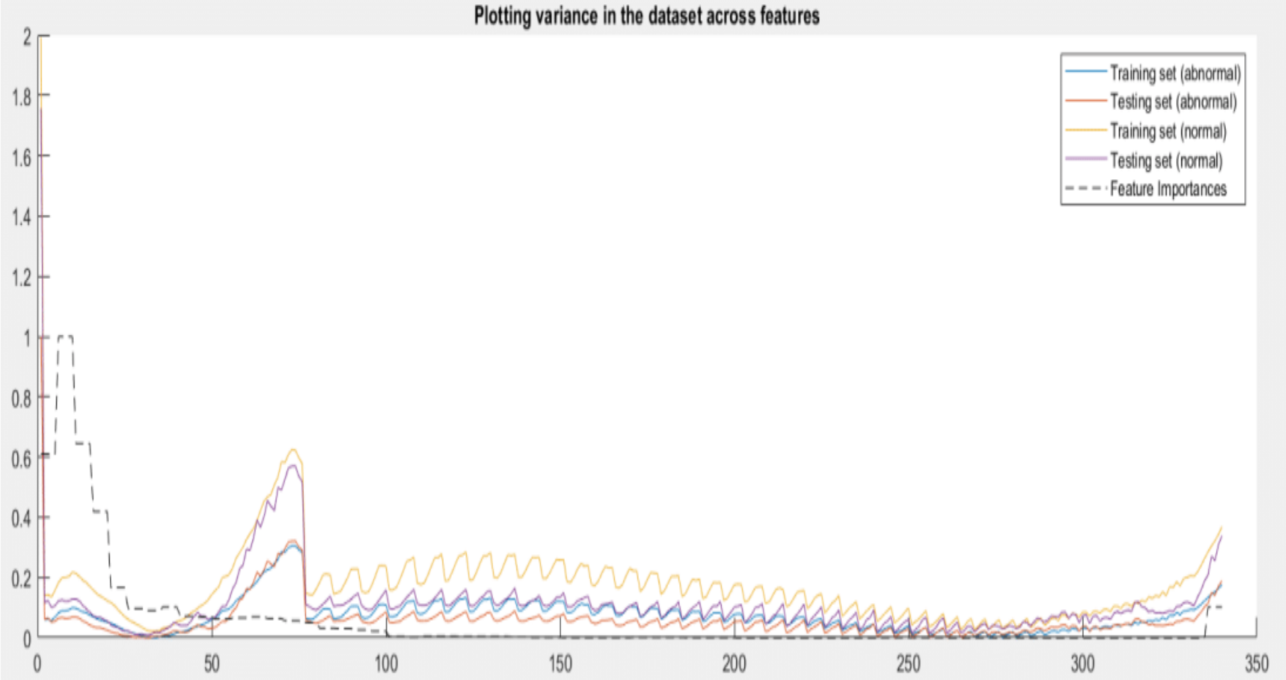


Figure 1: shows the variance of the features of the testing and training sets of normal and abnormal heart valve sounds acquired with a digital stethoscope. In the graph it is also shown the feature importance given by the LSTM algorithm to the various features' points.

it is a black box, is making choices based on decisions which we can support and understand thus giving more credibility to the outputs provided by the algorithm.

Appendix B

List of possible maladies which can affect the heart

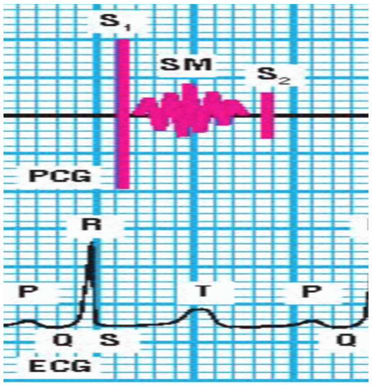
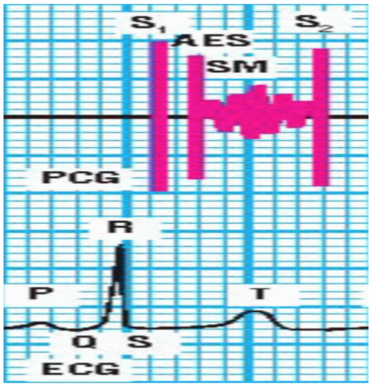
sound amplitude and ECG	Type of Murmur	outlook of altered heart valve sound
	Systolic	Supravalvular aortic stenosis murmurs: a left ventricular obstruction above the aortic valve produces this murmur which has a crescendo-decrescendo configuration.
	Systolic	Aortic valvular stenosis murmurs: this murmur is caused because after S1 the left ventricular pressure rises but the stenotic aortic valve doesn't open correctly.

Table 1: shows heart sounds variations due to different diseases. Figures adapted from [48].

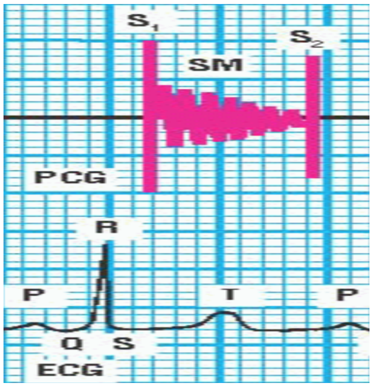
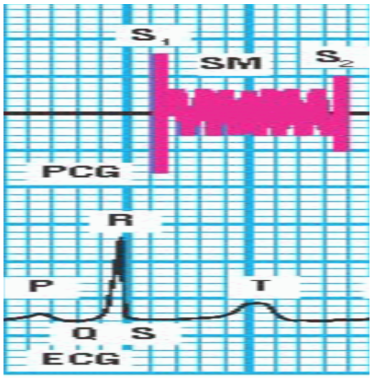
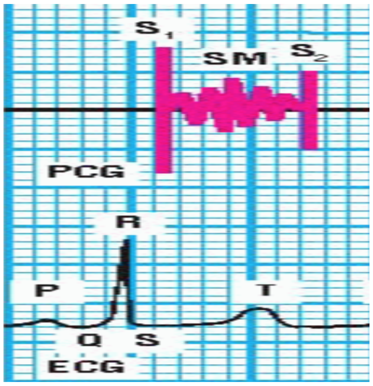
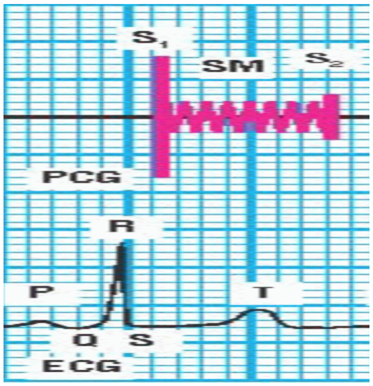
sound amplitude and ECG	Type of Murmur	outlook of altered heart valve sound
	Systolic	Acute mitral regurgitation murmur: this murmur can have many causes but it's usually due to severe damage to the mitral valve, which causes blood flowing backwards.
	Systolic	Tricuspid regurgitation murmurs: this murmur is die to ventricular dilation.
	Systolic	Subvalvular aortic stenosis murmurs: this happens when there is a left ventricular outflow obstruction below the aortic valve and it produces a murmur with a crescendo-decrescendo configuration.
	Systolic	Mitral valve prolapse murmurs: the murmur is caused by a prolapsed mitral valve

Table 2: shows heart sounds variations due to different diseases. Figures adapted from [48].

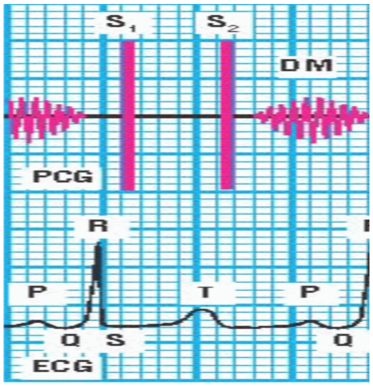
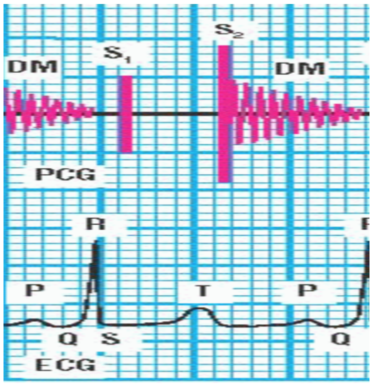
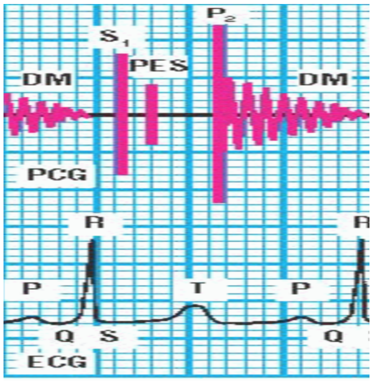
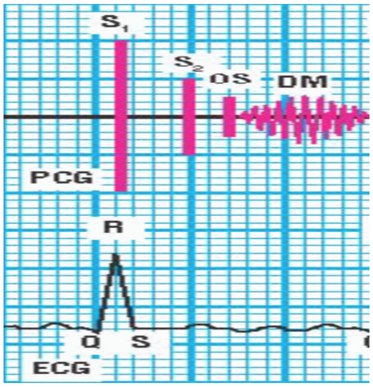
sound amplitude and ECG	Type of Murmur	outlook of altered heart valve sound
 S1 S2 DM PCG R P T P Q S Q ECG	Diastolic	Normal pressure pulmonic valve murmurs: this murmur is caused when the pulmonary artery diastolic pressure is too low thus blood flows backwards.
 DM S1 S2 DM PCG R P T P Q S Q ECG	Diastolic	Early diastolic aortic regurgitation murmurs: due to aortic pressure exceeding the left ventricular pressure at the beginning of diastole which causes the blood to backwards across an the aortic valve. This turbulent blood flow produces the murmur.
 P2 S1 PES DM DM PCG R R P T P Q S Q ECG	Diastolic	Mid diastolic aortic regurgitation murmurs: caused my the blood flowing backwards across the mitral valve.
	Diastolic	Mitral stenosis murmurs: this happens because of a scarred and calcified scar which then ceases to function normally. The murmur is produced by rapid, turbulent blood flow through a rigid, narrowed mitral valve opening.

Table 3: shows heart sounds variations due to different diseases. Figures adapted from [48].

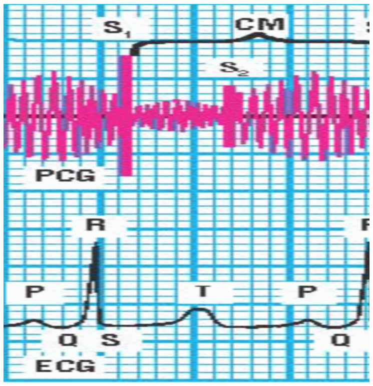
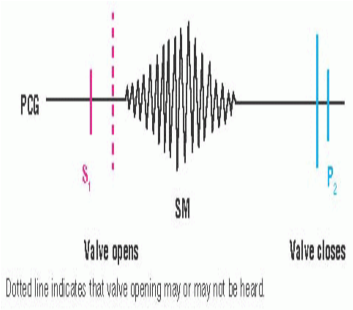
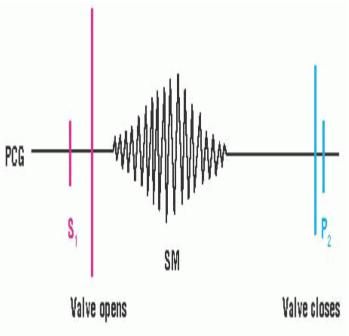
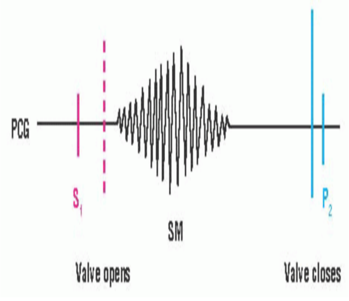
sound amplitude and ECG	Type of Murmur	outlook of altered heart valve sound
	Continuous	Cervical venous hum murmurs: this is caused by rapid downward blood flow through the jugular veins in the lower part of the neck.
 Dotted line indicates that valve opening may or may not be heard	Murmur from aortic tilting-disk	A problem with the aortic tilting-disk valve manifests through a murmur has a crescendo-decrescendo configuration.
	Murmur from aortic prosthetic ball-in-cage valve	Dysfunction of an aortic ball-in-cage valve prosthesis commonly causes the aortic opening and closing clicks to almost not be heard. It also causes murmurs.
	Murmur from aortic bileaflet valve	Aortic bileaflet valve murmur: the murmur has a rough or harsh quality.

Table 4: shows heart sounds variations due to different diseases. Figures adapted from [48].

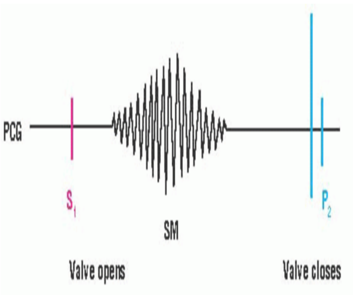
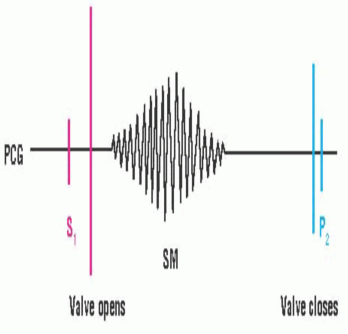
sound amplitude and ECG	Type of Murmur	outlook of altered heart valve sound
	Murmur from aortic porcine valve	A problem with the aortic porcine valve prosthesis may cause a diastolic murmur.
	Murmur from mitral ball-in-cage valve	Dysfunction of a mitral ball-in-cage can manifest through a holosystolic murmur which indicates mitral regurgitation.

Table 5: shows heart sounds variations due to different diseases. Figures adapted from [48].

Appendix C

Retrieval of Sound from Speckle Patterns Simulation Code

```
%% gaussian input
sigm =1000; %now this is in microns (beam size = 2.35 sigma)
N1 = 1024; %resolution
rng1 = 40000; % and this as well (field of view)
xx1 = linspace(-rng1,rng1,N1);
yy1 = linspace(-rng1,rng1,N1);
[X,Y] = meshgrid(xx1,yy1);
gaussian = exp(-(X.^2+Y.^2)/(2*sigm^2));%THIS LINE PRODUCES A ...
    GAUSSIAN SHAPED LASER BEAM
figure, imagesc(gaussian)

%%

%first_wall
continuos_phaseshift = zeros(N1);

av_grain_size = 200; %average bump size in microns
Nh =35; %this parameter controls the disorder => max frequency in the ...
    FT spectrum of the wall profile
wx = pi/(Nh*av_grain_size);
wy = pi/(Nh*av_grain_size);

for a = 1:Nh
    for b = 1:Nh
        continuos_phaseshift =continuos_phaseshift+(rand()-0.5)*\\
        sin(a*X*wx+(rand()-0.5)*2*pi).
        *sin(b*Y*wy+(rand()-0.5)*2*pi); %RAND FUNCTION USED TO ...
            SIMULATE FIRST WALL
    end
end

```

```
rz = 10;
smooth_first_wall = rescale(continuos_phaseshift,-rz,rz);
imagesc(abs(smooth_first_wall))

%% speckle pattern from the first wall

rescale_factor = 10; %improve the resolution to resolve the fresnel ...
    fringes
gaussian_big =imresize(gaussian,rescale_factor);%gaussian;%
figure,imagesc(gaussian_big)
smooth_first_wall_big =  imresize(smooth_first_wall,rescale_factor);
%smooth_first_wall;%
figure,imagesc(abs(smooth_first_wall_big))

lambda=0.5;
z = 100000;

N2 = length(gaussian_big);

xx2 = linspace(-rng1,rng1,N2);
yy2 = linspace(-rng1,rng1,N2);

[FX,FY]=meshgrid(xx2/(lambda*z),yy2/(lambda*z));

u=fftshift(fft2(fftshift(((gaussian_big).*exp(1j*
(smooth_first_wall_big+pi*(FX.^2+FY.^2)*(lambda*z)))))));
%THIS LINE PERFORMS THE 2D FT BETWEEN THE GAUSSIAN SHAPED BEAM AND ...
    THE SIMULATED WALL AND IT INCLUDES THE TRASNFER FUNCTION H, WHICH, ...
    AS STATED IN THE FORMULA IS e^(jkz)*e[-i*pi*yz(fx^2+fy^2)]
figure,imagesc('XData',xx2,'YData',yy2,'CData',abs(u))


freq_nyquist = 0.5/(2*rng1/N2);
rng2 = lambda* freq_nyquist*z;
xx02 = linspace(-rng2,rng2,N2);
yy02 = xx02;

u_small_res = imresize(u,0.05); %reduce back the resolution to ...
    improve speed and for the FT
clear u
clear gaussian_big
clear smooth_first_wall_big
N3 = length(u_small_res);
figure,imagesc('XData',xx02,'YData',yy02,'CData',abs(u))
%%
%%SECOND WALL--
```

```
[X,Y] = meshgrid(linspace(-rng2,rng2,N3),linspace(-rng2,rng2,N3));
continuos_phaseshift1 = zeros(N3);

av_grain_size = 200; %is change: average bump size in microns
Nh =35;
wx1 = pi/(Nh*av_grain_size);
wy1 = pi/(Nh*av_grain_size);


for a = 1:Nh
    for b = 1:Nh
        continuos_phaseshift1 ...
            =continuos_phaseshift1+(rand()-0.5)*2*sin(a*X*wx1+
        (rand()-0.5)*2*pi).*sin(b*Y*wy1+(rand()-0.5)*2*pi);
    end
end

second_wall = rescale(continuos_phaseshift1,-27,27);

figure,imagesc('XData',xx02,'YData',yy02,'CData',abs(second_wall))
%% SECOND SPECKLE
%making some chirp sound
Ns = 1000;
fs = 10e2; %sampling frequency 20 KHz bc we want to reach max vocal ...
    sound.
t = 0 : 1/fs : 1;
y = chirp(t,110,10,15*110);
sound(y,fs)
y1=rescale(y,-1,1);


z0 = 100000;

vibration_amp = 100;
sec_sp_resized_crop = ...
    imresize(zeros(Nsec_sp,Nsec_sp),(z0-vibration_amp)/z0);
sp_min_sz = size(sec_sp_resized_crop);
sec_speckle = zeros([sp_min_sz,Ns]);



xx3 = linspace(-rng2,rng2,N3);
yy3 = linspace(-rng2,rng2,N3);

for i=1:1000
    z =  z0 + vibration_amp*y1(i);
    [FX,FY]=meshgrid(xx3/(lambda*z),yy3/(lambda*z));
    sec_sp_initial = fftshift(fft2(fftshift(abs(u_small_res)
    *exp(1j*((angle(u_small_res))+
    (second_wall+pi*lambda*z*(FX.^2+FY.^2)))))));
    sec_sp_resized = imresize(sec_sp_initial,z/z0);
    sp_cur_size = length(sec_sp_resized);
    Nt = 1+floor((sp_cur_size-sp_min_sz)/2);
    sec_speckle(:,:,i) = ...
        sec_sp_resized(Nt:Nt+sp_min_sz-1,Nt:Nt+sp_min_sz-1);
    %place the image in the center of a bigger zero padded array.
    sec_speckle(:,:,i) = ...
        sec_speckle(:,:,i)./sum(sum(abs(sec_speckle(:,:,i))));
    %  added abs inside the sums
    i
end
%%
% sec_spec_resolved =  imresize(sec_speckle,4);
% soun=rescale(abs(sec_spec_resolved(:,:,:)),-1,1);
summ = sum(sum(soun,1),2);sound(squeeze(summ),1000)


summ = squeeze(sum(sum(abs(sec_speckle(:,:,:)),1),2));
sound(rescale(summ,-1,1),1000);
%%
sound(rescale(squeeze(abs(sec_speckle(250,250,:))),-1,1),1000);
%%
figure()
subplot(1,2,1)
plot(squeeze(abs(sec_speckle(250,251,1:100))))
hold on
plot(squeeze(abs(sec_speckle(250,250,1:100))))
plot(squeeze(abs(sec_speckle(249,250,1:100))))
plot(squeeze(abs(sec_speckle(250,253,1:100))))
plot(squeeze(abs(sec_speckle(248,251,1:100))))
plot(squeeze(abs(sec_speckle(247,253,1:100))))

subplot(1,2,2)
plot(summ)
%%
freq_nyquist = 0.5/(2*rng2/N3);
rng3 = lambda* freq_nyquist*z;
xx3 = linspace(-rng2,rng2,N3);
yy3 = xx3;

figure,imagesc('XData',xx3,'YData',yy3,'CData',abs(sec_speckle(:,:,100)))

```

```


%% Third speckle

thi_speckle = zeros(size(sec_speckle));

z = 100000;

xx3 = linspace(-rng3,rng3,N3);
yy3 = linspace(-rng3,rng3,N3);

[FX,FY]=meshgrid(xx3/(lambda*z),yy3/(lambda*z));


for i=1:1000
    thi_speckle(:,:,i) = fftshift(fft2(fftshift(abs(sec_speckle(:,:,i))
    .*exp(1j*((angle(sec_speckle(:,:,i)))+(second_wall+pi*lambda*(z)
    *(FX.^2+FY.^2)))))));
    thi_speckle(:,:,i) = ...
        thi_speckle(:,:,i)./sum(sum(abs(thi_speckle(:,:,i))));
    %  added abs inside the sums
    i
end


% soun=rescale(abs(thi_speckle(200,200,:)),-1,1);
summ = sum(sum(soun,1),2);sound(squeeze(summ),1000)
% thi_spec_resolved =  imresize(thi_speckle,4); %this is very slow
%%
summ1 = squeeze(sum(sum(abs(thi_speckle),1),2));
sound(rescale(summ1,-1,1),1000);
%%
sound(rescale(squeeze(abs(thi_speckle(250,250,:))),-1,1),1000);
%%
figure()
subplot(1,2,1)
plot(squeeze(abs(thi_speckle(250,251,1:100))))
hold on
plot(squeeze(abs(thi_speckle(250,250,1:100))))
plot(squeeze(abs(thi_speckle(249,250,1:100))))
plot(squeeze(abs(thi_speckle(250,253,1:100))))
plot(squeeze(abs(thi_speckle(248,251,1:100))))
plot(squeeze(abs(thi_speckle(247,253,1:100))))

subplot(1,2,2)
plot(summ1)
```

```
%% I didn't go here
N04=length(thi_spec_resolved(:,:,1))
freq_nyquist = 0.5/(2*rng3/N04);
rng04 = lambda* freq_nyquist*z;
xx04 = linspace(-rng03,rng03,N04);
yy04 = xx04;

figure,imagesc('XData',xx04,'YData',yy04,'CData',
abs(thi_spec_resolved(:,:,100)))

%%third wall
camera_act=ones(2,2); %CAMERA ACTIVE AREA
camera_pitch= zeros(50,50);%CAMERA PITCH
camera_pitch(25,25)=camera_act(:,:);
camera=repmat(camera_pitch,320);  %THIS LINE SIMULATES THE SPAD ARRAY
image=sec_speckle.*camera;
```

Appendix D

Heart Health Assessment Code

```
 %%SVM
%%
Labels_mine=heartSoundData.Classes;

afibX = Signals_mine4(Labels_mine=='abnormal');
afibY = Labels_mine(Labels_mine=='abnormal');

normalX = Signals_mine4(Labels_mine=='normal');
normalY = Labels_mine(Labels_mine=='normal');

[trainIndA,¬,testIndA] = dividerand(1254,0.9,0.0,0.1);
[trainIndN,¬,testIndN] = dividerand(2575,0.9,0.0,0.1);

XTrainA = afibX(trainIndA);
YTrainA = afibY(trainIndA);

XTrainN = normalX(trainIndN);
YTrainN = normalY(trainIndN);

XTestA = afibX(testIndA);
YTestA = afibY(testIndA);

XTestN = normalX(testIndN);
YTestN = normalY(testIndN);

%%%%
%train abnormal=1129   train normal = 2317
%test abnormal= 125    test normal= 258

XTrain = [repmat(XTrainA(1:1129),2,1); XTrainN(1:2317)];
YTrain = [repmat(YTrainA(1:1129),2,1); YTrainN(1:2317)];

```

```
XTest = [XTestA(1:125); XTestN(1:250)];
YTest = [YTestA(1:125); YTestN(1:250);];

summary(YTrain)
summary(YTest)



N=10000;
sn = waveletScattering('SignalLength',N,'InvarianceScale',N,
'QualityFactor',[16]);%,'oversamplingFactor',1);
%WHERE YOU PUT THE DATA
XTrain=cell2mat(XTrain);XTrain=reshape(XTrain,[10000,4575]);
scat_features_train = featureMatrix(sn,XTrain.','Transform','log');
%[S,U] = scatteringTransform(sf,res);
%scattergram(sf,U,'FilterBank',1)

Nseq = size(scat_features_train,2);
scat_features_train = permute(scat_features_train,[2 3 1]);
scat_features_train = reshape(scat_features_train,...
    size(scat_features_train,1)*size(scat_features_train,2),[]);

XTest=cell2mat(XTest);XTest=reshape(XTest,[10000,375]).';
scat_features_test = featureMatrix(sn,XTest.','Transform','log');

scat_features_test = permute(scat_features_test,[2 3 1]);
scat_features_test = reshape(scat_features_test,...
    size(scat_features_test,1)*size(scat_features_test,2),[]);

[sequence_labels_train,sequence_labels_test] = ...
    createSequenceLabels_heartsounds(Nseq,YTrain,YTest);


features = [scat_features_train; scat_features_test];
rng(1)
template = templateSVM(...
    'KernelFunction','polynomial',...
    'PolynomialOrder',6,...
    'KernelScale','auto',...
    'BoxConstraint',1,...
    'Standardize',true);
model_N_A_try_3 = fitcecoc(...
    features,...
    [sequence_labels_train;sequence_labels_test],...
    'Learners',template,...
    'Coding','onevsone',...
    'ClassNames',{'abnormal','normal'});
kfoldmodel = crossval(model_N_A_try_2,'KFold',5);
classLabels = kfoldPredict(kfoldmodel);
loss = kfoldLoss(kfoldmodel)*100
%%




%%test data needs to go through
%%
scat_features_test = ...
    featureMatrix(sn,realData((10000*(a-1)+1):(10000*a)).','Transform',
'log');

scat_features_test = permute(scat_features_test,[2 3 1]);
scat_features_test = reshape(scat_features_test,...
    size(scat_features_test,1)*size(scat_features_test,2),[]);


predLabels = predict(model_N_A_try_2,scat_features_test)
%%




%%TSNE
%%
for a=1:5
    state(a)={'tay'}
end
state=categorical(state)
state=state.'
 Labels_mine1=[sequence_labels_train ;state]


scat_all=[scat_features_train;scat_features_test];



Y5 = tsne(scat_all,'Algorithm','barneshut','NumPCAComponents
',30,'Perplexity',5);
gscatter(Y5(:,1),Y5(:,2),Labels_mine1,'ymc','*oX*',8)
```

Appendix E

Biometric Authentication Code

```

[b,a]=butter(5,100/750,'high') %THIS IS THE FILTER TO BE USED IF YOU ...
    WANT TO ONLY KEEP HIGH FREQUENCIES
%ABOVE 500 HZ

ypos_person1_1=ypos_person1_1(1,1:44100); % NEED TO MAKE THE DATA AN ...
    EVEN NUMBER SO WHEN YOU DIVIDE YOU GET EQUAL NUMBER IN EACH SET
ypos_person1_2=ypos_person1_2(1,1:44100);
ypos_person1_3=ypos_person1_3(1,1:44100);
ypos_person1_4=ypos_person1_4(1,1:44100);
ypos_person1_5=ypos_person1_5(1,1:44100);
ypos_person1_6=ypos_person1_6(1,1:44100);
ypos_person1_7=ypos_person1_7(1,1:44100);
ypos_person1_8=ypos_person1_8(1,1:44100);
ypos_person1_9=ypos_person1_9(1,1:44100);


ypos_person1_11=filter(b,a,ypos_person1_1);  %APPLY YOUR HIGH PASS ...
    FILTER TO EACH DATA SET
ypos_person1_21=filter(b,a,ypos_person1_2);
ypos_person1_31=filter(b,a,ypos_person1_3);
ypos_person1_41=filter(b,a,ypos_person1_4);
ypos_person1_51=filter(b,a,ypos_person1_5);
ypos_person1_61=filter(b,a,ypos_person1_6);
ypos_person1_71=filter(b,a,ypos_person1_7);
ypos_person1_81=filter(b,a,ypos_person1_8);
ypos_person1_91=filter(b,a,ypos_person1_9);



%NEED TO RESHAPE YOUR DATA IN ORDER TO FEED IT TO THE SCATTERING ...
    TRANSFORM
```

```
%AND SVM
all_ypos_person1_1= [ypos_person1_11;ypos_person1_21;ypos_person1_31;
ypos_person1_41;ypos_person1_51;ypos_person1_61
;ypos_person1_71;ypos_person1_81;ypos_person1_91];
all_ypos_person1_1= mat2cell( all_ypos_person1_1,
[1 1 1 1 1 1 1 1 1 ],
[3675 3675 3675 3675 3675 3675  3675   3675   3675  3675   3675   3675])
all_ypos_person1_1= reshape(all_ypos_person1_1,[108,1])
all_ypos_person1_1 = cell2mat(all_ypos_person1_1)


%%
ypos_person2_1=ypos_person2_1(1,1:44100);
ypos_person2_2=ypos_person2_2(1,1:44100);
ypos_person2_3=ypos_person2_3(1,1:44100);
ypos_person2_4=ypos_person2_4(1,1:44100);
ypos_person2_5=ypos_person2_5(1,1:44100);
ypos_person2_6=ypos_person2_6(1,1:44100);
ypos_person2_7=ypos_person2_7(1,1:44100);
ypos_person2_8=ypos_person2_8(1,1:44100);
ypos_person2_9=ypos_person2_9(1,1:44100);



ypos_person2_11=filter(b,a,ypos_person2_1);
ypos_person2_21=filter(b,a,ypos_person2_2);
ypos_person2_31=filter(b,a,ypos_person2_3);
ypos_person2_41=filter(b,a,ypos_person2_4);
ypos_person2_51=filter(b,a,ypos_person2_5);
ypos_person2_61=filter(b,a,ypos_person2_6);
ypos_person2_71=filter(b,a,ypos_person2_7);
ypos_person2_81=filter(b,a,ypos_person2_8);
ypos_person2_91=filter(b,a,ypos_person2_9);




all_ypos_person2_1= [ypos_person2_11;ypos_person2_21;ypos_person2_31;
ypos_person2_41;ypos_person2_51;ypos_person2_61;
ypos_person2_71;ypos_person2_81;ypos_person2_91];
all_ypos_person2_1= mat2cell(all_ypos_person2_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
    3675  3675   3675   3675])
all_ypos_person2_1= reshape(all_ypos_person2_1,[108,1])
all_ypos_person2_1 = cell2mat(all_ypos_person2_1)

```

```


%%

ypos_person3_1=ypos_person3_1(1,1:44100);
ypos_person3_2=ypos_person3_2(1,1:44100);
ypos_person3_3=ypos_person3_3(1,1:44100);
ypos_person3_4=ypos_person3_4(1,1:44100);
ypos_person3_5=ypos_person3_5(1,1:44100);
ypos_person3_6=ypos_person3_6(1,1:44100);
ypos_person3_7=ypos_person3_7(1,1:44100);
ypos_person3_8=ypos_person3_8(1,1:44100);
ypos_person3_9=ypos_person3_9(1,1:44100);



ypos_person3_11=filter(b,a,ypos_person3_1);
ypos_person3_21=filter(b,a,ypos_person3_2);
ypos_person3_31=filter(b,a,ypos_person3_3);
ypos_person3_41=filter(b,a,ypos_person3_4);
ypos_person3_51=filter(b,a,ypos_person3_5);
ypos_person3_61=filter(b,a,ypos_person3_6);
ypos_person3_71=filter(b,a,ypos_person3_7);
ypos_person3_81=filter(b,a,ypos_person3_8);
ypos_person3_91=filter(b,a,ypos_person3_9);


all_ypos_person3_1= [ypos_person3_11;ypos_person3_21;ypos_person3_31;
ypos_person3_41;ypos_person3_51;ypos_person3_61;
ypos_person3_71;ypos_person3_81;ypos_person3_91];
all_ypos_person3_1= mat2cell(all_ypos_person3_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
   3675  3675   3675   3675])
all_ypos_person3_1= reshape(all_ypos_person3_1,[108,1])
all_ypos_person3_1 = cell2mat(all_ypos_person3_1)


%%
ypos_person4_1=ypos_person4_1(1,1:44100);
ypos_person4_2=ypos_person4_2(1,1:44100);
ypos_person4_3=ypos_person4_3(1,1:44100);
ypos_person4_4=ypos_person4_4(1,1:44100);
ypos_person4_5=ypos_person4_5(1,1:44100);
ypos_person4_6=ypos_person4_6(1,1:44100);
ypos_person4_7=ypos_person4_7(1,1:44100);
ypos_person4_8=ypos_person4_8(1,1:44100);
```

```
ypos_person4_9=ypos_person4_9(1,1:44100);


ypos_person4_11=filter(b,a,ypos_person4_1);
ypos_person4_21=filter(b,a,ypos_person4_2);
ypos_person4_31=filter(b,a,ypos_person4_3);
ypos_person4_41=filter(b,a,ypos_person4_4);
ypos_person4_51=filter(b,a,ypos_person4_5);
ypos_person4_61=filter(b,a,ypos_person4_6);
ypos_person4_71=filter(b,a,ypos_person4_7);
ypos_person4_81=filter(b,a,ypos_person4_8);
ypos_person4_91=filter(b,a,ypos_person4_9);



all_ypos_person4_1= [ypos_person4_11;ypos_person4_21;ypos_person4_31;
ypos_person4_41;ypos_person4_51;ypos_person4_61;
ypos_person4_71;ypos_person4_81;ypos_person4_91];
all_ypos_person4_1= mat2cell(all_ypos_person4_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
   3675  3675   3675   3675])
all_ypos_person4_1= reshape(all_ypos_person4_1,[108,1])
all_ypos_person4_1 = cell2mat(all_ypos_person4_1)

%%
ypos_person5_1=ypos_person5_1(1,1:44100);
ypos_person5_2=ypos_person5_2(1,1:44100);
ypos_person5_3=ypos_person5_3(1,1:44100);
ypos_person5_4=ypos_person5_4(1,1:44100);
ypos_person5_5=ypos_person5_5(1,1:44100);
ypos_person5_6=ypos_person5_6(1,1:44100);
ypos_person5_7=ypos_person5_7(1,1:44100);
ypos_person5_8=ypos_person5_8(1,1:44100);
ypos_person5_9=ypos_person5_9(1,1:44100);


ypos_person5_11=filter(b,a,ypos_person5_1);
ypos_person5_21=filter(b,a,ypos_person5_2);
ypos_person5_31=filter(b,a,ypos_person5_3);
ypos_person5_41=filter(b,a,ypos_person5_4);
ypos_person5_51=filter(b,a,ypos_person5_5);
ypos_person5_61=filter(b,a,ypos_person5_6);
ypos_person5_71=filter(b,a,ypos_person5_7);
ypos_person5_81=filter(b,a,ypos_person5_8);
ypos_person5_91=filter(b,a,ypos_person5_9);

```

```

all_ypos_person5_1= [ypos_person5_11;ypos_person5_21;ypos_person5_31;
ypos_person5_41;ypos_person5_51;ypos_person5_61;
ypos_person5_71;ypos_person5_81;ypos_person5_91];
all_ypos_person5_1= mat2cell(all_ypos_person5_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
    3675  3675   3675   3675])
all_ypos_person5_1= reshape(all_ypos_person5_1,[108,1])
all_ypos_person5_1 = cell2mat(all_ypos_person5_1)

%%
ypos_person6_1=ypos_person6_1(1,1:44100);
ypos_person6_2=ypos_person6_2(1,1:44100);
ypos_person6_3=ypos_person6_3(1,1:44100);
ypos_person6_4=ypos_person6_4(1,1:44100);
ypos_person6_5=ypos_person6_5(1,1:44100);
ypos_person6_6=ypos_person6_6(1,1:44100);
ypos_person6_7=ypos_person6_7(1,1:44100);
ypos_person6_8=ypos_person6_8(1,1:44100);
ypos_person6_9=ypos_person6_9(1,1:44100);

ypos_person6_11=filter(b,a,ypos_person6_1);
ypos_person6_21=filter(b,a,ypos_person6_2);
ypos_person6_31=filter(b,a,ypos_person6_3);
ypos_person6_41=filter(b,a,ypos_person6_4);
ypos_person6_51=filter(b,a,ypos_person6_5);
ypos_person6_61=filter(b,a,ypos_person6_6);
ypos_person6_71=filter(b,a,ypos_person6_7);
ypos_person6_81=filter(b,a,ypos_person6_8);
ypos_person6_91=filter(b,a,ypos_person6_9);


all_ypos_person6_1= [ypos_person6_11;ypos_person6_21;ypos_person6_31;
ypos_person6_41;ypos_person6_51;ypos_person6_61;
ypos_person6_71;ypos_person6_81;ypos_person6_91];
all_ypos_person6_1= mat2cell(all_ypos_person6_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
    3675  3675   3675   3675])
all_ypos_person6_1= reshape(all_ypos_person6_1,[108,1])
all_ypos_person6_1 = cell2mat(all_ypos_person6_1)



%%
%%PAT
ypos_person7_1=ypos_person7_1(1,1:44100);
```

```
ypos_person7_2=ypos_person7_2(1,1:44100);
ypos_person7_3=ypos_person7_3(1,1:44100);
ypos_person7_4=ypos_person7_4(1,1:44100);
ypos_person7_5=ypos_person7_5(1,1:44100);
ypos_person7_6=ypos_person7_6(1,1:44100);
ypos_person7_7=ypos_person7_7(1,1:44100);
ypos_person7_8=ypos_person7_8(1,1:44100);
ypos_person7_9=ypos_person7_9(1,1:44100);

ypos_person7_11=filter(b,a,ypos_person7_1);
ypos_person7_21=filter(b,a,ypos_person7_2);
ypos_person7_31=filter(b,a,ypos_person7_3);
ypos_person7_41=filter(b,a,ypos_person7_4);
ypos_person7_51=filter(b,a,ypos_person7_5);
ypos_person7_61=filter(b,a,ypos_person7_6);
ypos_person7_71=filter(b,a,ypos_person7_7);
ypos_person7_81=filter(b,a,ypos_person7_8);
ypos_person7_91=filter(b,a,ypos_person7_9);

all_ypos_person7_1= [ypos_person7_11;ypos_person7_21;ypos_person7_31;
ypos_person7_41;ypos_person7_51;ypos_person7_61;
ypos_person7_71;ypos_person7_81;ypos_person7_91];
all_ypos_person7_1= mat2cell(all_ypos_person7_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
   3675  3675   3675   3675])
all_ypos_person7_1= reshape(all_ypos_person7_1,[108,1])
all_ypos_person7_1 = cell2mat(all_ypos_person7_1)

%%
ypos_person8_1=ypos_person8_1(1,1:44100);
ypos_person8_2=ypos_person8_2(1,1:44100);
ypos_person8_3=ypos_person8_3(1,1:44100);
ypos_person8_4=ypos_person8_4(1,1:44100);
ypos_person8_5=ypos_person8_5(1,1:44100);
ypos_person8_6=ypos_person8_6(1,1:44100);
ypos_person8_7=ypos_person8_7(1,1:44100);
ypos_person8_8=ypos_person8_8(1,1:44100);
ypos_person8_9=ypos_person8_9(1,1:44100);

ypos_person8_11=filter(b,a,ypos_person8_1);
ypos_person8_21=filter(b,a,ypos_person8_2);
ypos_person8_31=filter(b,a,ypos_person8_3);
ypos_person8_41=filter(b,a,ypos_person8_4);
ypos_person8_51=filter(b,a,ypos_person8_5);
ypos_person8_61=filter(b,a,ypos_person8_6);
ypos_person8_71=filter(b,a,ypos_person8_7);
```

```
ypos_person8_81=filter(b,a,ypos_person8_8);
ypos_person8_91=filter(b,a,ypos_person8_9);

all_ypos_person8_1= [ypos_person8_1;ypos_person8_2;ypos_person8_3;
ypos_person8_4;ypos_person8_5;ypos_person8_6;
ypos_person8_7;ypos_person8_8;ypos_person8_9];
all_ypos_person8_1= mat2cell(all_ypos_person8_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
    3675  3675   3675   3675])
all_ypos_person8_1= reshape(all_ypos_person8_1,[108,1])
all_ypos_person8_1 = cell2mat(all_ypos_person8_1)


%%
ypos_person9_1=ypos_person9_1(1,1:44100);
ypos_person9_2=ypos_person9_2(1,1:44100);
ypos_person9_3=ypos_person9_3(1,1:44100);
ypos_person9_4=ypos_person9_4(1,1:44100);
ypos_person9_5=ypos_person9_5(1,1:44100);
ypos_person9_6=ypos_person9_6(1,1:44100);
ypos_person9_7=ypos_person9_7(1,1:44100);
ypos_person9_8=ypos_person9_8(1,1:44100);
ypos_person9_9=ypos_person9_9(1,1:44100);

ypos_person9_11=filter(b,a,ypos_person9_1);
ypos_person9_21=filter(b,a,ypos_person9_2);
ypos_person9_31=filter(b,a,ypos_person9_3);
ypos_person9_41=filter(b,a,ypos_person9_4);
ypos_person9_51=filter(b,a,ypos_person9_5);
ypos_person9_61=filter(b,a,ypos_person9_6);
ypos_person9_71=filter(b,a,ypos_person9_7);
ypos_person9_81=filter(b,a,ypos_person9_8);
ypos_person9_91=filter(b,a,ypos_person9_9);

all_ypos_person9_1= [ypos_person9_11;ypos_person9_21;ypos_person9_31;
ypos_person9_41;ypos_person9_51;ypos_person9_61;
ypos_person9_71;ypos_person9_81;ypos_person9_91];
all_ypos_person9_1= mat2cell(all_ypos_person9_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
    3675  3675   3675   3675])
all_ypos_person9_1= reshape(all_ypos_person9_1,[108,1])
all_ypos_person9_1 = cell2mat(all_ypos_person9_1)
%%
ypos_person10_1=ypos_person10_1(1,1:44100);
ypos_person10_2=ypos_person10_2(1,1:44100);
ypos_person10_3=ypos_person10_3(1,1:44100);
```

```
ypos_person10_4=ypos_person10_4(1,1:44100);
ypos_person10_5=ypos_person10_5(1,1:44100);
ypos_person10_6=ypos_person10_6(1,1:44100);
ypos_person10_7=ypos_person10_7(1,1:44100);
ypos_person10_8=ypos_person10_8(1,1:44100);
ypos_person10_9=ypos_person10_9(1,1:44100);


ypos_person10_11=filter(b,a,ypos_person10_1);
ypos_person10_21=filter(b,a,ypos_person10_2);
ypos_person10_31=filter(b,a,ypos_person10_3);
ypos_person10_41=filter(b,a,ypos_person10_4);
ypos_person10_51=filter(b,a,ypos_person10_5);
ypos_person10_61=filter(b,a,ypos_person10_6);
ypos_person10_71=filter(b,a,ypos_person10_7);
ypos_person10_81=filter(b,a,ypos_person10_8);
ypos_person10_91=filter(b,a,ypos_person10_9);

all_ypos_person10_1= [ypos_person10_11;ypos_person10_21;ypos_person10_31;
ypos_person10_41;ypos_person10_51;ypos_person10_61;
ypos_person10_71;ypos_person10_81;ypos_person10_91];
all_ypos_person10_1= mat2cell(all_ypos_person10_1,
[1 1 1 1 1 1 1 1 1 ],[3675 3675 3675 3675 3675 3675  3675   3675   ...
    3675  3675   3675   3675])
all_ypos_person10_1= reshape(all_ypos_person10_1,[108,1])
all_ypos_person10_1 = cell2mat(all_ypos_person10_1)










%%

%% NEED TO MAKE CATHEGORICAL STATES FOR EACH CLASS (LABELS)



for a=1:108
    state1(a)={'P_1'}
end
state1 = categorical(state1)
```

```
for a=1:108
    state2(a)={'P_2'}
end
state2 = categorical(state2)
for a=1:108
    state3(a)={'P_3'}
end
state3 = categorical(state3)
for a=1:108
    state4(a)={'P_4'}
end
state4 = categorical(state4)
for a=1:108
    state5(a)={'P_5'}
end
state5 = categorical(state5)
for a=1:108
    state6(a)={'P_6'}
end
state6 = categorical(state6)
for a=1:108
    state7(a)={'P_7'}
end
state7 = categorical(state7)
for a=1:108
    state8(a)={'P_8'}
end
state8 = categorical(state8)
for a=1:108
    state9(a)={'P_9'}
end
state9 = categorical(state9)
for a=1:108
    state10(a)={'P_10'}
end
state10 = categorical(state10)


state_all=[state1 state2 state3 state4 state5 state6 state7 state8 ...
    state9 state10]

classes=state_all.'

%%


```

```

all_data=[all_ypos_person1_1;all_ypos_person2_1;
all_ypos_person3_1;all_ypos_person4_1;all_ypos_person5_1;
all_ypos_person6_1;all_ypos_person7_1;all_ypos_person8_1;
all_ypos_person9_1;all_ypos_person10_1]; %PUT DATA TOGETHER



all_data=all_data(:,1:2:3674); %RESAMPLE

all_data_res=zeros(1080,1837);
for a = 1:1080
   all_data_res(a,:)=rescale(all_data(a,:),-1,1); %RESCALE
end




Signals=num2cell(all_data_res,2);
Signals1=Signals  %RESHAPE

P_1X = Signals1(classes=='P_1');  %ASSIGN DATA TO LABEL
P_1Y = classes(classes=='P_1');

P_2X = Signals1(classes=='P_2');
P_2Y = classes(classes=='P_2');

P_3X = Signals1(classes=='P_3');
P_3Y = classes(classes=='P_3');

P_4X = Signals1(classes=='P_4');
P_4Y = classes(classes=='P_4');

P_5X = Signals1(classes=='P_5');
P_5Y = classes(classes=='P_5');

P_6X = Signals1(classes=='P_6');
P_6Y = classes(classes=='P_6');

P_7X = Signals1(classes=='P_7');
P_7Y = classes(classes=='P_7');

P_8X = Signals1(classes=='P_8');
P_8Y = classes(classes=='P_8');

P_9X = Signals1(classes=='P_9');
```

```
P_9Y = classes(classes=='P_9');

P_10X = Signals1(classes=='P_10');
P_10Y = classes(classes=='P_10');




[trainIndP_1,¬,testIndP_1] = dividerand(108,0.9,0.0,0.1);  %DIVIDE ...
    TRAINING AND TESTING
[trainIndP_2,¬,testIndP_2] = dividerand(108,0.9,0.0,0.1);
[trainIndP_3,¬,testIndP_3] = dividerand(108,0.9,0.0,0.1);
[trainIndP_4,¬,testIndP_4] = dividerand(108,0.9,0.0,0.1);
[trainIndP_5,¬,testIndP_5] = dividerand(108,0.9,0.0,0.1);
[trainIndP_6,¬,testIndP_6] = dividerand(108,0.9,0.0,0.1);
[trainIndP_7,¬,testIndP_7] = dividerand(108,0.9,0.0,0.1);
[trainIndP_8,¬,testIndP_8] = dividerand(108,0.9,0.0,0.1);
[trainIndP_9,¬,testIndP_9] = dividerand(108,0.9,0.0,0.1);
[trainIndP_10,¬,testIndP_10] = dividerand(108,0.9,0.0,0.1);



XTrainP_1 = P_1X(trainIndP_1);
YTrainP_1 = P_1Y(trainIndP_1);

XTrainP_2 = P_2X(trainIndP_2);
YTrainP_2 = P_2Y(trainIndP_2);

XTrainP_3 = P_3X(trainIndP_3);
YTrainP_3 = P_3Y(trainIndP_3);

XTrainP_4 = P_4X(trainIndP_4);
YTrainP_4 = P_4Y(trainIndP_4);

XTrainP_5 = P_5X(trainIndP_5);
YTrainP_5 = P_5Y(trainIndP_5);

XTrainP_6 = P_6X(trainIndP_6);
YTrainP_6 = P_6Y(trainIndP_6);

XTrainP_7 = P_7X(trainIndP_7);
YTrainP_7 = P_7Y(trainIndP_7);

XTrainP_8 = P_8X(trainIndP_8);
YTrainP_8 = P_8Y(trainIndP_8);

```

```
XTrainP_9 = P_9X(trainIndP_9);
YTrainP_9 = P_9Y(trainIndP_9);

XTrainP_10 = P_10X(trainIndP_10);
YTrainP_10 =P_10Y(trainIndP_10);





XTestP_1 = P_1X(testIndP_1);
YTestP_1 = P_1Y(testIndP_1);

XTestP_2 = P_2X(testIndP_2);
YTestP_2=P_2Y(testIndP_2);

XTestP_3 = P_3X(testIndP_3);
YTestP_3= P_3Y(testIndP_3);

XTestP_4 = P_4X(testIndP_4);
YTestP_4= P_4Y(testIndP_4);

XTestP_5 = P_5X(testIndP_5);
YTestP_5= P_5Y(testIndP_5);

XTestP_6 = P_6X(testIndP_6);
YTestP_6= P_6Y(testIndP_6);

XTestP_7= P_7X(testIndP_7);
YTestP_7= P_7Y(testIndP_7);

XTestP_8= P_8X(testIndP_8);
YTestP_8= P_8Y(testIndP_8);

XTestP_9= P_9X(testIndP_9);
YTestP_9=P_9Y(testIndP_9);

XTestP_10= P_10X(testIndP_10);
YTestP_10= P_10Y(testIndP_10);



XTrain = [XTrainP_1(1:97); XTrainP_2(1:97); XTrainP_3(1:97); ...
    XTrainP_4(1:97);XTrainP_5(1:97);XTrainP_6(1:97);
XTrainP_7(1:97);XTrainP_8(1:97);XTrainP_9(1:97);
XTrainP_10(1:97)];
```

```
YTrain = [YTrainP_1(1:97); YTrainP_2(1:97); YTrainP_3(1:97); ...
   YTrainP_4(1:97); YTrainP_5(1:97); ...
   YTrainP_6(1:97);YTrainP_7(1:97);YTrainP_8(1:97);
YTrainP_9(1:97);YTrainP_10(1:97)];

XTest = [XTestP_1(1:11); XTestP_2(1:11); XTestP_3(1:11); ...
   XTestP_4(1:11);XTestP_5(1:11);XTestP_6(1:11);XTestP_7(1:11);
XTestP_8(1:11);XTestP_9(1:11);XTestP_10(1:11)];
YTest = [YTestP_1(1:11); YTestP_2(1:11); YTestP_3(1:11);
YTestP_4(1:11);YTestP_5(1:11);YTestP_6(1:11);YTestP_7(1:11);
YTestP_8(1:11);YTestP_9(1:11);YTestP_10(1:11)];

summary(YTest)

%FEATURE EXTRACTION


N = 1837;

sn = waveletScattering('SignalLength',N,'InvarianceScale',
N,'QualityFactor',[8 4 1]);


XTrain=cell2mat(XTrain).';


scat_features_train = featureMatrix(sn,XTrain,'Transform','log');

Nseq = size(scat_features_train,2);
scat_features_train = permute(scat_features_train,[2 3 1]);
scat_features_train = reshape(scat_features_train,...
    size(scat_features_train,1)*size(scat_features_train,2),[]);


XTest=cell2mat(XTest).';
scat_features_test = featureMatrix(sn,XTest,'Transform','log');
scat_features_test = permute(scat_features_test,[2 3 1]);
scat_features_test = reshape(scat_features_test,...
    size(scat_features_test,1)*size(scat_features_test,2),[]);



[sequence_labels_train,sequence_labels_test] = ...
    createSequenceLabels_heartsounds(Nseq,YTrain,YTest);

%% CLASSIFICATION
```

```

features = [scat_features_train; scat_features_test];
rng(1)
template = templateSVM(...
    'KernelFunction','polynomial',...
    'PolynomialOrder',3,...
    'KernelScale','auto',...
    'BoxConstraint',1,...
    'Standardize',true);
model_butter_band20350= fitcecoc(...
    features,...
    [sequence_labels_train;sequence_labels_test],...
    'Learners',template,...
    'Coding','onevsone',...
    'ClassNames',{'P_1','P_2','P_3','P_4','P_5','P_6','P_7',
    'P_8','P_9','P_10'});
kfoldmodel = crossval(model_butter_band20350,'KFold',2);
classLabels = kfoldPredict(kfoldmodel);
loss = kfoldLoss(kfoldmodel)*100


%TESTING

ypos=filter(b,a,test_person_8); %APPLY FILTER TO THE TEST SET
test_person_8=ypos(1:44100);
test_person_8= reshape(test_person_8,[3675,12]); %RESHAPE


ypos=filter(b,a,test_person_10);
test_person_10=ypos(1:44100);
test_person_10= reshape(test_person_10,[3675,12]);


ypos=filter(b,a,test_person_5);
test_person_5=ypos(1:44100);
test_person_5= reshape(test_person_5,[3675,12]);



ypos=filter(b,a,test_person_6);
test_person_6=ypos(1:44100);
test_person_6= reshape(test_person_6,[3675,12]);


ypos=filter(b,a,test_person_7);
test_person_7=ypos(1:44100);
```

```
test_person_7= reshape(test_person_7,[3675,12]);



ypos=filter(b,a,test_person_4);
test_person_4=ypos(1:44100);
test_person_4= reshape(test_person_4,[3675,12]);



ypos=filter(b,a,test_person_3);
test_person_3=ypos(1:44100);
test_person_3= reshape(test_person_3,[3675,12]);



ypos=filter(b,a,test_person_2);
test_person_2=ypos(1:44100);
test_person_2= reshape(test_person_2,[3675,12]);




ypos=filter(b,a,test_person_1);
test_person_1=ypos(1:44100);
test_person_1= reshape(test_person_1,[3675,12])



ypos=(filter(b,a,test_person_9));
test_person_9=ypos(1:44100);
test_person_9= reshape(test_person_9,[3675,12])
%%


%% PUT ALL THE TEST SET TOGETHER AND ADD THE CORRESPONDING LABELS
YTest = [YTrainP_1(1:12); YTrainP_2(1:12); YTrainP_3(1:12);
YTrainP_4(1:12); YTrainP_5(1:12);YTrainP_6(1:12);YTrainP_7(1:12);
YTrainP_8(1:12);YTrainP_9(1:12);YTrainP_10(1:12)];

XTest = [test_person_1.';test_person_2.';test_person_3.';
test_person_4.';test_person_5.';test_person_6.';
test_person_7.';test_person_8.';test_person_9.';
test_person_10.'].';


```

```
%% REDUCE DATA BY 2
XTest=XTest(1:2:3674,:);

for a = 1:120
 XTest(:,a)=rescale(XTest(:,a),-1,1);
end



%%FEATURE EXTRACTION

scat_features_test = featureMatrix(sn,XTest,'Transform','log');
scat_features_test = permute(scat_features_test,[2 3 1]);
scat_features_test = reshape(scat_features_test,...
    size(scat_features_test,1)*size(scat_features_test,2),[]);



[sequence_labels_train,sequence_labels_test] = ...
    createSequenceLabels_heartsounds(Nseq,YTrain,YTest); %SINCE THERE ...
        ARE NOW MORE COEFFICIENTS SETS FOR EACH ORIGINAL DATA SET, ...
        THIS PRODUCES THE LABELS FOR THE NEW SETS








predLabels = predict(model_butter_band20350,scat_features_test);

predLabels = categorical(predLabels)



%%TEST

[confmatTest,grouporder] = confusionmat(sequence_labels_test,predLabels)
cm=confusionchart(sequence_labels_test,predLabels)
cm.RowSummary = 'row-normalized';
cm.ColumnSummary = 'column-normalized';




```

```



  for a=1:120
       result(a)= mode(predLabels((4*(a-1)+1):(4*a))) %TAKE THE MODE ...
          BECAUSE WE HAVE 4 TIMES AS COEFFICIENTS SETS AS THE ...
          ORIGINAL NUMBER OF TETSING SETS
    end
classes = categorical({'P_1','P_2','P_3','P_4','P_5','P_6',
'P_7','P_8','P_9','P_10'});
[confmatTest,classes] = confusionmat(YTest,result)
cm=confusionchart(YTest,result)
```

Bibliography

[1] Ishak, W. H. W., Siraj, F. (2002). *The Topol Reviewn*. The Topol Review. Preparing the Healthcare Workforce to Deliver the Digital Future, 1-48.

[2] Coravos, A., Goldsack, J. C., Karlin, D. R., Nebeker, C., Perakslis, E., Zimmerman, N., Erb, M. K. (2019). *Digital medicine: a primer on measurement.* Digital Biomarkers, 3(2), 31-71.

[3] Topol, E. J. (2019). *A decade of digital medicine innovation*. Science translational medicine, 11(498).

[4] Cester, L., Lyons, A., Starshynov, I., Walker, R., Warburton, R., Faccio, D. (2020, June). *Laser vibrometry through a scattering medium with a single-photon camera.* In Imaging Systems and Applications (pp. IW3D-4). Optical Society of America.

[5] Cester, L., Starshynov, I., Jones, Y., Pellicori, P., Faccio, D. (2021, June) *Remote heart sound characterisation and classification using computational imaging.* In 2021 Conference on Lasers and Electro-Optics Europe European Quantum Electronics Conference (CLEO/Europe-EQEC) (pp. 1-1). IEEE.

[6] Sarj, A., (2018). *What is artificial intelligence* https://aisb.org.uk/newsite/?p=73.

[7] El Naqa, I., Murphy, M. J. (2015). *What is machine learning?*. Springer, Cham.

[8] Benjamens, S., Dhunnoo, P., MeskÃş, B. (2020). *The state of artificial intelligence-based FDA-approved medical devices and algorithms: an online database*. NPJ digital medicine, 3(1), 1-8.

[9] Ishak, W. H. W., Siraj, F. (2002). *Artificial intelligence in medical application: An exploration*. Health Informatics Europe Journal, 16.

[10] Thaulow, E., Erikssen, J., Sandvik, L., Erikssen, G., Jorgensen, L., Colin, P. F. (1993). *Initial clinical presentation of cardiac disease in asymptomatic men with silent myocardial ischemia and angiographically documented coronary artery disease (the Oslo Ischemia Study).* The American journal of cardiology, 72(9), 629-633.

[11] Khan, M. A. (2020). *An IoT framework for heart disease prediction based on MDCNN classifier.* IEEE Access, 8, 34717-34727.

[12] Pagidipati, N. J., Gaziano, T. A. (2013). *Estimating deaths from cardiovascular disease: a review of global methodologies of mortality measurement*. Circulation, 127(6), 749-756.

[13] Writing Committee:, Smith Jr, S. C., Collins, A., Ferrari, R., Holmes Jr, D. R., Logstrup, S., ... Zoghbi, W. A. (2012). *Our time: a call to save preventable death from cardiovascular disease (heart disease and stroke).*. European heart journal, 33(23), 2910-2916.

[14] Zeleznik, R., Foldyna, B., Eslami, P., Weiss, J., Alexander, I., Taron, J., ... Aerts, H. J. (2021). *Deep convolutional neural networks to predict cardiovascular risk from computed tomography*. Nature communications, 12(1), 1-9.

[15] De Bacquer, D., De Backer, G., Kornitzer, M., Blackburn, H. (1998). *Prognostic value of ECG findings for total, cardiovascular disease, and coronary heart disease death in men and women*. Heart, 80(6), 570-577.

[16] NHS (2018). *Electrocardiogram (ECG).* https://www.nhs.uk/conditions/electrocardiogram/

[17] Thygesen, K., Alpert, J. S., Jaffe, A. S., Chaitman, B. R., Bax, J. J., Morrow, D. A., ... Executive Group on behalf of the Joint European Society of Cardiology (ESC)/American College of Cardiology (ACC)/American Heart Association (AHA)/World Heart Federation (WHF) Task Force for the Universal Definition of Myocardial Infarction. (2018). *Fourth universal definition of myocardial infarction (2018).* Journal of the American College of Cardiology, 72(18), 2231-2264.

[18] Schultz, J. C., Hilliard, A. A., Cooper Jr, L. T., Rihal, C. S. (2009, November). *Diagnosis and treatment of viral myocarditis* In Mayo Clinic Proceedings (Vol. 84, No. 11, pp. 1001-1009). Elsevier.

[19] Beckerman, J., (2020). *When Do I Need a Chest X-Ray for Heart Disease?.* https://www.webmd.com/heart-disease/guide/diagnosing-chest-x-ray

[20] Ratini, M., (2020). *Echocardiogram.* https://www.webmd.com/heart-disease/guide/diagnosing-echocardiogram

[21] Oh, J. K. (2007). *Echocardiography in heart failure: beyond diagnosis*. European Journal of Echocardiography, 8(1), 4-14.

[22] KalinauskienÄŮ, E., Razvadauskas, H., Morse, D. J., Maxey, G. E., NaudÅ¿iÅńnas, A. (2019). *A comparison of electronic and traditional stethoscopes in the heart auscultation of obese patients.* Medicina, 55(4), 94.

[23] Grenier, M. C., Gagnon, K., Genest, J., Durand, J., Durand, L. G. (1998). *Clinical comparison of acoustic and electronic stethoscopes and design of a new electronic stethoscope.* American Journal of Cardiology, 81(5), 653-656.

[24] Nowak, L. J., Nowak, K. M. (2018). *Sound differences between electronic and acoustic stethoscopes.* Biomedical engineering online, 17(1), 1-11.

[25] Leng, S., San Tan, R., Chai, K. T. C., Wang, C., Ghista, D., Zhong, L. (2015). *The electronic stethoscope.* Biomedical engineering online, 14(1), 1-37.

[26] Adams, J., Apple, F. (2004). *New blood tests for detecting heart disease*. Circulation, 109(3), e12-e14.

[27] NIH. (2020). *Nuclear Heart Scan.* https://www.nhlbi.nih.gov/health-topics/nuclear-heart-scan

[28] Vasanawala, S. S., Hanneman, K., Alley, M. T., Hsiao, A. (2015). *Congenital heart disease assessment with 4D flow MRI.* Journal of Magnetic Resonance Imaging, 42(4), 870-886.

[29] Rangayyan, R. M., Reddy, N. P. (2002). *Biomedical signal analysis: a case-study approach.* Annals of Biomedical Engineering, 30(7), 983-983.

[30] SkanÃĺ'r, Y., Bring, J., Ullman, B., Strender, L. E. (2003). *Heart failure diagnosis in primary health care: clinical characteristics of problematic patients. A clinical judgement analysis study.* BMC Family Practice, 4(1), 1-8.

[31] WHO, (June 2021). *Cardiovascular diseases (CVDs).* BMC Family Practice, 4(1), 1-8. http://www.who.int/mediacentre/factsheets/fs317/en/.

[32] McGee, S. (2021). *Evidence-based physical diagnosis e-book.* Elsevier Health Sciences.

[33] Chowdhury, M. E., Khandakar, A., Alzoubi, K., Mansoor, S., M Tahir, A., Reaz, M. B. I., Al-Emadi, N. (2019). *Real-time smart-digital stethoscope system for heart diseases monitoring.* Sensors, 19(12), 2781.

[34] Crawford, M. H., Education, M. H. (Eds.). (2003). *Current diagnosis treatment in cardiology.* Lange Medical Books/McGraw-Hill.

[35] Sherwood, L. (2015). *Human physiology: from cells to systems*. Cengage learning.

[36] Iaizzo, P. A. (2015). *General features of the cardiovascular system*. Handbook of Cardiac Anatomy, Physiology, and Devices (pp. 3-12). Springer, Cham.

[37] Kumar, V., Abbas, A. K., Aster, J. C. (2017). *Robbins basic pathology e-book*. Elsevier Health Sciences.

[38] Tortora, G. J., Derrickson, B. H. (2018). *Principles of anatomy and physiology*. John Wiley Sons.

[39] Ambler, P. (2005). *AHeart sounds made incredibly easy*. Lippincott Williams Wilkins.

[40] Lehner, R. J., Rangayyan, R. M. (1987). *A three-channel microcomputer system for segmentation and characterization of the phonocardiogram.* IEEE Transactions on Biomedical Engineering, (6), 485-489.

[41] Varghees, V. N., Ramachandran, K. I. (2014). *A novel heart sound activity detection framework for automated heart sound analysis.* Biomedical Signal Processing and Control, 13, 174-188.

[42] Babu, K. A., Ramkumar, B. (2020). *Automatic Recognition of Fundamental Heart Sound Segments From PCG Corrupted With Lung Sounds and Speech.* IEEE Access, 8, 179983-179994.

[43] Son, G. Y., Kwon, S. (2018). *Classification of heart sound signal using multiple features.* Applied Sciences, 8(12), 2344.

[44] Sharma, L. N. (2015, November). *Multiscale analysis of heart sound for segmentation using multiscale Hilbert envelope.* In 2015 13th International Conference on ICT and Knowledge Engineering (ICT Knowledge Engineering 2015) (pp. 33-37). IEEE.

[45] Arnott, P. J., Pfeiffer, G. W., Tavel, M. E. (1984). *Spectral analysis of heart sounds: relationships between some physical characteristics and frequency spectra of first and second heart sounds in normals and hypertensives.* Journal of biomedical engineering, 6(2), 121-128.

[46] Littman tradesmark (2017). *Cardiac Auscultation* https://multimedia.3m.com/mws/media/1372905C
littmannr-stethoscopes-auscultation-posters.pdf

[47] Oldani, P. (2019). *How to use a stethoscope.* https://insidefirstaid.com/diagnosis/how-to-use-a-stethoscope-and-where-to-auscultate

[48] Coviello, J. S. (2013). *Auscultation skills: Breath heart sounds*. ippincott Williams Wilkins.

[49] Mitchell, J. R., Wang, J. J. (2014). *Expanding application of the Wiggers diagram to teach cardiovascular physiology*. Advances in physiology education, 38(2), 170-175.

[50] Clark, V. L., Kruse, J. A. (1990). *Clinical methods: the history, physical, and laboratory examinations*. Jama, 264(21), 2808-2809

[51] Crews, T. L., Pridie, R. B., Benham, R., Leatham, A. (1972). *Auscultatory and phonocardiographic findings in Ebstein's anomaly. Correlation of first heart sound with ultrasonic records of tricuspid valve movement.*. British heart journal, 34(7), 681.

[52] AYGEN, M. M., Braunwald, E. (1962). *The splitting of the second heart sound in normal subjects and in patients with congenital heart disease*. Circulation, 25(2), 328-345.

[53] Walker, H. K., Hall, W. D., Hurst, J. W. (1990). *Clinical methods: the history, physical, and laboratory examinations.*.

[54] Mallat S. (2008). *A wavelet tour of Signal Processing*. Academic Press.

[55] Bernardino A., SantosâĂŞVictor S. (2005). *A Real-Time Gabor Primal Sketch for Visual Attention*. Iberian Conference of Pattern Recognition.

[56] Lancia, L., Tiede, M. (2012). *A survey of methods for the analysis of the temporal evolution of speech articulator trajectories.*. Speech planning and dynamics, 233-271.

[57] Grossmann, A., Morlet, J. (1984). *Decomposition of Hardy functions into square integrable wavelets of constant shape*. SIAM journal on mathematical analysis, 15(4), 723-736.

[58] Mallat, S. (1996). *Wavelets for a vision*. Proceedings of the IEEE, 84(4), 604-614.

[59] Burt, P. J., Adelson, E. H. (1987). *The Laplacian pyramid as a compact image code.*. In Readings in computer vision (pp. 671-679). Morgan Kaufmann.

[60] Mallat, S., Hwang, W. L. (1992). *Singularity detection and processing with wavelets.*. IEEE transactions on information theory, 38(2), 617-643.

[61] Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM, Philadelphia (1992)

[62] Oyallon, E., Zagoruyko, S., Huang, G., Komodakis, N., Lacoste-Julien, S., Blaschko, M., Belilovsky, E. (2018). *Scattering networks for hybrid representation learning*. IEEE transactions on pattern analysis and machine intelligence.

[63] Westermark, P. (2017). *Wavelets, Scattering transforms and Convolutional neural networks: Tools for image processing*.

[64] Camps-Valls, G., Bruzzone, L. (2005). *Kernel-based methods for hyperspectral image classification.*. IEEE Transactions on Geoscience and Remote Sensing, 43(6), 1351-1362.

[65] Grauman, K., Darrell, T. (2005). *The pyramid match kernel: Discriminative classification with sets of image features*. In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 (Vol. 2, pp. 1458-1465). IEEE.

[66] Hofmann, T., SchÃűlkopf, B., Smola, A. J. (2008). *Kernel methods in machine learning*. he annals of statistics, 1171-1220.

[67] LeCun, Y. (1988). *THE MNIST DATABASE of handwritten digits*. Courant Institute, NYU Corinna Cortes, Google Labs, New York Christopher J.C. Burges, Microsoft Research, Redmond.

[68] Lyon, R. (1982, May). *A computational model of filtering, detection, and compression in the cochlea*. In ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 7, pp. 1282-1285). IEEE.

[69] AndÃl'n, J., Mallat, S. (2014). *Deep scattering spectrum*. IEEE Transactions on Signal Processing, 62(16), 4114-4128.

[70] Van der Maaten, L., Hinton, G. (2008). *Visualizing data using t-SNE*. ournal of machine learning research, 9(11).

[71] Weisbuch, C., and Vinter, B. (1991). *Quantum semiconductor structures*. Boston: Academic Press.

[72] Al Bassam, N., Ramachandran, V., Parameswaran, S. E. (2021). *Wavelet Theory and Application in Communication and Signal Processing*. IntechOpen.

[73] Kyriazis, P., Anastasiadis, C., Triantis, D., Vallianatos, F. (2006). *Wavelet analysis on pressure stimulated currents emitted by marble samples*. Natural Hazards and Earth System Sciences, 6(6), 889-894.

[74] Bruna, J., Mallat, S. (2013). *Invariant scattering convolution networks*. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1872-1886.

[75] Cherif, L. H., Debbal, S. M., Bereksi-Reguig, F. (2010). *Choice of the wavelet analyzing in the phonocardiogram signal analysis using the discrete and the packet wavelet transform*. Expert Systems with Applications, 37(2), 913-918

[76] Recording Corporation (2021). *GLOSSARY OF AUDIO, RECORDING AND MUSIC TERMS "*. https://www.recordingconnection.com/glossary/

[77] Boureau, Y. L., Bach, F., LeCun, Y., Ponce, J. (2010, June). *Learning mid-level features for recognition*. 2010 IEEE computer society conference on computer vision and pattern recognition (pp. 2559-2566). IEEE.

[78] LeCun, Y., Kavukcuoglu, K., Farabet, C. (2010, May). *Convolutional networks and applications in vision*. In Proceedings of 2010 IEEE international symposium on circuits and systems (pp. 253-256). IEEE.

[79] Smola, A., Vishwanathan, S. V. N. (2008). *Introduction to machine learning*. Cambridge University, UK, 32(34), 2008.

[80] Datta N. (2019). *Support Vector Machine in Python*. https://medium.com/@dattanaman213/support-vector-machine-in-python-576eaac337ae

[81] Kong, Q., Xu, Y., Wang, W., Plumbley, M. D. (2018, April). *A joint separation-classification model for sound event detection of weakly labelled data*. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 321-325). IEEE.

[82] Chang, Y. H. S., Liao, Y. F., Wang, S. M., Wang, J. H., Wang, S. Y., Chen, J. W., Chen, Y. D. (2017, June). *Development of a large-scale Mandarin radio speech corpus*. 2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW) (pp. 359-360). IEEE.

[83] Bengio, Y., Simard, P., Frasconi, P. (1994). *Learning long-term dependencies with gradient descent is difficult*. IEEE transactions on neural networks, 5(2), 157-166.

[84] Sak, H., Senior, A. W., Beaufays, F. (2014). *Long short-term memory recurrent neural network architectures for large scale acoustic modeling.*.

[85] Jordan, M. I., Mitchell, T. M. (2015). *Machine learning: Trends, perspectives, and prospects.* Science, 349(6245), 255-260.

[86] Wang, Q., Ma, Y., Zhao, K., Tian, Y. (2020). *A comprehensive survey of loss functions in machine learning.* Annals of Data Science, 1-26.

[87] Rebala G., Ravi A., Churiwala S. (2019) *An Introduction to Machine Learning.* Springer, Cham.

[88] Brownlee, J., (2019). *A Gentle Introduction to Probability Density Estimation* https://machinelearningmastery.com/probability-density-estimation/

[89] Greene, D., Cunningham, P., Mayer, R. (2008). *Unsupervised learning and clustering* In Machine learning techniques for multimedia (pp. 51-90). Springer, Berlin, Heidelberg.

[90] Van Der Maaten, L., Postma, E., Van den Herik, J. (2009). *Dimensionality reduction: a comparative.* J Mach Learn Res, 10(66-71), 13.

[91] Yu, Y., Si, X., Hu, H., Zhang, J. (2019). *A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures.* Neural Comput 2019; 31 (7): 1235âĂŞ1270.

[92] Tendolkar, G., (2016) *Machine Learning Notebook* https://sites.google.com/site/machinelearningnotebook2/about-me

[93] Kavukcuoglu, K., Ranzato, M. A., LeCun, Y. (2010). *Fast inference in sparse coding algorithms with applications to object recognition.* arXiv preprint arXiv:1010.3467.

[94] Chen, D., He, Q., Wang, X. (2007). *On linear separability of data sets in feature space.* Neurocomputing, 70(13-15), 2441-2448.

[95] Vapnik, V. (1999). *The nature of statistical learning theory.* Springer science business media.

[96] Rosenblatt, F. (1958). *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychological review, 65(6), 386.

[97] Kim, E., (2017). *Everything You Wanted to Know about the Kernel Trick* http://www.eric-kim.net/eric-kim-net/posts/1/kernel$_t$*rick.html*

[98] Mit, I., (2019). *Introduction to Regression* https://openlearninglibrary.mit.edu/courses/course-v1:MITx+6.036+1T2019/courseware/Week5/regression/

[99] Dietterich, T., (2019). *Machine Learning for Sequential Data: a Review* http://web.engr.oregonstate.edu/ tgd/publications/mlsd-ssspr.pdf

[100] Kadam, S. (2020). *Neural Network Part1: Inside a Single Neuron.* https://medium.com/analytics-vidhya/neural-network-part1-inside-a-single-neuron-fee5e44f1e

[101] Zhang, J. (2019). *Reinforcement Learning âĂŤ Multi-Arm Bandit Implementation.* https://towardsdatascience.com/reinforcement-learning-multi-arm-bandit-implementation-5399ef67b24b

[102] Zaremba, W., Sutskever, I., Vinyals, O. (2014). *RRecurrent neural network regularization.* arXiv preprint arXiv:1409.2329.

[103] simplilearn, (May 2021). *What is Perceptron: A Beginners Guide for Perceptron* https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron

[104] Genesis, (June 2018). *Gradient Descent Part1* https://www.fromthegenesis.com/gradient-descent-part1

[105] Elayi, C. S., Charnigo, R. J., Heron, P. M., Lee, B. K., Olgin, J. E. (2017). *Primary Prevention of Sudden Cardiac Death Early Post-Myocardial Infarction: Root Cause Analysis for Implantable CardioverterâĂŞDefibrillator Failure and Currently Available Options.* Circulation: Arrhythmia and Electrophysiology, 10(6), e005194.

[106] Naghavi, M., Falk, E., Hecht, H. S., Jamieson, M. J., Kaul, S., Berman, D., ... Shah, P. K. (2006). *From vulnerable plaque to vulnerable patientâĂŤpart III: executive summary of the*

Screening for Heart Attack Prevention and Education (SHAPE) Task Force report. The American journal of cardiology, 98(2), 2-15.

[107] Diedler, J., Sykora, M., Juttler, E., Steiner, T., Hacke, W. (2009). *Intensive care management of acute stroke: general management.* International Journal of Stroke, 4(5), 365-378.

[108] Moon, R. Y., Horne, R. S., Hauck, F. R. (2007). *Sudden infant death syndrome.* The Lancet, 370(9598), 1578-1587.

[109] Nascimento, F. A., Tseng, Z. H., Palmiere, C., Maleszewski, J. J., Shiomi, T., McCrillis, A., Devinsky, O. (2017). *Pulmonary and cardiac pathology in sudden unexpected death in epilepsy (SUDEP).* Epilepsy Behavior, 73, 119-125.

[110] Huang, H., Yang, D., Yang, X., Lei, Y., Chen, Y. (2019, March).

[111] In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 691-694). IEEE.

[112] Leng, S., San Tan, R., Chai, K. T. C., Wang, C., Ghista, D., Zhong, L. (2015). *The electronic stethoscope.* Biomedical engineering online, 14(1), 1-37.

[113] Will, C., Shi, K., Schellenberger, S., Steigleder, T., Michler, F., Fuchs, J., ... Koelpin, A. (2018). *Radar-based heart sound detection.* Scientific reports, 8(1), 1-14.

[114] Koelpin, A., Lurz, F., Linz, S., Mann, S., Will, C., Lindner, S. (2016). *Six-port based interferometry for precise radar and sensing applications.* Sensors, 16(10), 1556.

[115] Koegelenberg, S., Scheffer, C., Blanckenberg, M. M., Doubell, A. F. (2014, August). *Application of laser doppler vibrometery for human heart auscultation.* In 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 4479-4482). IEEE.

[116] Smith, W. (1992). *Effect of light on selenium during the passage of an electric current.* SPIE MILESTONE SERIES MS, 56, 3-3.

[117] *"How Microphones Work". Mediacollege.com. Media College.*

[118] Prakash, O., Khare, M., Srivastava, R. K., Khare, A. (2015). *Tracking of deformable object in complex video using steerable pyramid wavelet transform.* In Computational Vision and Robotics (pp. 1-6). Springer, New Delhi.

[119] Hutt, D. L., Snell, K. J., Belanger, P. A. (1993). *Alexander Graham Bell's Photophone.* Optics and Photonics News, 4(6), 20-25.

[120] Bell, A. G. (1880). *The photophone* Science, 1(11), 130-134.

[121] P. Smid, P. Horvath, M. Hrabovsky (2014). *Methods for Determination of Mean Speckle Size in Simulated Speckle Pattern* Science Review, Vol. 14.

[122] Chouinard, J., (2018). *The Fundamentals of Camera and Image Sensor Technology*. https://www.visiononline.org/userassets/aiauploads/file/cvp$_{t}he - fundamentals - of - camera - and - image - sensor - technology_{j}on - chouinard.pdf$

[123] Lucid Sales Staff, (2020). *Understanding The Digital Image Sensor*. https://thinklucid.com/tech-briefs/understanding-digital-image-sensors/

[124] Edmund Optics, (2020). *Understanding Camera Sensors for Machine Vision Applications*. https://www.edmundoptics.co.uk/knowledge-center/application-notes/imaging/understanding-camera-sensors-for-machine-vision-applications/

[125] wavelength electronics (2020). *PHOTODIODE BASICS*. https://www.teamwavelength.com/photodiode-basics/

[126] Vines, P., Kuzmenko, K., Kirdoda, J., Dumas, D. C., Mirza, M. M., Millar, R. W., ... Buller, G. S. (2019). *High performance planar germanium-on-silicon single-photon avalanche diode detectors.* Nature communications, 10(1), 1-9.

[127] Zappa, F., Tisa, S., Tosi, A., Cova, S. (2007). *Principles and features of single-photon avalanche diode arrays*. Sensors and Actuators A: Physical, 140(1), 103-112.

[128] Qu, Y., Wang, T., Zhu, Z. (2010). *Vision-aided laser Doppler vibrometry for remote automatic voice detection*. IEEE/ASME Transactions on Mechatronics, 16(6), 1110-1119.

[129] Avargel, Y., Cohen, I. (2011, May). *Speech measurements using a laser Doppler vibrometer sensor: Application to speech enhancement*. Joint Workshop on Hands-free Speech Communication and Microphone Arrays (pp. 109-114). IEEE.

[130] Danielsson, P. E., Seger, O. (1990). *Rotation invariance in gradient and higher order derivative detectors*. Computer Vision, Graphics, and Image Processing, 49(2), 198-221.

[131] Davis, A., Rubinstein, M., Wadhwa, N., Mysore, G. J., Durand, F., Freeman, W. T. (2014). *The visual microphone: Passive recovery of sound from video.*.

[132] Johansmann, M., Siegmund, G., Pineda, M. (2005). *Targeting the limits of laser Doppler vibrometry*. Proc. IDEMA, 1-12

[133] Hii, A. J. H., Hann, C. E., Chase, J. G., Van Houten, E. E. (2006). *Fast normalized cross correlation for motion tracking using basis functions*. Computer methods and programs in biomedicine, 82(2), 144-156.

[134] Zalevsky, Z., Beiderman, Y., Margalit, I., Gingold, S., Teicher, M., Mico, V., Garcia, J. (2009) *Simultaneous remote extraction of multiple speech sources and heart beats from secondary speckles pattern*. Optics express, 17(24), 21566-21580.

[135] Freeman, W. T., Adelson, E. H. (1991). *The design and use of steerable filters.* IEEE Transactions on Pattern analysis and machine intelligence, 13(9), 891-906.

[136] Horn, B. K., Schunck, B. G. (1981). *Determining optical flow.* Artificial intelligence, 17(1-3), 185-203.

[137] En-Lin, Ch. (2019) *Introduction to Motion Estimation with Optical Flow* Retrieved from: https://nanonets.com/blog/optical-flow/

[138] Zivkovic, S. (2021) *013 Optical Flow Using Horn and Schunck Method* Retrieved from: http://datahacker.rs/013-optical-flow-using-horn-and-schunck-method/

[139] Lucas, B. D., Kanade, T. (1981, April). *An iterative image registration technique with an application to stereo vision*

[140] Steve Seitz. (2019) *Motion and Optic Flow.* CS 4495 Computer Vision âĂŞ A. Bobick.

[141] Wu, N., Haruyama, S. (2020). *Real-time audio detection and regeneration of moving sound source based on optical flow algorithm of laser speckle images.* Optics express, 28(4), 4475-4488.

[142] Shi, J., Tomasi. (1994) *Good features to track.* Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593-600.

[143] Veber, A. A., Lyashedko, A., Sholokhov, E., Trikshev, A., Kurkov, A., Pyrkov, Y., ... Tsvetkov, V. (2011). *Laser vibrometry based on analysis of the speckle pattern from a remote object* Applied Physics-Section B-Lasers and Optics, 105(3), 613.

[144] Chen, Z., Wang, C., Huang, C., Fu, H., Luo, H., Wang, H. (2014). *Audio signal reconstruction based on adaptively selected seed points from laser speckle images*. Optics Communications, 331, 6-13.

[145] Ge Zhu, Xu-Ri Yao, Peng Qiu, Waqas Mahmood, Wen-Kai Yu, Zhi-Bin Sun, Guang-Jie Zhai, Qing Zhao (2018). *Sound recovery via intensity variations of speckle pattern pixels selected with variance-based method*. Optical Engineering, 57(2), 026117

[146] Bianchi, S. (2014) *Vibration detection by observation of speckle pattern*. Applied optics, 53(5), 931-936.

[147] Briers (2000). *Time-varying laser speckle for measuring motion and flow*. Proc. SPIE 4242, Saratov Fall Meeting

[148] Cester, L., Lyons, A., Starshynov, I., Walker, R., Warburton, R., Faccio, D. (2020, June). *Laser vibrometry through a scattering medium with a single-photon camera*. In Imaging Systems and Applications (pp. IW3D-4). Optical Society of America.

[149] Goodman, J., (1968). *Introduction to Fouruer Optics.* W. H. Freeman

[150] Ohtsuki, R., (2013). *Analysis of Skin Surface Roughness by Visual Assessment and Surface Measurement*. OPTICAL REVIEW Vol. 20, No. 2 (2013) 94âĂŞ101

[151] Kostis, J. B., Moreyra, A. E., Amendo, M. T., Di Pietro, J. O. A. N. N. E., Cosgrove, N. O. R. A., Kuo, P. T. (1982). *The effect of age on heart rate in subjects free of heart disease*. Circulation, 65(1), 141-145.

[152] Rangayyan, R. M., Lehner, R. J. (1987). *Phonocardiogram signal analysis: a review.* Critical reviews in biomedical engineering, 15(3), 211-236.

[153] Shi, W. Y., Chiao, J. C. (2018). *Neural network based real-time heart sound monitor using a wireless wearable wrist sensor.* Analog Integrated Circuits and Signal Processing, 94(3), 383-393.

[154] Oh, S. L., Jahmunah, V., Ooi, C. P., Tan, R. S., Ciaccio, E. J., Yamakawa, T., ... Acharya, U. R. (2020). *Classification of heart sound signals using a novel deep WaveNet model.* Computer Methods and Programs in Biomedicine, 196, 105604.

[155] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... Stanley, H. E. (2000). *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.* Circulation [Online]. 101 (23), pp. e215âĂŞe220.

[156] Liang, H., Lukkarinen, S., Hartimo, I. (1997, September). *Heart sound segmentation algorithm based on heart sound envelogram.* In Computers in Cardiology 1997 (pp. 105-108). IEEE.

[157] Li, S., Li, F., Tang, S., Xiong, W. (2020). *A review of computer-aided heart sound detection techniques.* BioMed research international, 2020.

[158] Salman, A. H., Ahmadi, N., Mengko, R., Langi, A. Z., Mengko, T. L. (2015, November). *Performance comparison of denoising methods for heart sound signal.* In 2015 international symposium on intelligent signal processing and communication systems (ISPACS) (pp. 435-440). IEEE.

[159] Hall, L. T., Maple, J. L., Agzarian, J., Abbott, D. (2000). *Sensor system for heart sound biomonitor* Microelectronics Journal, 31(7), 583-592.

[160] Cherif, L. H., Debbal, S. M., Bereksi-Reguig, F. (2010). *Choice of the wavelet analyzing in the phonocardiogram signal analysis using the discrete and the packet wavelet transform.* Expert Systems with Applications, 37(2), 913-918.

[161] Rudin, L. I., Osher, S., Fatemi, E. (1992). *Nonlinear total variation based noise removal algorithms* Physica D: nonlinear phenomena, 60(1-4), 259-268.

[162] Karahanoglu, F. I., Bayram, I., Van De Ville, D. (2011). *Nonlinear total variation based noise removal algorithms.* IEEE Transactions on Signal Processing, 59(11), 5265-5274.

[163] Varghees, V. N., Ramachandran, K. I. (2014). *A novel heart sound activity detection framework for automated heart sound analysis.* Biomedical Signal Processing and Control, 13, 174-188.

[164] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., ... Liu, H. H. (1998). *The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis.* Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 454(1971), 903-995.

[165] Potdar, M. R., Meshram, M., Dewangan, N., Kumar, R. (2015). *Implementation of adaptive algorithm for PCG signal denoising.* system, 3(4).

[166] Barbosh, M., Singh, P., Sadhu, A. (2020). *Empirical mode decomposition and its variants: a review with applications in structural health monitoring.* Smart Materials and Structures, 29(9), 093001.

[167] Vogel, C. R., Oman, M. E. (1996). *Iterative methods for total variation denoising.* SIAM Journal on Scientific Computing, 17(1), 227-238.

[168] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., ... Liu, H. H. (1998). *The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis.* Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 454(1971), 903-995.

[169] Molau, S., Pitz, M., Schluter, R., Ney, H. (2001, May). *EComputing mel-frequency cepstral coefficients on the power spectrum.* Proceedings (cat. No. 01CH37221) (Vol. 1, pp. 73-76). IEEE.

[170] Nnamoko, N., Arshad, F., England, D., Vora, J., Norman, J. (2014). *Evaluation of filter and wrapper methods for feature selection in supervised machine learning.* Age, 21(81), 33-2.

[171] Huang, S. H. (2015). *Supervised feature selection: A tutorial.* Res., 4(2), 22-37.

[172] Jha, S. K., Yadava, R. D. S. (2010). *Denoising by singular value decomposition and its application to electronic nose data processing.* IEEE Sensors Journal, 11(1), 35-44.

[173] Shorten, C., Khoshgoftaar, T. M. (2019). *A survey on image data augmentation for deep learning.* Journal of big data, 6(1), 1-48.

[174] Brownlee, J., (January 2010). *Random Oversampling and Undersampling for Imbalanced Classification.* https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

[175] Beritelli, F., Serrano, S. (2007). *Biometric identification based on frequency analysis of cardiac sounds.*. IEEE Trans. Inf. Forensics

[176] Khan, M. U., Aziz, S., Zainab, A., Tanveer, H., Iqtidar, K., Waseem, A. (2020, June). *Biometric system using PCG signal analysis: a new method of person identification*. In 2020 international conference on electrical, communication, and computer engineering (icecce) (pp. 1-6). IEEE.

[177] Bhardwaj, I., Londhe, N. D., Kopparapu, S. K. (2017). *A novel behavioural biometric technique for robust user authentication.* IETE Technical Review, 34(5), 478-490.

[178] Yusuf, N., Marafa, K. A., Shehu, K. L., Mamman, H., Maidawa, M. (2020). *A survey of biometric approaches of authentication.* International Journal of Advanced Computer Research, 10(47), 96-104.

[179] Bhattacharyya, D., Ranjan, R., Alisherov, F., Choi, M. (2009). *Biometric authentication: A review.* International Journal of u-and e-Service, Science and Technology, 2(3), 13-28.

[180] Guidino, M., (2018). *How Do Fingerprint Scanners Work? Optical vs Capacitive* https://www.arrow.com/en/research-and-events/articles/how-fingerprint-sensors-work

[181] Qiu, L. (2014, June). *Fingerprint sensor technology*. In 2014 9th IEEE Conference on Industrial Electronics and Applications (pp. 1433-1436). IEEE.

[182] Dantcheva, A., Chen, C., Ross, A. (2012, September). *Can facial cosmetics affect the matching accuracy of face recognition systems?*. In 2012 IEEE Fifth international conference on biometrics: theory, applications and systems (BTAS) (pp. 391-398). IEEE.

[183] Taigman, Y., Yang, M., Ranzato, M. A., Wolf, L. (2014). *Deepface: Closing the gap to human-level performance in face verification.* In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).

[184] Wildes, R. P. (1997). *Iris recognition: an emerging biometric technology.* Proceedings of the IEEE, 85(9), 1348-1363.

[185] Daugman, J. G. (1988). *Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression.* IEEE Transactions on acoustics, speech, and signal processing, 36(7), 1169-1179.

[186] Faundez-Zanuy, M., Elizondo, D. A., Ferrer-Ballester, M. ÃĄ., Travieso-GonzÃąlez, C. M. (2007). *Authentication of individuals using hand geometry biometrics: A neural network approach.* Neural Processing Letters, 26(3), 201-216.

[187] MariÃśo, C., Penedo, M. G., Penas, M., Carreira, M. J., Gonzalez, F. (2006). *Personal authentication using digital retinal images.* Pattern Analysis and Applications, 9(1), 21-33.

[188] Mazumdar, J. B., Nirmala, S. R. (2018).

[189] International Journal of Advanced Research in Computer Science, 9(1).

[190] Annapurani, K., Sadiq, M. A. K., Malathy, C. (2015). *Fusion of shape of the ear and tragusâĂŞa unique feature extraction method for ear authentication system.* Expert Systems with Applications, 42(1), 649-656.

[191] Bai, L., Shen, L. (2003, February). *Face detection by orientation map matching.* In International Conference on Computational Intelligence for Modelling Control and Automation, Austria.

[192] Khan, M. U., Aziz, S., Zainab, A., Tanveer, H., Iqtidar, K., Waseem, A. (2020, June). *Biometric system using PCG signal analysis: a new method of person identification.* In 2020 international conference on electrical, communication, and computer engineering (icecce) (pp. 1-6). IEEE.

[193] ZÃžquete, A., Quintela, B., da Silva Cunha, J. P. (2010, January). *Biometric Authentication using Brain Responses to Visual Stimuli.* In Biosignals (pp. 103-112).

[194] El-Dahshan, E. S. A., Bassiouni, M. M., Sharvia, S., Salem, A. B. M. (2021). *PCG signals for biometric authentication systems: An in-depth review.* Computer Science Review, 41, 100420.

[195] electronicnotes, (2021). *What is an Elliptic / Cauer Filter: the basics* https://www.electronics-notes.com/articles/radio/rf-filters/what-is-elliptical-cauer-filter-basics.php

[196] Laghari, W. M., Baloch, M. U., Mengal, M. A., Shah, S. J. (2014). *Performance analysis of analog butterworth low pass filter as compared to Chebyshev type-I filter, Chebyshev type-II filter and elliptical filter.* Circuits and Systems, 2014.

[197] Cester, L., Starshynov, I., Jones, Y., Pellicori, P., Cleland, J., Faccio, D. (April 2022). *Remote laser-speckle sensing of heart sounds for health assessment and biometric identification.* https://doi.org/10.1364/BOE.451416

[198] Cherif, L. H., Debbal, S. M., Bereksi-Reguig, F. (2010). *Choice of the wavelet analyzing in the phonocardiogram signal analysis using the discrete and the packet wavelet transform.* Expert Systems with Applications, 37(2), 913-918.

[199] Topol, E. J. (2014). *Individualized medicine from prewomb to tomb* Cell, 157(1), 241-253.

List of Publications

Cester, L., Lyons, A., Braidotti, M. C., Faccio, D. (2019). Time-of-Flight Imaging at 10 ps Resolution with an ICCD Camera. Sensors, 19(1), 180.

Cester, L., Lyons, A., Starshynov, I., Walker, R., Warburton, R., Faccio, D. (2020, June). Laser vibrometry through a scattering medium with a single-photon camera. In Imaging Systems and Applications (pp. IW3D-4). Optical Society of America.

Cester, L., Starshynov, I., Jones, Y., Pellicori, P., Faccio, D. (2021, June). Remote heart sound characterisation and classification using computational imaging. In 2021 Conference on Lasers and Electro-Optics Europe European Quantum Electronics Conference (CLEO/Europe-EQEC) (pp. 1-1). IEEE.

Cester, L., Starshynov, I., Jones, Y., Pellicori, P., Cleland, J., Faccio, D. (April 2022). Remote laser-speckle sensing of heart sounds for health assessment and biometric identification. https://doi.org/10.1364/BOE.451416