UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Desarrollo y evaluación de algoritmos para la resolución del problema inverso de la electrocardiografía mediante técnicas de aprendizaje profundo

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

AUTOR/A: Strazzeri Muñiz, Sabrina

Tutor/a: Guillem Sánchez, María Salud

Cotutor/a externo: MOLERO ALABAU, RUBEN

Director/a Experimental: ESCALANTE FERNANDEZ, JOSE MARIA

CURSO ACADÉMICO: 2022/2023

Acknowledgements

I would like to thank María for allowing me to be part of COR Group and for being an incredible professor and tutor. Also, thank you to my co-tutors, José María and Rubén, for helping me out and giving me valuable pieces of advice.

Thank you Andrea, Ismael and Raúl for helping me whenever I had doubts. And to the rest, thank you for making me laugh so much and have such a good time while working.

I want to specially thank my family for giving me all the love, courage and motivation I needed while studying the degree. You have always believed and supported my dreams, and I will always be grateful for that.

Last, but not least, I want to thank my fifteen year old self. You sure did not know how difficult everything was going to be, but your fearlessness and audacity brought me here, to the other side of the world, to achieve my dreams. Thank you for not giving up and for facing every challenge that came up. All the efforts were worth it.

Abstract

Cardiac arrhythmias are a condition suffered by a big part of the population; between the 1.5% and 5% of it. Thus, during the last few years, medical diagnosis methods have been improved with the purpose of preventing possible consequent cardiac pathologies or avoiding hasty surgeries by offering a previous optimal treatment.

Electrocardiographic imaging (ECGi) is a non-invasive method which can obtain electric cardiac propagation on the surface of the heart with the registered potentials on the body surface. Since this problem is an inverse one, its ill-posedness could be corrected by the use of regularization methods, such as Tikhonov regularization. However, this cannot guarantee a perfect reconstruction of the cardiac potentials.

Considering the previous statement, this thesis (TFG) aims to develop and evaluate different deep learning algorithms to improve the current Tikhonov inverse solution. In order to do that, as a proof of concept, a database of 16.000 aleatory ellipsoids, each of which will correspond to the heart and body surfaces, is generated; along with dipoles that will act as the electric potential. From this analytically generated database, the inverse problem will be calculated. Later, it will be used to train the proposed neural network architectures, to obtain the predictions from a 1.000 test dataset, with the goal of evidencing that the new solution has enhanced the Tikhonov one.

A reduction of the 46.72% of the mean absolute error and an increment of the 15.5% of the correlation has been shown in the obtained predictions compared to the regularization of Tikhonov. Leading to the conclusion that the use of simple neural networks on images acquired from the inverse problem can potentially increase their similarity to the original ones. Allowing this to act as a possible post-processing technique for any image that comes from an inverse problem resolution (resting importance to the regularization method used), precisely of electrocardiographic imaging.

Key words: Deep Learning, Neural networks, Denoising, Inverse Problem, Cardiology, Arrhythmias, ECGi.

Resumen

Las arritmias son una condición padecida por gran parte de la población, entre el 1.5% y el 5% de ella. Por ello, durante los últimos años, se han mejorado los métodos de diagnóstico médico, con el propósito de prevenir posibles patologías cardíacas consecuentes o para evitar cirugías apresuradas al poder ofrecer un tratamiento previo óptimo.

El problema inverso de la electrocardiografía (ECGi) es un método no-invasivo que permite obtener la propagación eléctrica cardíaca en la superficie del corazón a través de los potenciales registrados en la superficie del torso del paciente. Debido a que se trata de un problema inverso, su característica de mal condicionamiento puede intentar corregirse a través del uso de métodos de regularización, como la regularización de Tikhonov. Sin embargo, esto no siempre garantiza una perfecta reconstrucción de los potenciales cardíacos.

Considerando lo anterior, el propósito de este Trabajo de Fin de Grado (TFG) es el desarrollo y evaluación de diferentes algoritmos de aprendizaje profundo para mejorar la actual solución inversa de Tikhonov. Para lograrlo, y tratándose de una prueba de concepto, una base de datos de 16.000 elipsoides aleatorios, que representarán a la superficie del corazón y del cuerpo, es creada; así

como dipolos que actuarán como el potencial eléctrico. A partir de esta base de datos analítica, el problema inverso será calculado. Posteriormente, será utilizada para entrenar las arquitecturas de redes neuronales propuestas, para obtener las predicciones de un grupo de 1.000 casos de prueba, con el fin de evidenciar que la nueva solución es mejor que la de Tikhonov.

Se ha obtenido una reducción del 46.72% del error medio absoluto y un incremento del 15.5% de la correlación en las predicciones obtenidas de la solución propuesta, en comparación con las de la regularización de Tikhonov. Demostrando, así, que el uso de redes neuronales simples, sobre las imágenes adquiridas del problema inverso, puede aumentar su similitud con respecto a las originales. Sirviendo así, como una posible técnica de post-procesado para cualquier imagen proveniente de la solución de un problema inverso (sin importar su método de regularización), precisamente el del ECGi.

Palabras clave: Aprendizaje profundo, Redes neuronales, Eliminación de ruido, Problema inverso, Cardiología, Arritmias, ECGi.

Resum

Les arítmies cardíaques són una condició que pateix una gran part de la població; entre l'1,5% i el 5%. Així, durant els darrers anys, s'han millorat els mètodes de diagnòstic mèdic amb la finalitat de prevenir possibles patologies cardíaques conseqüents o evitar intervencions quirúrgiques precipitades oferint un tractament previ òptim.

La imatge electrocardiogràfica (ECGi) és un mètode no invasiu que pot obtenir la propagació cardíaca elèctrica a la superfície del cor amb els potencials registrats a la superfície corporal. L'ECGi és un problema invers i conseqüentment mal condicionat, ha de ser corregit mitjançant l'ús de mètodes de regularització, com ara la regularització de Tikhonov. Tanmateix, això no pot garantir una perfecta reconstrucció dels potencials cardíacs.

Tenint en compte l'enunciat anterior, aquesta treball final de grau (TFG) pretén desenvolupar i avaluar diferents algoritmes d'aprenentatge profund per millorar la solució inversa actual de Tikhonov. Per a això, com a prova de concepte, hem generat una base de dades de 16.000 el·lipsoides aleatoris, cadascun dels quals correspondrà a les superfícies del cor i del tors; juntament amb dipols que actuaran com a potencial elèctric. A partir d'aquesta base de dades generada analíticament, es calcularà el problema invers. Posteriorment, s'utilitzarà per entrenar les arquitectures de xarxes neuronals proposades, per obtenir les prediccions a partir d'un conjunt de dades de prova de 1.000 dades, amb l'objectiu d'evidenciar que la nova solució ha millorat la de Tikhonov.

Les xarxes proposades han mostrat una reducció del 46,72% de l'error absolut mitjà i un increment del 15,5% de la correlació en les prediccions obtingudes en comparació amb la regularització de Tikhonov. El que porta a la conclusió que l'ús de xarxes neuronals simples en imatges adquirides a partir del problema invers pot augmentar potencialment la seva similitud amb les originals. Permetent que aquesta actuï com una possible tècnica de postprocessament per a qualsevol imatge que provinga d'una resolució inversa del problema (independent al mètode de regularització utilitzat), precisament de la imatge electrocardiogràfica.

Paraules clau: Aprenentatge Profund, Xarxes neuronals, Eliminació de Soroll, Problema invers, Cardiologia, Arrítmies, ECGi.

To my family and all the girls out there with the dream of becoming an engineer.

Contents

I Project report

1 Introduction **1**
- 1.1 Motivation and Context . . . 1
- 1.2 Objectives . . . 2
- 1.3 Thesis structure . . . 2

2 Literature Review **5**
- 2.1 The Inverse Problem . . . 5
  - 2.1.1 Definition of an ill-posed problem . . . 7
  - 2.1.2 Regularization methods . . . 7
    - 2.1.2.1 Singular Value Decomposition . . . 7
    - 2.1.2.2 Tikhonov regularization . . . 8
- 2.2 Electrophysiological methods of study . . . 8
  - 2.2.1 Fundamentals of electrophysiology . . . 9
  - 2.2.2 The electrocardiogram as a method to detect arrhythmias . . . 12
  - 2.2.3 Electrocardiographic Imaging . . . 12
    - 2.2.3.1 The inverse problem of electrocardiography . . . 13
- 2.3 Fundamentals of Artificial Intelligence . . . 14
- 2.4 Deep Learning . . . 15
  - 2.4.1 Neural Networks . . . 15
    - 2.4.1.1 Processing units . . . 15
    - 2.4.1.2 Connections . . . 16
    - 2.4.1.3 Activation functions . . . 17
- 2.5 Computer vision . . . 18
  - 2.5.1 Convolutional Neural Networks . . . 19
    - 2.5.1.1 Pooling layers . . . 19
    - 2.5.1.2 Convolutional layers . . . 20
  - 2.5.2 Data processing . . . 20
  - 2.5.3 Training hyperparameters . . . 21
  - 2.5.4 Output analysis . . . 21
    - 2.5.4.1 Underfitting . . . 21
    - 2.5.4.2 Overfitting . . . 22
    - 2.5.4.3 Good fit . . . 23
  - 2.5.5 Image denoising . . . 24
    - 2.5.5.1 The autoencoder model . . . 25

3 Methodology **27**

3.1 Generation of the ellipsoids . . . 27
3.1.1 Database creation . . . 30
3.2 Architectures . . . 30
3.2.1 First architecture . . . 30
3.2.2 Second architecture . . . 31
3.2.3 Third architecture . . . 32
3.3 Data pre-processing and definition of the hyperparameters . . . 33

4 Results **35**
4.1 Learning curves . . . 35
4.2 Evaluation of the mean absolute errors and correlation . . . 37
4.3 Representation of the predictions . . . 40

5 Discussion **45**

6 Conclusions and future work **47**
6.1 Conclusions . . . 47
6.2 Future work . . . 48

References **49**

II Appendices

A Detailed structure of the network **57**

List of Figures

2.1 Forward and inverse problem. . . . 6
2.2 Total number of deaths with causes in Spain (left) and Comunitat Valenciana (right) during 2021. Extracted from: (Instituto Nacional de Estadística, 2021). . . . 9
2.3 Cross-sectional view of the heart normal direction of blood flow. Extracted from: (Gupta et. al, 2022). . . . 10
2.4 ECG electrodes. Extracted from: (Shutterstock, n.d.). . . . 11
2.5 P, QRS, T waves, PR interval, and QT interval. Extracted from: (Madona et. al, 2021). . . . 11
2.6 Workflow of electrocardiographic imaging technique. Extracted from: (Hernández 2019). . . . 13
2.7 A. 252 electrode vest, B. heart-torso geometry (CT scan), C. acquisition of the cardiac signals (forward problem), D. pre or per-procedural instantaneous maps (inverse problem). Extracted from: (Shah et. al, 2013). . . . 13
2.8 Processing units of neural networks. Extracted from: (Bhardwaj, 2022). . . . 16
2.9 Connections of neural networks. . . . 16
2.10 A simple neural network with with its weight, activation function, and output. Extracted from: (Debnath, 2021). . . . 17
2.11 Different activation functions and their graphs. Extracted from: (Tanner, n.d.). . . . 18
2.12 Example of low and high level features in images of faces. Extracted from: (Nanos, 2023). . . . 19
2.13 Example of convolutional operation. Extracted from: (Potrimba, 2023). . . . 20
2.14 Example of underfitting. Extracted from: (Brownlee, 2019). . . . 22
2.15 Example of overfitting. Extracted from: (Brownlee, 2019). . . . 23
2.16 Example of a good fit. Extracted from: (Brownlee, 2019). . . . 24
2.17 Image denoising. Extracted from: (Quick Image Processing Research Guide, 2020). 25
2.18 Structure of an autoencoder. Extracted from: (Flores, n.d.). . . . 26

3.1 Analogy between the ECGi and the proposed thesis. . . . 28
3.2 Workflow of the proposed proof-of-concept. . . . 29
3.3 Remeshing process using spherical parametrizations. Extracted from: (Hernández 2019). . . . 29
3.4 Example of the remeshing process for a generated ellipsoid. . . . 29
3.5 Schematic of the first architecture. . . . 31
3.6 Schematic of the second architecture. . . . 32
3.7 Schematic of the third architecture. . . . 33

4.1 Comparison between the learning curves of the loss function for the first (left), second (center) and third distribution (right). . . . 36

4.2 Comparison between the learning curves of the MAE for the first (left), second (center) and third distribution (right). 37
4.3 Comparison of the violin plots representing the correlation and MAE for Tikhonov and each architecture proposed. 39
4.4 Comparison between analytic, Tikhonov and predicted 3D ellipsoids from the selected best case. 41
4.5 Comparison between analytic, Tikhonov and predicted 3D ellipsoids from the selected average case. 41
4.6 Comparison between analytic, Tikhonov and predicted 3D ellipsoids from the selected worst case. 42
4.7 Comparison between analytic, Tikhonov and predicted 2D images for three different cases. 43

A.1 First part of the architecture. 58
A.2 Second part of the architecture. 59

List of Tables

4.1 Comparison of the obtained MAE and correlation values for the first, second and third aleatory distribution of the training datasets. 38
4.2 Comparison of the obtained MAE and correlation values for the three selected predictions. 40

Acronyms

2D Two-Dimensional.

3D Three-Dimensional.

AI Artificial Intelligence.

ANN Artificial Neural Network.

BSPM Body Surface Potential Mapping.

CNN Convolutional Neural Network.

DL Deep Learning.

ECG Electrocardiogram.

ECGi Electrocardiographic Imaging.

EGM Electrogram.

GAI General Artificial Intelligence.

INE Instituto Nacional de Estadística.

MAE Mean Absolute Error.

ML Machine Learning.

MSE Mean Squared Error.

NAI Narrow Artificial Intelligence.

NN Neural Network.

ReLU Rectified Linear Unit.

RGB Red, Green and Blue color model.

SVD Singular Value Decomposition.

TFG Trabajo de Fin de Grado.

Part I

Project report

Chapter 1

Introduction

1.1 Motivation and Context

Cardiovascular diseases (CVDs) are an important matter that needs to be studied in a very detailed way, considering they are, as Mc Namara et. al (2019) explains: "a leading cause of mortality internationally. When combined, ischemic heart disease and all forms of stroke were the attributed causes of death for an estimated 13 million people globally in 2010" [1]. This is why it is a major priority to increase the effectiveness rate when diagnosing and treating CVDs and arrhythmias.

Healthcare technological devices had shown their robustness by allowing the medical professionals detect CVDs sooner. Consequently, more instruments are not only needed, but a lot of new improvements in the current methods they use. Electrocardiography is used as a non-invasive method to detect symptoms of cardiac pathologies, such as arrhythmias, which are caused by abnormal electrical activity [2].

However, the analysis of electrocardiographic signals obtained is still a complicated task for the specialists, due to its complex nature, as Mc Namara et. al continue to explain. Thus, in certain situations, the ECG could not be enough to detect cardiac pathologies. In fact, this has awoken the interest of many researchers in the biomedical field, leading to the subsequent development of the electrocardiographic imaging (ECGi), which is based on the mathematical inverse problem. This formulation allows ECGi to reconstruct the electrical sources that come from the heart by the signals obtained from the ECG, making the study a more understandable one.

Recent improvements have put new diagnosis methods in the spotlight, especially for researchers that are interested in expanding the biomedical field by increasing the use of more innovative technologies. Furthermore, considering how fast the use of Artificial Intelligence (AI) has been increasing during the last decade, it has also turned into the latest approach to solve the ECGi. Due to this, the detection of cardiac diseases and the experiences of the patients will be enhanced [3]. ECGi signals are traditionally obtained by the Tikhonov regularization, which can be further improved by achieving a better noise reduction and reconstruction of images; something that could be evaluated with the use of AI.

One of the sub-fields of AI has been mostly used for image processing applications: Deep Learning (DL) [4]. One of the most common implementations is image denoising, which focuses on the use of algorithms to learn the principal features of the desired output; so, it can reconstruct a given

noisy image later, finally getting a denoised version of it.

Denoising algorithms can be immensely useful for the study of ECGi, considering how it could make the image, obtained after applying the inverse problem, a more accurate one. Moreover, this could potentially increase the significance of a medical diagnosis, leading to earlier prevention and treatment. Of course, the constant upgrades in DL techniques had granted the research field a lot of possibilities to improve the standard ECGi, motivated by the possibility of making a representative change for the good in the quality of life for many people in the world.

1.2 Objectives

The present undergraduate thesis project (TFG) is a proof-of-concept with the principal objective of the development and evaluation of algorithms to solve the inverse problem of electrocardiography with deep learning techniques improving the current methods for obtaining ECGi signals. Therefore, the images obtained, after applying the Tikhonov regularization method, will be considered as noisy images. Later, the problem will be approached as a denoising one, to reconstruct the images and increase the similarity to the original ones.

Thus, the specific objectives are:

- Review of bibliography related to the inverse problem of electrocardiography, the current improvements of deep learning and how denoising techniques have been applied to images.
- Creation of a database with the analytic and Tikhonov images, with 16.000 cases for each one of them, making sure that they are diverse enough to offer the neural network the variety it needs to improve its generalization ability.
- Establishment of three aleatory training dataset distributions to evaluate the data dependency of the models.
- Study of different neural networks topologies for denoising applications to start training the neural network and, finally, choose the one that fits the problem the most.
- Analysis of the obtained results by evaluating the learning curves and different metrics, such as the mean absolute error and the correlation between the Tikhonov and analytic, and the predicted and analytic images.

1.3 Thesis structure

The thesis is structured as follows:

- In Chapter 1, the introduction, motivation, objectives, and the thesis structure are indicated.
- In Chapter 2, the literature review, which funds the rest of the thesis, is explained.
- In Chapter 3, the implemented methodology to create the database, and the architectures used is described.

- In Chapter 4, the results obtained with different architectural models are shown.
- In Chapter 5, the discussion is described.
- In Chapter 6, the conclusions and future work are presented.

Chapter 2

Literature Review

This chapter aims to explore the relevant literature regarding the formulation of the mathematical inverse problem, the application of electrophysiological methods in cardiovascular research, and the fundamental principles of artificial intelligence, deep learning, and computer vision.

The chapter is structured as follows:

- In Section 2.1, the mathematical inverse problem and the most used regularization methods are explained.
- In Section 2.2, the used electrophysiological methods of study are presented.
- In Section 2.3, the fundamentals of artificial intelligence are described.
- In Section 2.4, the sub-field of artificial intelligence: deep learning, and the most important definitions of neural networks are explained.
- In Section 2.5, the definition of computer vision for image processing applications is described.

2.1 The Inverse Problem

Focusing on a mathematical and scientific point of view, a problem is considered an inverse of another one, which will be called forward, when its solution depends on the results obtained by the latter. In other words, the forward problem "consists in computing the consequences of given causes then, the corresponding inverse problem is associated with the reversal of the cause-effect sequence and consists in finding the unknown causes of known consequences" [5].

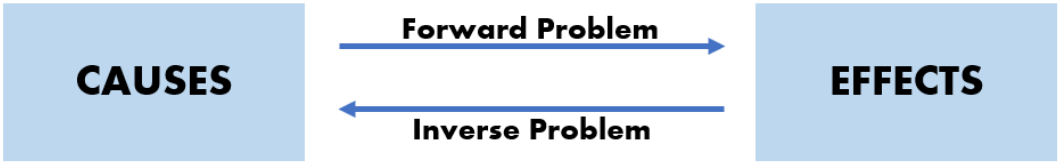

Figure 2.1: Forward and inverse problem.

A better way to understand the concept of the inverse problem is by using common applications as examples [6, 7, 8]:

- During CAT scans, X-rays will hit the detector, after passing through the patient. If every ray is measured in different directions and angles, then it is possible to reconstruct and estimate the dimension of the object studied.
- In geology, the seismic inversion is an illustration, which frequently entails generating a source wavefield at the surface of the earth and gathering the scattered data at receivers from various locations along it.
- Denoising images consists of smoothing them by removing as much noise as possible. This problem has an inverse nature as well because it is trying to get the original image from a noisy one.
- The electrocardiographic imaging, which consists of getting the potentials recordings and geometric information about the heart and body surfaces, to reconstruct the electrical activity of the heart without using invasive methods.

It is well known that one of these problems has been usually studied more (the forward problem) and, as a result, it makes it even harder to obtain accurate values from the other one (the inverse problem) considering the irrefutable uncertainty it contains. Nevertheless, the simplest formulation of the problem is, as it follows (2.1):

$$y = A \cdot x + n, \tag{2.1}$$

where y and x are both vectors that contain the physical measurements and the variables of interest to recover respectively. Considering a non-noisy scenario, the forward operator that associates y and x, in the form of a matrix is A. To consider the noise presence, n is added to model the measurement one.

If the equation was solely the first part of both sides, meaning there is no noise added, it would appear the “inverse crime”, which could end up offering an unrecognizable bad solution. In fact, when inverse problems are evaluated, synthetic data is frequently used to compute the predicted and estimated values, finally leading to too optimistic results [9].

2.1.1 Definition of an ill-posed problem

One of the typical characteristics an inverse problem has is ill-posedness, which consists of the impossibility of being solved by traditional methods without previously establishing an adequate mathematical setting. This means that, even if the forward problem is well-conditioned, the variables of interest, i.e., the desired results, cannot be recovered with a simple approach.

To determine whether a problem can be, or not, ill-posed, it is necessary to meet the Hadamard criteria, which defines three conditions that need to be met to not consider a problem an ill-posed one:

- *Existence*. A solution must exist, at least, to a few data points.
- *Uniqueness*. There should be an only cause that explains the data, so it can be a unique solution.
- *Stability*. The solution should depend continuously on the data, without it being totally altered if it ultimately changes. This leads the output to be more robust, allowing it to be capable of leading with noise.

2.1.2 Regularization methods

An approach to solve the ill-posed condition of the inverse problem is incorporating a regularization method. A regularization term is introduced, as additional information, to provide a more stable result by constraining it.

Although many regularization methods exist, two of them will be explained:

2.1.2.1 Singular Value Decomposition

Considering A as the matrix that will be inverted, also called the transfer matrix, the Singular Value Decomposition (SVD) [10] of it is:

$$A = U\Sigma V^T = \sum_{i=1}^{r} \overrightarrow{u_i}\sigma_i \overrightarrow{v_i}^T, \tag{2.2}$$

where both U and V are matrices, with mxm and nxn order respectively, whose contents are the left and right eigenvectors of A. Σ is the diagonal matrix that defines the eigenvalues of A and r represents the minimum value between the orders of the matrices ($\min(m, n)$).

The pseudoinverse of the transfer matrix (2.3) can be used to calculate the least squares solution of (2.4), as follows:

$$A^{\dagger} = \sum_{i=1}^{rank(A)} \frac{\overrightarrow{u_i}^T \overrightarrow{v_i}^T}{\sigma_i}. \tag{2.3}$$

$$\overrightarrow{x_{LS}} = A^{\dagger}\vec{y} = \sum_{i=1}^{rank(A)} \frac{\overrightarrow{u_i}^T \overrightarrow{v_i}^T \vec{y}}{\sigma_i}. \tag{2.4}$$

In this case, $\vec{u_i}$ contains the vectors with the basis from the signal space, and $\vec{v_i}$ represents the source space.

Solving the least squares solution of the SVD could increase the high frequency components, leading to an unstable solution when the measurement is noisy (which will normally be, considering the complexity of the measurement process). Then, the Tikhonov regularization is introduces as an alternative that could avoid this problem, by introducing a regularization parameter.

2.1.2.2 Tikhonov regularization

The regularization method postulated by Tikhonov is mostly used in the context of electrocardiographic imaging, and its accuracy directly depends on the correct selection of the regularization parameter [11]. The method is modelled by the following equation (2.5),

$$x_{\lambda} = \min(||Ax - y||_2^2 + \lambda^2 \cdot ||Rx||_2^2), \tag{2.5}$$

where new terms are introduced: λ is the regularization parameter, that quantifies the differences between the variables of interest and the measurements, and R is the regularizing operator, which controls the simplicity of the solution. Based on the election of R, it will be referring to zero, first or second-order Tikhonov regularization, that will directly impact the result and it can be chosen after considering the desired results.

A way to choose the regularization parameter is the L-curve method, which is one of the most powerful methods to determine an adequate value for the regularization parameter [12]. Then, considering the equation in (2.6), the optimal lambda would be the value obtained at the maximal curvature of the L-curve.

$$||Rx||_2 = f(||Axy||_2^2). \tag{2.6}$$

2.2 Electrophysiological methods of study

Ischemic heart diseases and cardiac insufficiency are part of the principal causes of death in Spain, according to the 2021 statistics offered by the Instituto Nacional de Estadística (INE) [13]. In Valencia, precisely, 5.387 persons died from cardiovascular diseases, representing the twenty percent of the total deaths for that year. Considering the biomedical and technological improvements throughout the past years, it is concerning that these are still leading causes of death.

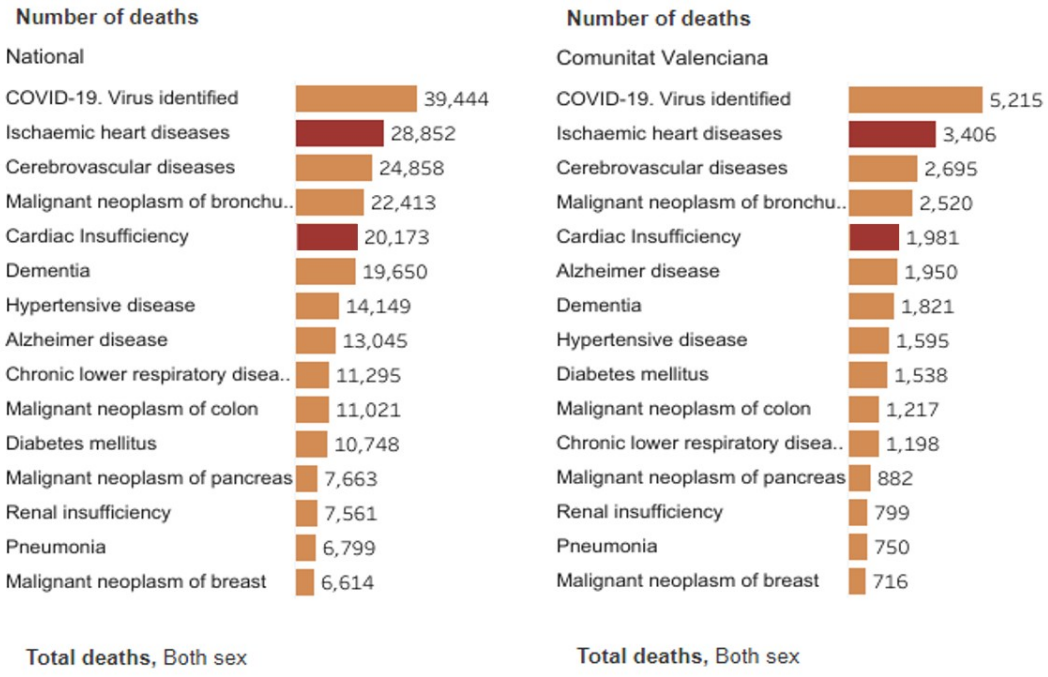

Figure 2.2: Total number of deaths with causes in Spain (left) and Comunitat Valenciana (right) during 2021. Extracted from: (Instituto Nacional de Estadística, 2021).

2.2.1 Fundamentals of electrophysiology

The bioelectrical phenomenon explains how the muscle cell contracts after experiencing an intracellular change. Because ions are electrically charged molecules, this event could be analyzed with equipment that measures electric flow, considering a potential difference that has been established [14].

The heart is the organ involved in the circulatory system and in charge of pumping the necessary blood to the lungs, so it can then drive oxygenated blood to the rest of the body. Its structure is divided into four parts: which are composed of the (left and right) atria, that receives the blood, and it passes it to the (left and right) ventricles, which pumps the blood out of the heart [15]. The principal characteristic that defines this organ is its constant movement of contraction (depolarization) and relaxation (repolarization), accompanied by the intervention of the action potential, that are defined by Wei, X. (2022) as "the rapid sequence of changes in the membrane potential, resulting in an electrical impulse". Thus, considering the bioelectrical definition, the cardiac cells can also be studied by their electrical behavior [16].

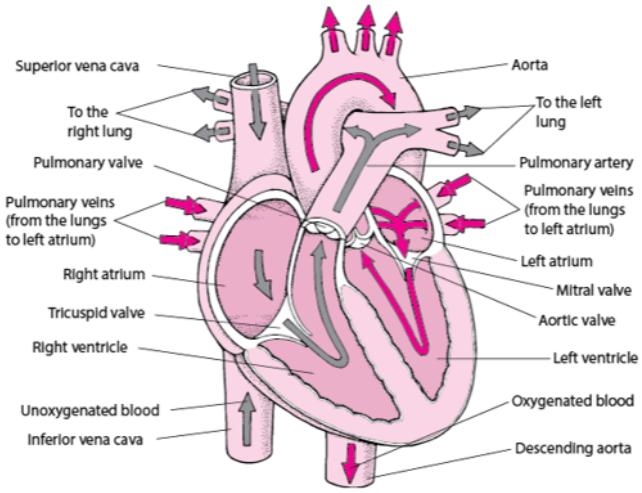

Figure 2.3: Cross-sectional view of the heart normal direction of blood flow. Extracted from: (Gupta et. al, 2022).

Then, the concept of electrophysiology englobes the study of the electrical interactions between biological cells. To be precise, the cardiac specialty focuses on the study of the electrical response of the heart after being stimulated or in an "idle" state.

Apart from the medical experience, adequate equipment is an important thing every electrophysiologist needs to achieve a remarkable work. Following this idea, the collected information from the patient leads to the selection of the equipment

Josephson, M. E. (2008) explains "the type of data collected determines what equipment is required". For instance, the electrogram (EGM) is a type of medical test with the purpose of diagnosing and monitoring cardiac problems. To register them, catheters are used, which can be defined as tubes with electrodes in the tip that are inserted in the body during treatment and surgical procedures. The actual market offers a lot of catheters, apart from the traditional ones, which has increased the performance of nowadays studies. A particular case is the one that has a deflectable tip, that helps evaluate determined areas of the heart that are difficult to reach. Thus, during the past few years, there have been a lot of advances and improvements with the initial motivation of obtaining a good electrical mapping during arrhythmia explorations [17]. However, a non-invasive method to study the behavior of the heart is the use of electrodes, an electrical conductor, on the body surface of the patient to obtain the electrocardiogram (ECG).

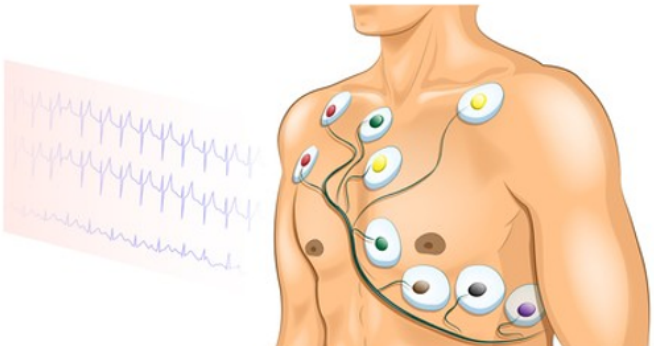

Figure 2.4: ECG electrodes. Extracted from: (Shutterstock, n.d.).

In the study of ECGs, the electrical impulses caused by the cardiac cells are detected by them, resulting in a signal with a characteristic waveform, that defines four components: the P-wave (depolarization of the atrium), the QRS complex (depolarization of the ventricle) and the T-wave (repolarization of the ventricle) [18].

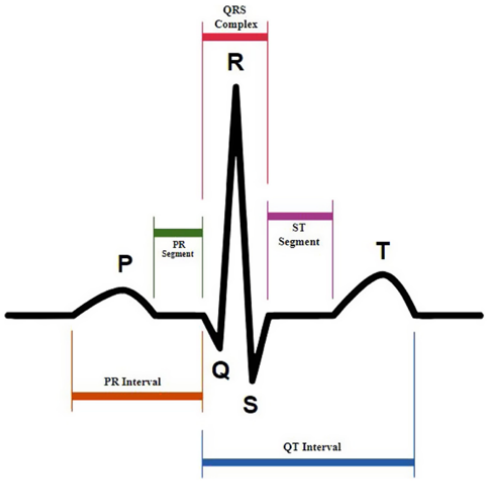

Figure 2.5: P, QRS, T waves, PR interval, and QT interval. Extracted from: (Madona et. al, 2021).

When analyzing these exams, healthcare professionals are in the position to determine whether there is an irregularity, or not, in the electrical signal. This could finally lead to the detection of cardiac pathologies and the medical expertise will advise either therapeutical treatment or a surgical intervention.

However, there is always a complexity within the invasive methods, making what was supposed to be a simple study, a surgical one. This is the main reason why a lot of researchers had invested their time looking for non-invasive alternatives that could provide more specific results, just as the electrograms. And this is when the electrocardiographic imaging (ECGi) appears as a new "modality that images cardiac arrhythmias from body surface potentials" [19].

2.2.2 The electrocardiogram as a method to detect arrhythmias

Although they have existed for more than a century [20], electrocardiograms are one of the most typical medical tests used to detect heart diseases and to monitor patients by registering the electrical changes from the cardiac activity with electrodes on the body surface.

While electrocardiograms are mostly effective and powerful, sometimes, more information is required to make an appropriate diagnosis. An arrhythmia is a condition determined by an abnormal rhythm of the heart [21]. Considering the heartbeat rate, arrhythmias can be classified as tachycardias and bradycardias, referring to it going too fast or slow, respectively. Nevertheless, this condition is not always a life threatening one, but it could be a clear lead to the detection of possible a posteriori illness.

For years, it has been studied the possibility to detect the location of arrhythmias on the cardiac tissue, which concludes in a very difficult task for the current medical methods as the ECG. The main reason why this represents a troublesome quest is because of the constant movement of the heart, ending up causing a positional variation within the electrical source and its reflected signal, that is being registered by the electrode on the thorax. In fact, this disrupts the geometry and location relation between the heart and body surfaces [22].

2.2.3 Electrocardiographic Imaging

According to MacLeod and Brooks (1998), “the ultimate, if utopian, goal of electrocardiography would be to describe the electrochemical activity of each cell in the heart, based on body-surface electrocardiograms (ECGs)”. Interesting results were found in an investigation, leaded by the Cardiovascular Research and Training Institute of the University of Utah, that used a human shaped torso tank with an isolated heart inside to map myocardial ischemia [23], concluding in the necessity to consider the geometrical relationships within heart and body surface to study the cardiac activity while using its electrical potentials.

Thus, a non-invasive alternative is the Body Surface Potential Mapping (BSPM), which records the electrical activity of the heart by using multiple electrodes (mostly more than the standard system that only has twelve), inside a vest, on the body surface of the patient. Then, the electrocardiographic imaging (ECGi) can be a helpful method to improve the obtained results, by “estimating the electrical activity on the heart surface from a dense array of body-surface ECG recordings and a patient-specific heart-torso geometry [8].

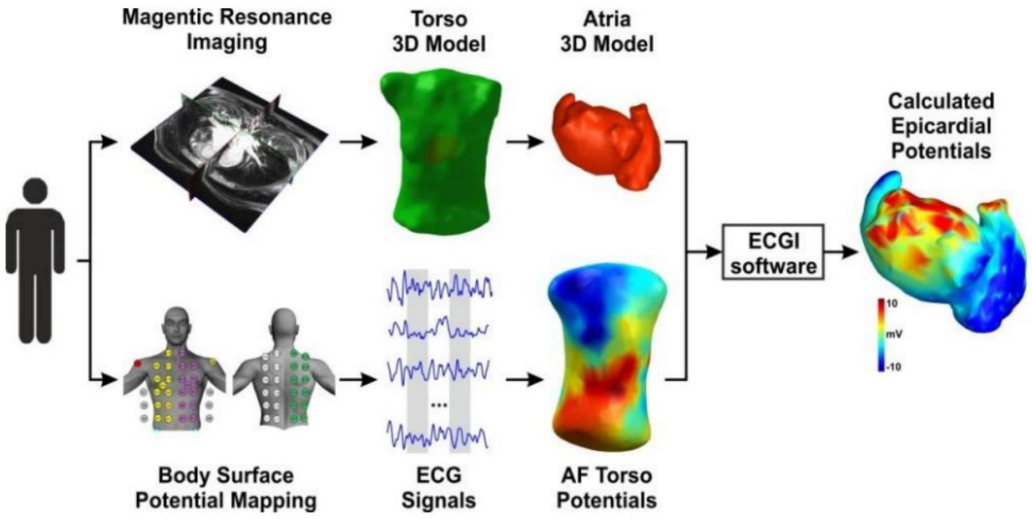

Figure 2.6: Workflow of electrocardiographic imaging technique. Extracted from: (Hernández 2019).

2.2.3.1 The inverse problem of electrocardiography

The principle of the ECGi is determined by the inverse problem of electrocardiography, a variation of the mathematical one, which features the same problems related to its ill-posedness condition and non-uniqueness solution. The already known information is the electrical potentials at the body surface, while the main goal is to reconstruct the cardiac information [24]. Subsequently, the forward problem involves calculating the body surface potentials, typically using dipoles (representing the electrical activity of the heart) or previously recorded potentials from the exterior of the heart (epicardium) [25].

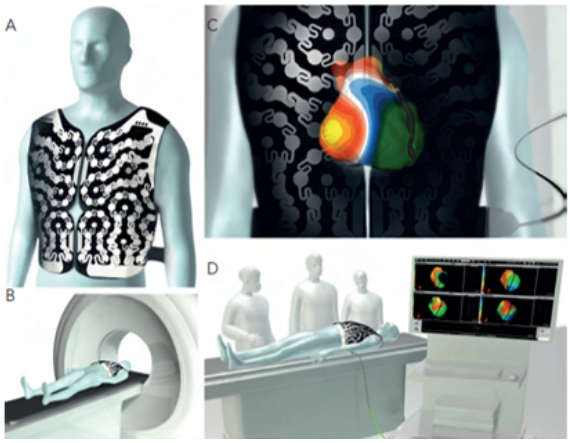

Figure 2.7: A. 252 electrode vest, B. heart-torso geometry (CT scan), C. acquisition of the cardiac signals (forward problem), D. pre or per-procedural instantaneous maps (inverse problem). Extracted from: (Shah et. al, 2013).

2.3 Fundamentals of Artificial Intelligence

For ages, the idea of simplifying a lot of procedures has been a goal for a lot of researchers and professionals in different areas. The concept of Artificial Intelligence (AI) refers to the use of machines to perform commonly human-like tasks and, just as the industrial revolution in its time, it is starting to become the new technological wave nowadays.

Although AI can be classified in a lot of different ways, there are two main categories: narrow (NAI) and general (GAI). The first one is focused on a "structured" task, that is normally an individual problem, like speech or face recognition. Instead, the latter one refers to the possibility of the machine to do multitasking, making it more robust to respond to complex environments. A good example of general AI are chatbots, whose use has increased incredibly in different areas during the last few years. But, of course, GAIs are still a work in progress, because they highly rely on NAIs, and a lot of them have not strong backgrounds yet.

The concept of AI has been around for a while, since the 20th century precisely. The considered "first AI program" was presented during 1956 by Newell, Shaw and Simon [26], researchers interested in the computer science field. This moment probably symbolized the beginning of the artificial intelligence, based on the idea that it was the first "automated reasoning" program developed. Since then, a lot of improvements have been made over the years.

Just as other fields, the AI sector had its golden years which were full of the high expectations from users and media. However, after a few decades, this "propaganda" ended up backfiring after most of these presumptions and goals were not being achieved as fast as it was expected. In consequence, a lot of investors dropped their funds for research from the 1970s to the 1990s [27], a period that was named "the AI winter".

Fortunately, this period ended after recent improvements during the 90s lifted the faith up in the field again. Since then, the artificial intelligence has become a constantly evolving technology that also accomplished the creation of different subsets such as machine and deep learning. The founder and Chief Technology Officer of BrainChip Ltd, a company that produces advanced AI processors in digital, Van Der Made (2023) confirms: "the future of artificial intelligence appears bright with continued advancements in technology. Investment in artificial intelligence reached 93.5 billion dollars in 2021, according to Statista" [28].

Further advances from artificial intelligence consist of the development of ways to allow the machines to learn from a priori information and improve their performance over time. This is defined as Machine Learning (ML) and, in other words, it is the technique that aims to enhance the performance of the system by learning from gained experience, which is defined by obtained data [29].

Machine Learning can be divided into two types of learning: unsupervised and supervised. Both categories refer to the labelling of the data, i.e., if the data is defined (supervised), or not (unsupervised), with labels. These concepts can be helpful to understand the way the machine is learning while training, but both correspond to different approaches that can lead to various implementations: with non-labeled data, the machine will start to study the underlying information and more specific features, which can be helpful when doing dimensional reduction for image improvement. On the other hand, labeled data can represent a wise decision if the application is related to data classification.

2.4 Deep Learning

Deep Learning (DL) is a subset of ML that uses neural networks (NN) to create modules that allow the change of the representation level, increasing it into a more abstract one [30]. DL is also characterized by its use in feature extraction, while ML is not utilized for that.

Another notorious difference between machine and deep learning is determined by the complex and accented type of learning the latter one involves, that highly increases the computational cost for training the datasets. Of course, there are other distinctions, e.g., the data type and volume, which can be smaller and more structured in machine learning approaches.

A regression algorithm evaluates the relationship between the extracted features from the input data and the metric that measures the error, which will be later explained as the loss function. Considering this, a simple neuron (also known as perceptron) executes an iterative process, comparing the obtained result with the expected one. The main purpose of the neuron is to decrease the loss value by learning from the previous results [31].

2.4.1 Neural Networks

Probably, the most important element of deep learning structures are neural networks, which are composed of layers with interconnected nodes, vaguely simulating the human brain anatomy. Each neuron is sending information to another one, assigning a "weight" [32] to the connection, which will be multiplied by the value the node receives. Later, a number is obtained after summing these products, and it will be used to compare it to a threshold value, to determine if the data is going to be passed to the next layer. For instance, if the number is higher than the reference value, it will be used for the rest of the connections. All the modifications of the weight values occur during the training, while passing the train dataset through different layers already defined by the model structure.

It is common to assume most NN use a sequential workflow but, in reality, the connections are not always like that. In fact, feed-forward networks are the ones that take the input and lead them to an output directly, and recurrent networks are the ones that have a feed-back approach. Although they are both significantly useful, it is necessary to know when it is better to use one or another.

On the other hand, when training the NN, the loss function is computed, which evaluates how accurate the predictions are, compared to the desired result. Following this idea, backpropagation is a method that aims to reduce the loss function, by calculating its gradients for each weight, going from the last to the first layer [33].

Furthermore, some aspects are considered important characteristics of neural networks: the processing units, the connections, and the activation functions.

2.4.1.1 Processing units

The processing units are supposed to get the input data and send information to another cell, assigning a "weight" to the connection, which will be multiplied by the value the node receives.

The processing units [34] can be classified as:

- *Input units*, that receive the data.
- *Hidden units*, which cannot be visualized from outside because their inputs and outputs are inside the network.
- *Output units*, that pushes data out of the neural network.

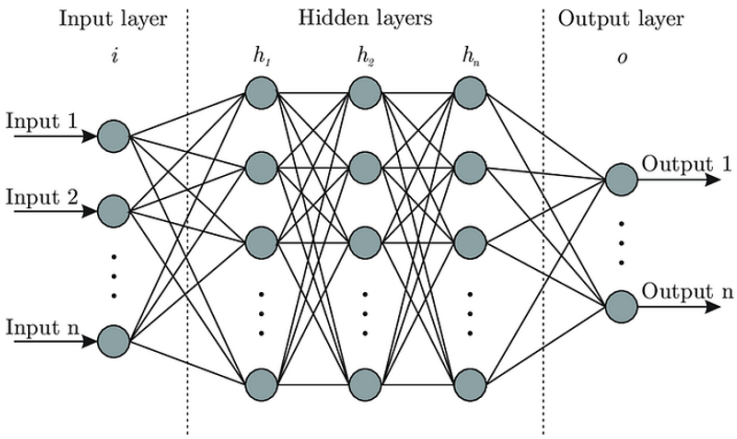

Figure 2.8: Processing units of neural networks. Extracted from: (Bhardwaj, 2022).

2.4.1.2 Connections

Normally, units offer some information to the next one through their connection. As was explained before, what each cell provides is the result of multiplying the weight with the input. Later, a number is obtained after summing these products, and it will be used to compare it to a threshold value, to determine if the data is going to be passed to the next layer. For instance, if the number is higher than the reference value, it will be used for the rest of the connections.

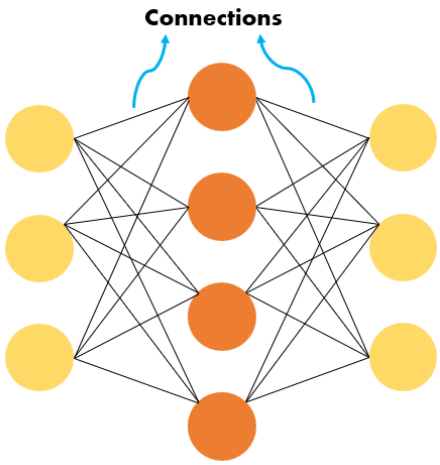

Figure 2.9: Connections of neural networks.

2.4.1.3 Activation functions

The transfer function is used to get a proper output from the input, limiting the result within a determined range. The activation functions can be classified as: (1) *linear*, which does not really map the result into a finite range, (2) *non-linear*, whose features allow the model to increase its generalization ability.

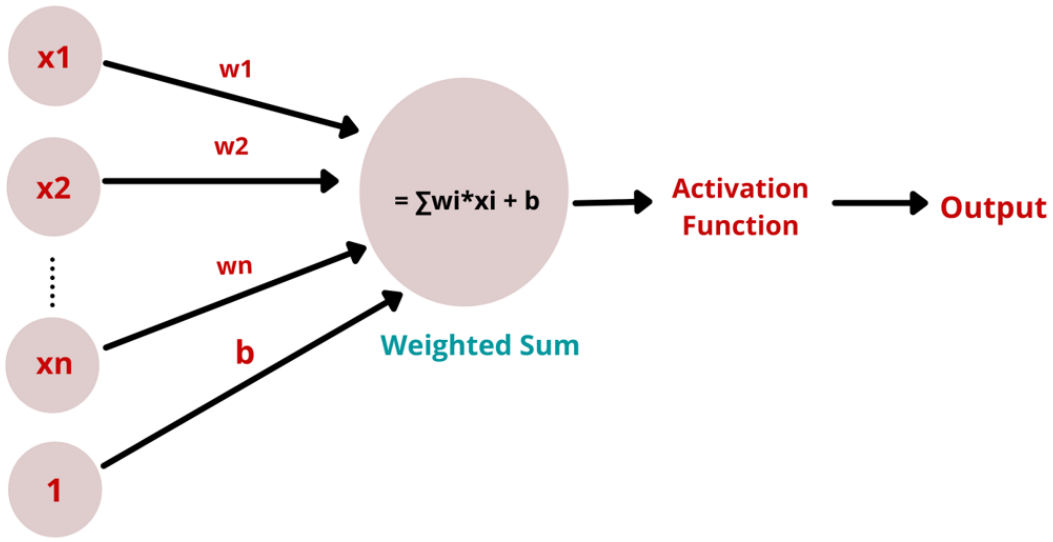

Figure 2.10: A simple neural network with with its weight, activation function, and output. Extracted from: (Debnath, 2021).

The non-linear activation functions are the most used type; therefore, there are other sub-classifications [35]:

- *Sigmoid* activation function limits the result within a 0 to 1 range, which could be very useful for probability problems.
- *Hyperbolic tangent* activation function resembles the behavior of the sigmoid but in a -1 to 1 range. Considering this, the function increases the intensity used to mark the negative, zero and positive values, which can be of help when treating data that will be classified into two different classes.
- *Rectified Linear Unit (ReLU)* activation function is similar to the sigmoid, but part of the curve is rectified, so the range is between 0 and infinite.
- *Leaky Rectified Linear Unit (Leaky ReLU)* activation function is a variation from the original ReLU function. This takes into consideration the possibility of not correctly mapping the negative data inputs because it is always transforming them into zeros. To solve this, it adds an offset to the range, usually 0.01, that can make the response more accurate.
- *Swish* activation function is also rectified, but when it gets near 0 in a smoother way, not as abrupt as with ReLU, for instance.

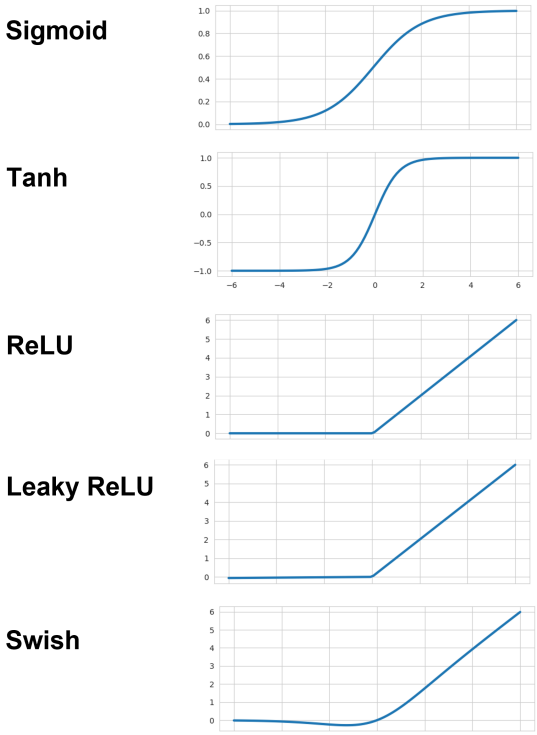

Figure 2.11: Different activation functions and their graphs. Extracted from: (Tanner, n.d.).

2.5 Computer vision

One of the many fields of artificial intelligence is computer vision, whose principal purpose is "teaching computers to analyze and comprehend visual input from their surroundings" [36].

Working with images or video can be a difficult task if data is not pre-processed or simplified with analysis concepts. Each frame has features, which can be defined as a particular characteristic that can help the computer identify elements. They can be classified as:

- *Low-level features*, which are the simplest ones. They can be edges, colors, or contours of the image. These attributes are usually a unique part of the image, meaning that it does not necessarily have a relationship with other types of images. Based on that idea, these features can be highly influenced by changes from the raw image.
- *High-level features*, whose characteristics are a consequence of the low-level ones. In fact, they represent complex details like objects, scenarios, or behaviors, which can also be generalized to other images. In general, these type of features can be considered more general and, therefore, robust.

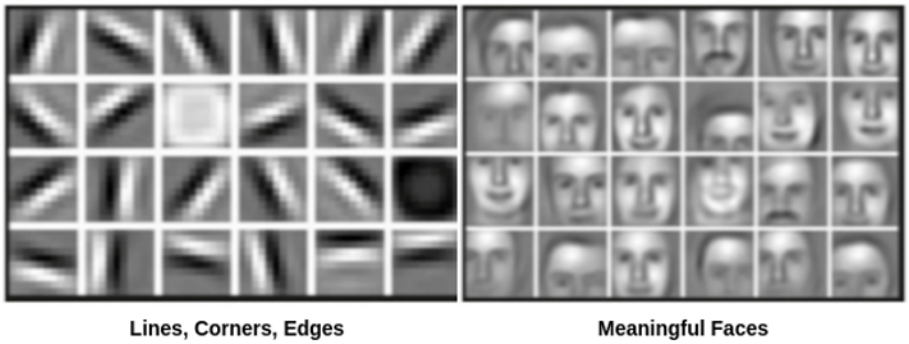

Figure 2.12: Example of low and high level features in images of faces. Extracted from: (Nanos, 2023).

High-level features can be more complicated to identify and learn, the main and basic tool used to extract this kind of features is the convolutional neural networks (CNNs).

2.5.1 Convolutional Neural Networks

A convolutional neural network (CNN) is considered a type of network that uses convolutional architectures to automatically extract features from data [37]. A few advantages from using this type of net can be pointed out: in the traditional artificial neural network (ANN), a lot of parameters coexist, which can lead to a slower behavior. In this case, this is abolished by disconnecting some neurons and making them share the same weight. The number of parameters can increase if the input image has big dimensions, but it can also be reduced if it has layers dedicated to downsample them (without losing important information).

2.5.1.1 Pooling layers

A whole convolutional neural network could be improved by adding different layers from the convolutional ones. The use of pooling layers can decrease the dimension of the feature maps and, as a consequence, not only reduce the number of parameters (as it was explained before), but also simplify the whole net model. An advantage these layers have is the "translational invariance" quality, which helps the CNN still recognize the characteristics, even if the input has changed position, which can be very helpful in case it is necessary to detect a feature that vary its location [38]. Considering this, there are two types of pooling layers to achieve this:

- *Max pooling*, that produces a sharper output by extracting the maximum values of the pools.
- *Average pooling*, which uses the average values of the pools, smoothing the image instead.

The pooling layers aim to downsample the input by reducing the dimension of the feature maps. However, at some point, it could be necessary to get back to a certain dimension. To do so, the Upsampling layers can be used, which repeat the columns and rows of the input [39].

2.5.1.2 Convolutional layers

Each convolutional layer provides a number of characteristics which will be called feature maps, that are an essential output to understand the most important patrons of the image. To calculate these maps, the layer will do the convolution, a mathematical linear operation., between the input and the kernel matrix; whose result will conclude in the way one of the shapes is being altered by the influence of the other one.

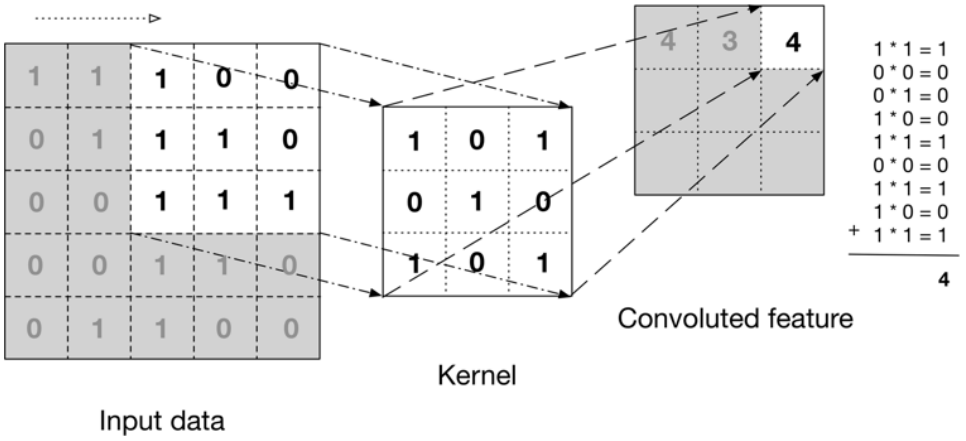

Figure 2.13: Example of convolutional operation. Extracted from: (Potrimba, 2023).

Normally, the kernel is a two-dimensional square matrix that will be doing the convolution with the image. Thus, to map the features, the kernel goes over every element of the input, proceeding with the operation, to finally result in a simpler matrix, with less dimensions. To do so, the kernel is moving from left to right with a size of the step named stride, e.g., if the stride is three, then the kernel will move by three rows per columns. On the other hand, if the image is an RGB one, a new dimension is added; therefore, a three-dimensional kernel would be required, i.e., a filter.

Apart from the kernel and the stride, the padding is also an important filter hyperparameter. Its definition relays on adding a value (usually zeros) to the boundary of the input image. Then, it can be (1) *valid*, if no padding is necessary; (2) *same*, so the feature maps will have the inverse of the stride as its size; or (3) *full*, when the padding will be maximum to the point its limits passed the convolution too.

2.5.2 Data processing

In deep learning applications, the data needs to be correctly processed so it will not negatively affect the training. If anything, a pre-processing can be done, just like the normalization of the input, that needs to be chosen taking into account the main purpose of the application. By normalizing the input, the output will be within a determined range.

Another a priori method is to augment the data, which would help the network to have a more accurate response, based on the idea that all deep learning approaches require a high volumes of information as an input. With data augmentation, the initial images are used to create new ones,

while doing techniques that can rotate, crop, flip or change its color, for instance. If that is not enough, a batch normalization could be applied, that can make the net more independent from the initialization. These layers aim to maintain the mean output close to 0 and its standard deviation to 1 [40].

2.5.3 Training hyperparameters

Just as the neural networks, when training, there are a few hyperparameters that could be useful to optimize the behavior of the results.

- *Epochs*. The number of times the model goes over the training data set and updates its weight, i.e., the total iterations.
- *Batches*. Defining a batch size delimits the group of data from the training dataset that will be used per epoch.
- *Loss function*. They are used to quantify and know how accurate obtained results are, considering the desired and predicted ones.
- *Learning rate*. It determines the rate at which the model updates its weights.

2.5.4 Output analysis

When results have been obtained and it is time to analyze them it is common to start looking for ways to improve the general behavior. But first, it is essential to previously understand the different possible ways a neural network could respond. To do so, it can be helpful the use of learning curves, a plot of learning over experience. Metrics that are better if their value is decreasing are usually used, such as errors, for example.

If the learning curve of the training is represented, then, it will offer a result that indicates how well the model is learning features from the training dataset. And, if the learning curve is based on the validation set, it will provide an orientation about the generalization of the model. Nevertheless, what is usual is to plot both of the loss curves together to understand the results better [41].

2.5.4.1 Underfitting

Usually, this can be identified when the validation curve is very over the training one, that keeps a flat response. This means that the model is not getting all the data from the training, leading to it not being capable of obtaining a low error.

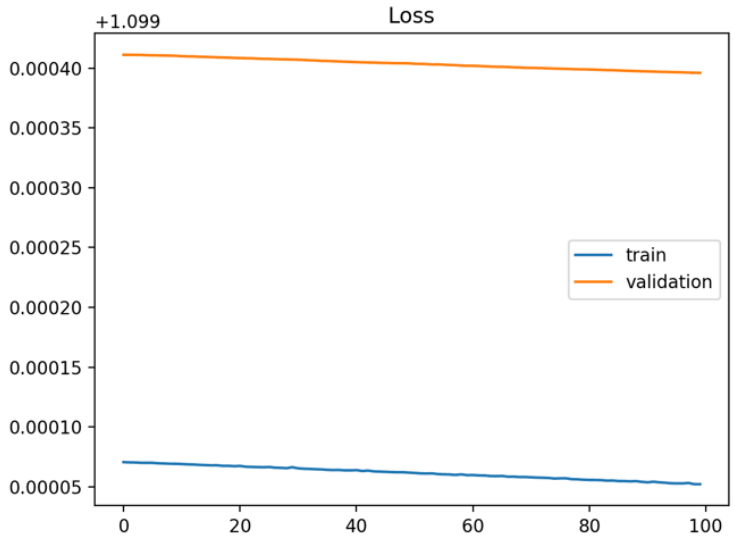

Figure 2.14: Example of underfitting. Extracted from: (Brownlee, 2019).

However, sometimes, it can also be underfitting if the validation curve is below the training one and it is constantly decreasing. This can end up being good news if the number of epochs were too low and the training process was cut off ahead of time.

Overcoming this problem can be an easy task if the model is more robust, i.e., it needs to have more hidden nodes. Also, increasing the number of epochs will make the training longer and the probability of the curves reaching stability could increase.

2.5.4.2 Overfitting

When the model has learned all features from the training dataset too well, to the point it cannot generalize anymore, the curve ends up showing overfitting. This can be detected if the training curve is decreasing non-stop or if at first the validation loss is correctly decreasing but then it starts increasing.

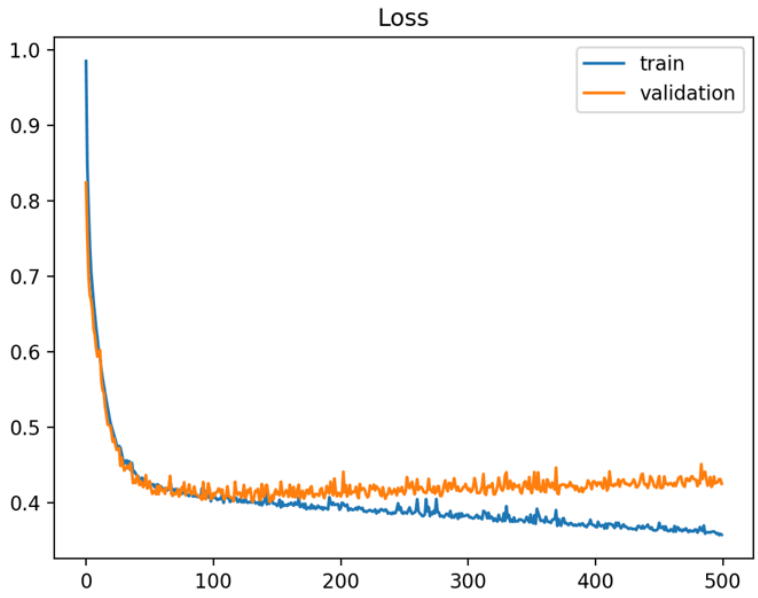

Figure 2.15: Example of overfitting. Extracted from: (Brownlee, 2019).

This is a usual problem that could be solved with a bigger and more representative dataset. But, if there is no possibility of getting more information, then using data augmentation techniques can be a good approach. It could also be appropriate to use regularization methods for the model, like the insertion of dropout layers, which will be explained later.

2.5.4.3 Good fit

When both loss learning curves are parallel (separated by a small gap) and decrease simultaneously until a stable point, the goal has been achieved. Nevertheless, it is important to prevent possible overfitting by establishing only the necessary number of epochs, so the model will not be over-trained.

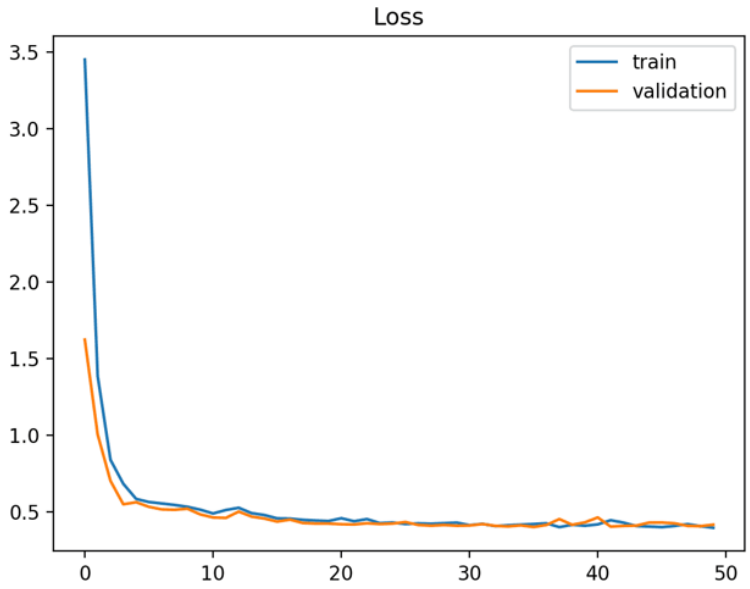

Figure 2.16: Example of a good fit. Extracted from: (Brownlee, 2019).

Although the previous behaviors are the most typical ones, it can always be possible to have a different response that shows an unrepresentative dataset. If the training set is unrepresentative, like not having enough examples compared to the validation data, it limits the learning progress. On the other hand, if the validation set has less variety of samples, relative to the training dataset, it will not have enough information to confirm the generalization ability of the model. Both of these are usually represented by some kind of oscillations in the plots.

After identifying the resultant behavior of the neural network, it is the time to optimize it a little bit more with different techniques: the learning rate defines how often the weights are being updated, this can be useful to improve the convergence of the model. Other techniques are regularization methods; for instance, dropout is one of the most used ones and it consists of disconnecting a few neurons, allowing the network to prevent overfitting by not learning non-general features. Still, this is not always the method that solves overfitting, that is why early stopping is also considered. It stops the training process, considering the time when the validation loss curve is about to increase. An advantage this method has is that the number of epochs it will be waiting for until it stops can be manually fixed.

2.5.5 Image denoising

When working with real images, distinct from images from previously elaborated databases, a lot of noise is present. The causes could vary depending on the sources, but it could be because of, as Solomon (2021) explains: “electric signal instabilities, malfunctioning of camera sensors, poor lighting conditions, errors in data transmission over long distances, etc” [42]. Nowadays, some areas that use the processing of images, such as the medical sector, need them to be as clear, detailed, and representative as possible. Focusing on that specialty, prevention and detection studies are really important to obtain the most accurate results, considering a life could be at stake.

One of the many applications of computer vision is image denoising, which intends to remove undesired noise while maintaining the principal features and quality. During the past few years, a lot of researchers had concluded that convolutional neural networks are a great approach to solve this problem, which opens up a new path to the discovery and development of new deep learning techniques in image processing processes. In this case, the net will receive the noisy images as the input dataset and the clean images as the "goal standard", i.e., the desired output.

Figure 2.17: Image denoising. Extracted from: (Quick Image Processing Research Guide, 2020).

2.5.5.1 The autoencoder model

To apply image denoising, there are different approaches using DL. One of them showed a better result: the autoencoders, whose structure consists of an encoder and a decoder. The first one is focused on learning features from a compressed version of the input image, and the latter one decodes the data by reconstructing it. Still, this definition can be dig up more to understand its real behavior. Encoders are in charge of taking the input and decreasing its dimensions by the use of different types of layers whose other function is extracting relevant attributes. Then, decoders take this representation and try to reconstruct it, so it can resemble the original image [43].

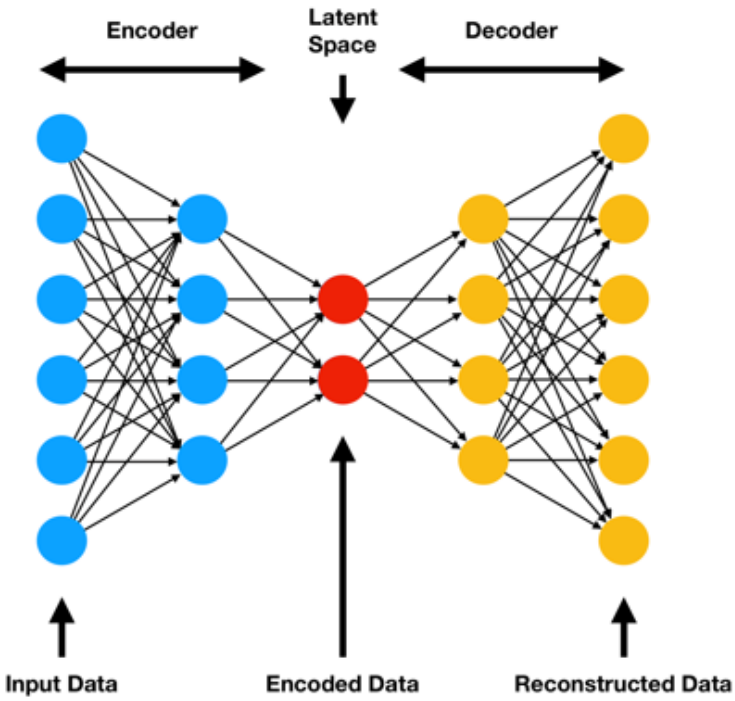

Figure 2.18: Structure of an autoencoder. Extracted from: (Flores, n.d.).

An interesting fact about autoencoders is the type of shape of their schematic, very similar to a bow-tie, where the central part contains the most compressed images. The latent space, as it is called this section, is probably the most important part of the network because it forces the learning of more specific features, discarding possible noise or unnecessary characteristics. In comparison, it could be helpful to think about what would happen if the input and latent space information had the same dimensions. In that case, the network would learn everything, and the output would be too similar to the original data. Then, a good structured latent space makes the autoencoder recognize general patterns, making it capable of reconstructing unseen images, while attempting to minimize the error as much as it can.

Chapter 3

Methodology

In this chapter, the methodology used to create the database and further development of neural network architectures and their parameters is presented.

The chapter is structured as follows:

- In Section 3.1, the methods to generate the ellipsoids and the database are described.
- In Section 3.2, three architectures that will be used and studied are presented.
- In Section 3.3, the way the database was divided into datasets, and the data pre-processing techniques that were applied are explained.

3.1 Generation of the ellipsoids

Considering the main concept of the inverse problem of electrocardiography, it could be of interest to determine if the utilization of artificial intelligence, precisely a deep learning approach with denoising techniques, would be a helpful post-processing way to provide more accurate information.

Contemplating this idea as a proof-of-concept, the heart and body surfaces can be modeled as two concentric ellipsoids with different sizes, always respecting that the geometry resembling the body needs to be greater than the other one acting as the heart. Each case is generated in a different way since every spherical mesh will be altered by aleatory changing the Cartesian coordinates for the three-dimensional (3D) space. Moreover, to represent the electrical activity of the geometries, a dipole that randomly changes its orientation will be added as well; taking into account that "an electric dipole is tagged as a pair of objects which possess equal and opposite charges, parted by a significantly small distance" [44].

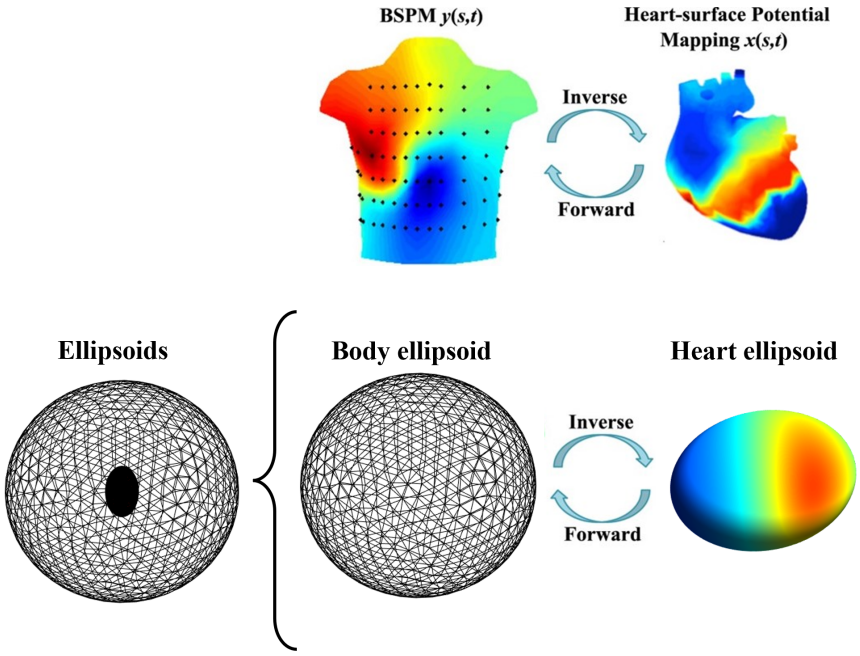

Figure 3.1: Analogy between the ECGi and the proposed thesis.

The generated dipoles are defined by the following equation (3.1), which represents the retarded scalar potential for an arbitrary charge and current located in a specific volume [45]:

$$V(r,t) \approx \frac{1}{4\pi\epsilon_0} \cdot \left[\frac{Q}{r} + \frac{\hat{r} \cdot p(t_0)}{r^2} + \frac{\hat{r} \cdot \dot{p}_{(t_0)}}{rc} \right], \tag{3.1}$$

where V is the electric potential of the dipole, ϵ_0 is the electric permittivity of free space, Q is the total charge at time t_0, r is a far field point, $\hat{r}$ is the unit vector of r, $p(t_0)$ represents the electric dipole moment at t_0, $\dot{p}(t_0)$ is the double differentiated electric dipole moment, and c is the ratio of the amplitudes of the electric and magnetic field ($c = \frac{E_0}{B_0}$).

Although the orientations of the dipoles are aleatory, they are delimited within a [-1, 1] range. On the other hand, the deformation of the geometries is done by changing the values of their positions in the x, y, and z axes. This variation is restricted within a [1, 7] range for the heart deformation, and the range for the body deformation will be the same plus 2 to avoid this geometry being smaller than the other one.

Thus, the main idea is to evaluate whether the results obtained from the Tikhonov regularization for obtaining the inverse dipole on the ellipsoid equivalent to the heart could be improved by a neural network:

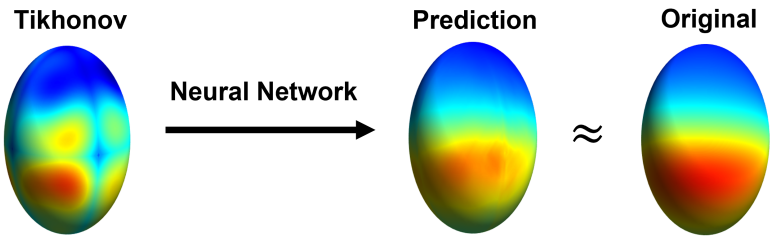

Figure 3.2: Workflow of the proposed proof-of-concept.

However, these geometries are not perfect; in fact, the irregular distribution of the nodes can alter their own connections, increasing the complexity of the models. Apart from that, to use two-dimensional (2D) convolutional layers, it would be necessary to have the ellipsoids in a plane. This process have been already done for cardiac geometries, that can be homogenized by being parameterized in spherical coordinates to prevent measurements incompatibilities. This aims to rearrange the model in a 2D spaced with regular spacing.

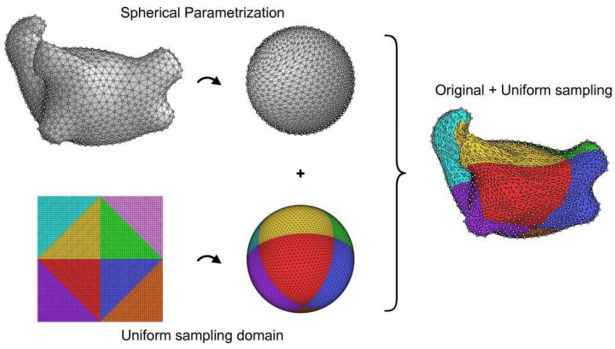

Figure 3.3: Remeshing process using spherical parametrizations. Extracted from: (Hernández 2019).

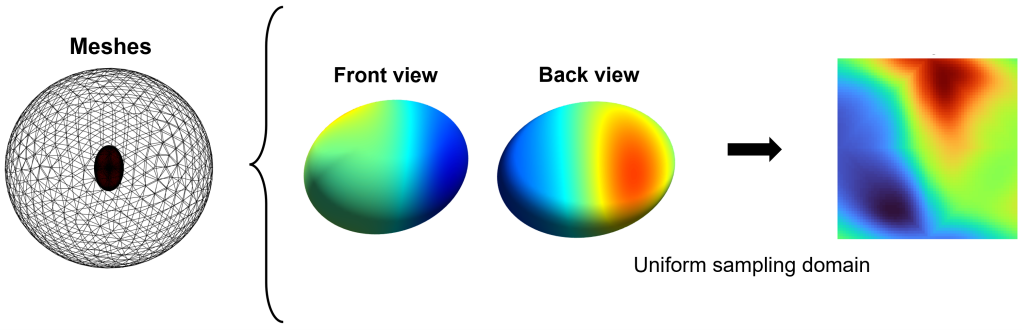

Figure 3.4: Example of the remeshing process for a generated ellipsoid.

After applying the previous method to the geometries, the method of regularization is applied.

Thus, there will be two matrices: the one that corresponds to the 2D plane of the data that was analytically calculated, and the result from Tikhonov. And, from now on, these matrices will be treated as images.

3.1.1 Database creation

A database is created with 16.000 cases for each group of data required: the original images, calculated by the analytic approach already explained; and Tikhonov (Sub-section 2.1.2) images, whose principal characteristic is the geometrical noise introduced by the Tikhonov regularization during the inverse problem computation. Finally, the images were generated with a resolution of 65x65 pixels, which will be later changed to 66x66, by duplicating the last column and row. This is a more manageable input for the models since the autoencoder will reduce and increment the image, by dividing or multiplying the dimensions with a value. For instance, if the dimension can be divided by more numbers, then there could be a bigger downsampling.

3.2 Architectures

The architecture of the neural network needed to be chosen based on the main purpose of this thesis, which is denoising the Tikhonov images, i.e., the ones that resulted from applying the inverse problem. As a matter of fact, the autoencoder was previously mentioned as an architecture that presented a satisfactory result, and that is why it will be the principal focus as a model.

3.2.1 First architecture

Defining a plain architecture could be a good first approach to evaluate the behavior of the neural network, by using it as the reference model that will be improved. A few basic examples of autoencoders [46] have been analyzed. Thus, 16 layers are proposed, with convolutional layers in the encoder and decoder, but also Max Pooling layers (2.5.1), which is used in the examples. Apart from that, and considering the possibility of getting results with overfitting, dropout layers are added before every convolutional layer, to “disconnect” some nodes. For all the convolutional layers, the size of the strides is (1, 1) to get more details from the data, the kernel dimension is (3, 3), and the filters go from 32 to 8 in total. The Max Pooling and Up Sampling layers will have a size of (2, 2) and (3, 3) to reduce the (66, 66) image to a (11, 11) one. The activation function will be ReLU for all the layes, except the last one that will be sigmoid.

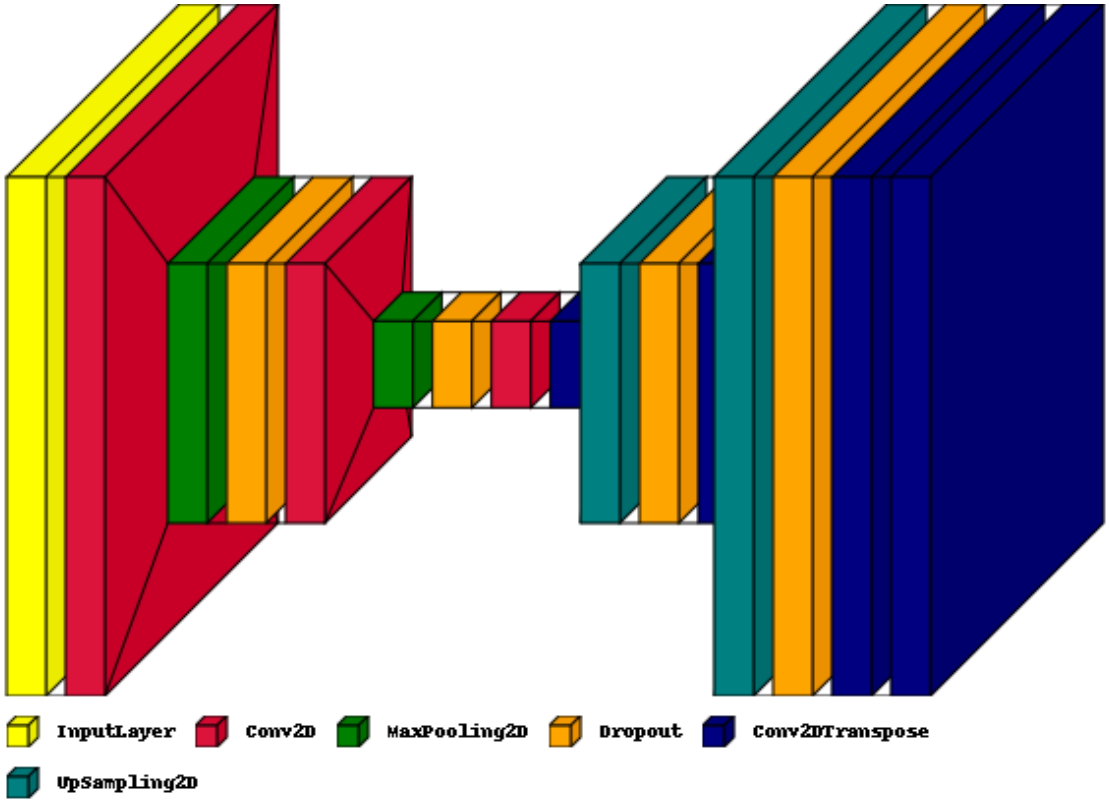

Figure 3.5: Schematic of the first architecture.

3.2.2 Second architecture

The previous structure needs to be studied, considering the characteristics of the generated dipoles. Then, the max pooling layers are replaced with average pooling layers, keeping in mind that the last ones aim to smooth the image (2.5.1). On the other hand, a batch normalization layer is introduced at the end, aiming to reduce the significance of weight initialization and, of course, evaluate if this could represent an improvement in the performance of the neural network. The convolutional layers will have the same stride and kernel size: (1, 1) and (3, 3) respectively. This time, the filters will constantly be 66, aiming to get more features. The Average Pooling and Up Sampling layers will have the same dimensions to reduce the (66, 66) image to a (11, 11) one. Once again, ReLU is used for all the layers, except the last one (sigmoid).

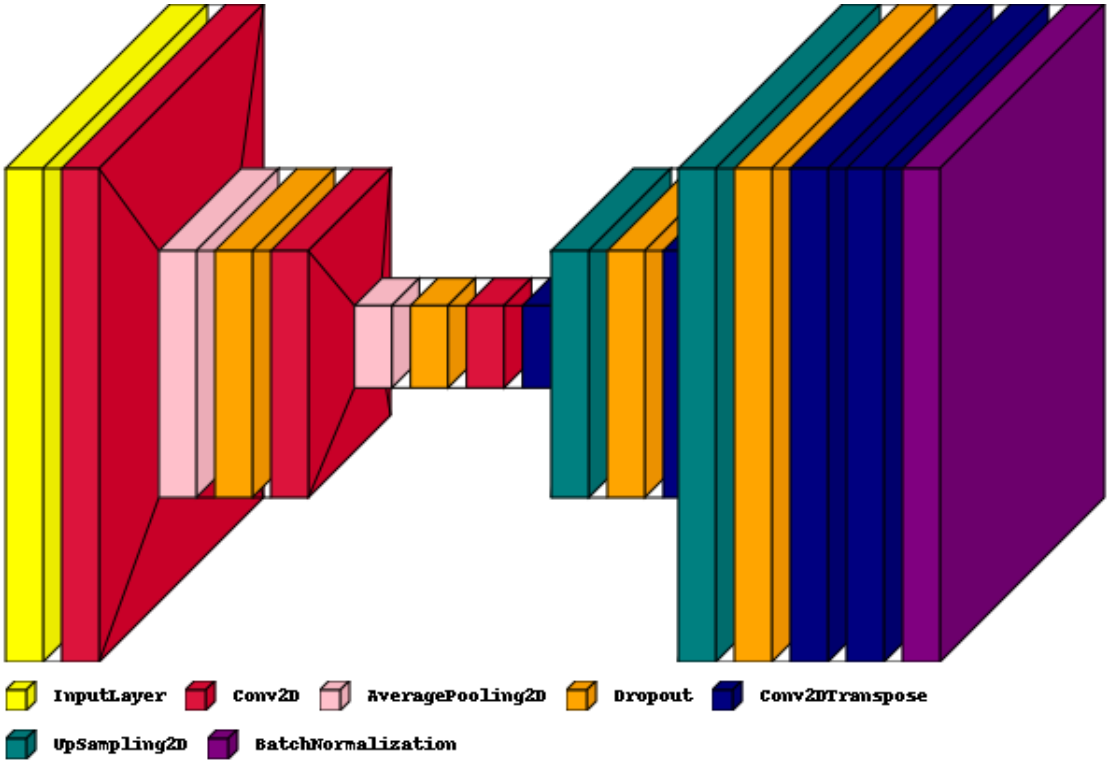

Figure 3.6: Schematic of the second architecture.

3.2.3 Third architecture

The last architecture presented has a more complex structure, with a total of 26 layers. Following the previous model, this one includes batch normalization layers in different parts, instead of only using them at the end. They have been added after the input layer and every convolutional layer, and before up sampling layers. The rest of parameters will be the same as before: 66 filters, (1, 1) strides size, and (3, 3) kernel size for convolutional layers; Average Pooling and Up Sampling layers with a size of (2, 2) and (3, 3). This last architecture also has all layers using ReLU as the activation function, but last one is swish, instead of sigmoid.

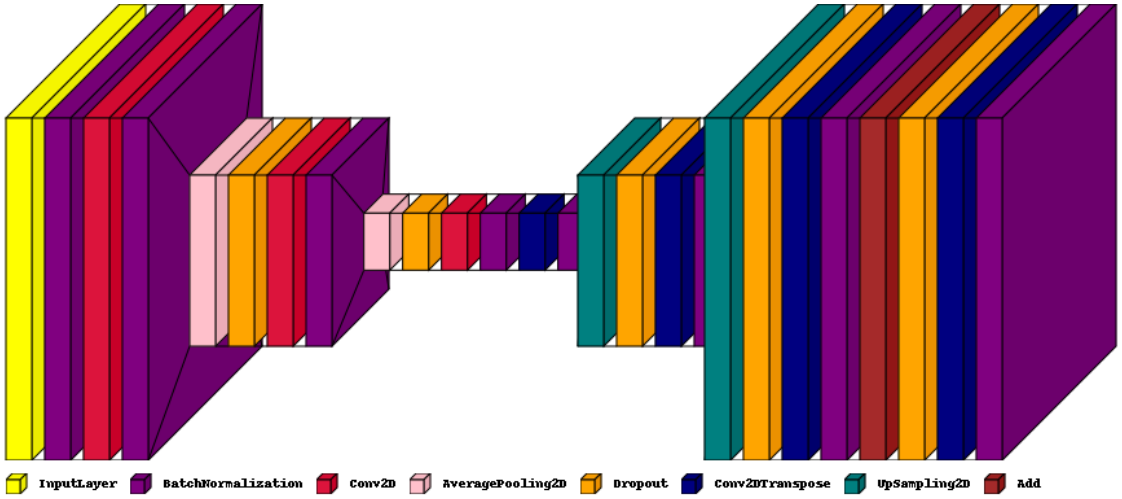

Figure 3.7: Schematic of the third architecture.

3.3 Data pre-processing and definition of the hyperparameters

Before doing the training, it is required to divide the database into three different datasets: (1) *training*, which is the dataset that will be used for training; (2) *validation*, which will also be used during training as a way to validate the ability of the model to learn these new features; and (3) *test*, which will be used later as the input data to predict the model. Then, the 16.000 analytic images and the other 16.000 Tikhonov images are split into: 12.000 for training, 3.000 for validation, and 1.000 for test.

As the potential of the dipoles were randomly defined, there are negative and positive values. Data is delimited within a [0, 1] range, considering that this could accelerate the training progress if sigmoid or ReLU activation functions are used, which also define their output in that range. To do so, the following normalization (3.2) is applied, which consists of using the 0 from the data as the 0.5 in the new range delimitation,

$$X_{norm} = \frac{X}{(2 \cdot max(X)) + 0.5}, \tag{3.2}$$

where X is the matrix of the image which will be normalized, Y is a matrix full of ones with the same dimension as X, and x_{norm} is the normalized matrix of data.

The hyperparameters are defined as follows: the number of epochs is specified as 100, the batch size as 32 and the learning rate as 0.0005. On the other hand, the metric used to quantitatively measure the results from the training is the mean absolute error (MAE), which will indicate if the NN is capable of reconstructing the amplitude of the values (3.3), and the loss function is the mean squared error (MSE), determined by 3.4:

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |x_i - \hat{x_i}|. \tag{3.3}$$

$$MSE = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \hat{x_i})^2, \tag{3.4}$$

where, in both cases, *x* is the ground truth, $\hat{x}$ is the predicted value, and *n* the total number of the dataset.

On the other hand, to do a further evaluation of the results, the correlation will be used to evaluate the performance of the NN as well. This will indicate whether the NN has learned the pattern of the dipoles. To do so, the correlation coefficients [47] are calculated, as follows:

$$CC = \frac{1}{n-1} \sum_{i=1}^{n} (\frac{x_i - \mu_x}{\sigma_x})(\frac{\hat{x_i} - \mu_{\hat{x}}}{\sigma_{\hat{x}}}). \tag{3.5}$$

Once again, *x*, $\hat{x}$ and *n* are the same as before (true and estimated values, and total number of data). μ_x and $\mu_{\hat{x}}$ correspond to the mean of *x* and $\hat{x}$, respectively. As well, σ_x and $\sigma_{\hat{x}}$ are the standard deviation of *x* and $\hat{x}$.

Chapter 4

Results

This chapter describes the results obtained from the implementation of three different NN architectures and presents a further analysis on the effects of the data distribution during the training.

The chapter is structured as follows:

- In Section 4.1, the training and validation learning curves when monitoring the loss function and MAE are presented.
- In Section 4.2, the MAE and correlation results for the three model architectures and training dataset distributions are shown.
- In Section 4.3, the predictions are represented as images.

4.1 Learning curves

As was explained before, the test dataset is always the same, allowing a better analysis of the predictions. However, it seems adequate to evaluate whether, or not, the model architecture is the only factor that can be influential to the results. Thus, three aleatory distributions of the training dataset are used to train, while the same test dataset is evaluated.

Both the loss function (4.1) and the MAE (4.2) were evaluated, and they can be analyzed with the learning curves of the training and validation, which will indicate the ability of the network to learn and generalize the datasets.

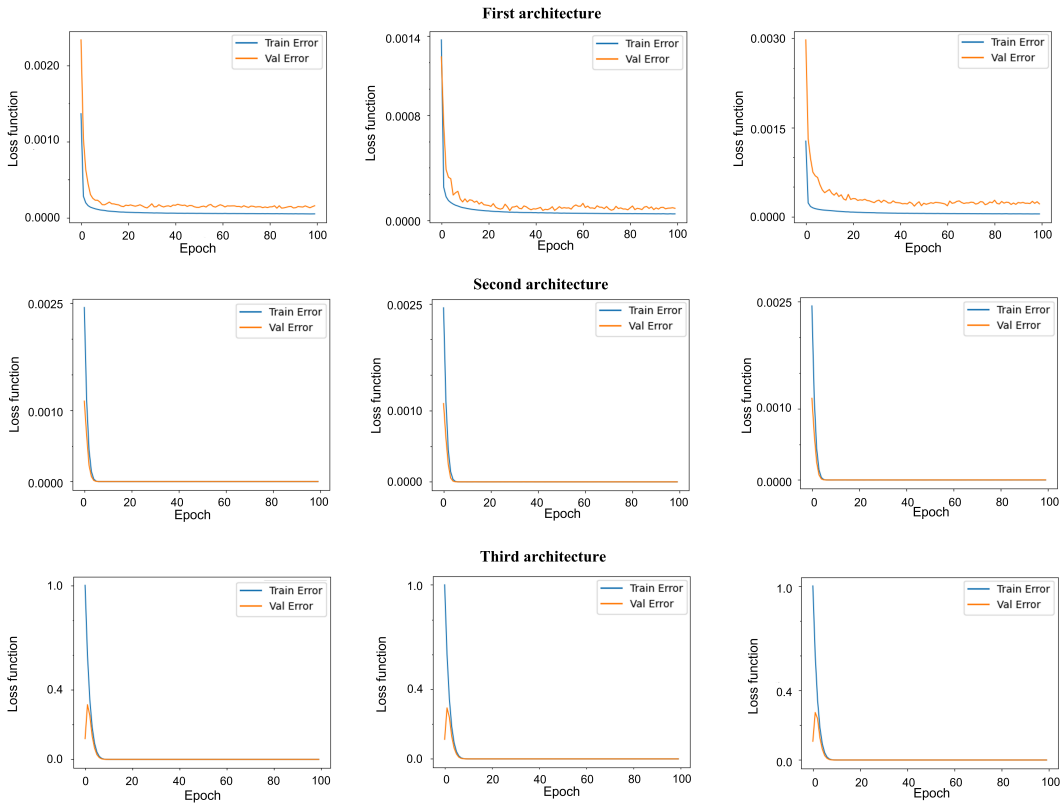

Figure 4.1: Comparison between the learning curves of the loss function for the first (left), second (center) and third distribution (right).

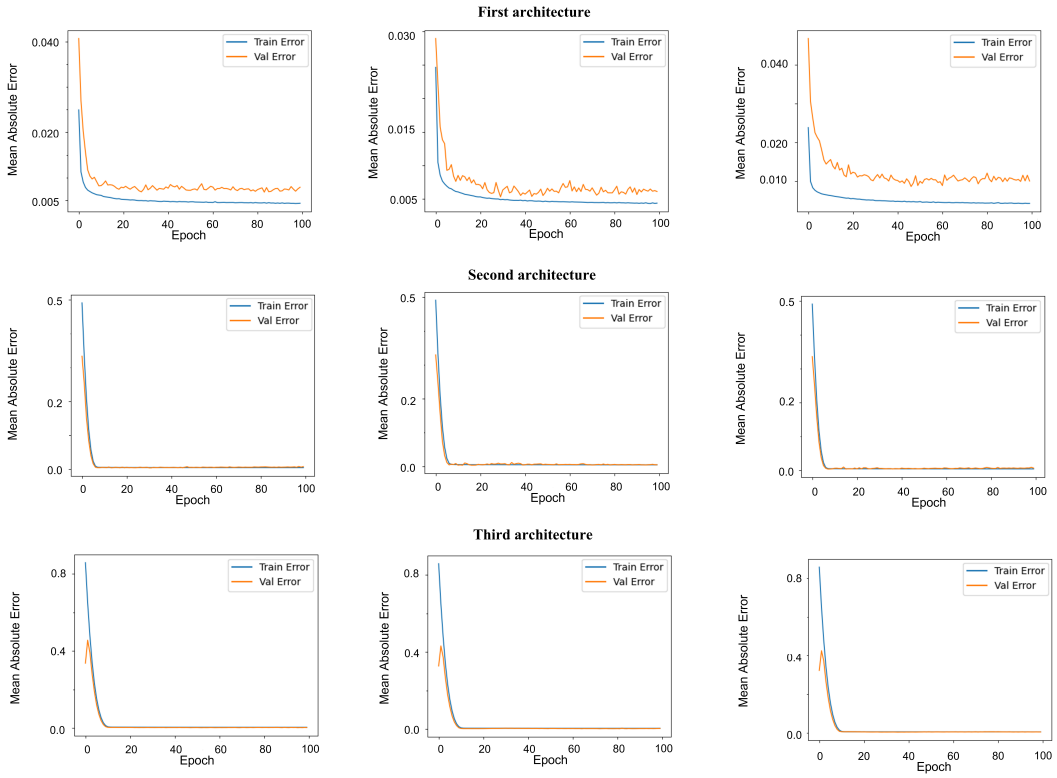

Figure 4.2: Comparison between the learning curves of the MAE for the first (left), second (center) and third distribution (right).

The learning curve from the first model, in every distribution, shows an evident overfitting, which is a reasonable response considering the simplicity of the architecture. On the other hand, the second and third curves, show a good fit behavior; although there is a peak at the beginning of the last one, corresponding to a moment in which it was more difficult for the model to learn from the dataset. However, the third architecture has shown a better performance by eliminating the slight peaks in the curve.

These graphs are the first indicators that more complex structures have shown a better response, anyways. To evaluate which one of these two options is better, it would be necessary to analyze the mean value of the correlation and the MAE.

4.2 Evaluation of the mean absolute errors and correlation

To quantitative evaluate the obtained predictions, the MAE and correlation are used as the indicative metrics. Considering that the main objective is to study algorithms whose implementation could improve the result obtained from the Tikhonov approach, the registered values between the Tikhonov and original images are used as a reference for comparison. Thus, both MAE and correlation (already described in 3.3 and 3.5) are evaluated for each prediction, considering the different models and data distributions. The following table presents the mean and the standard deviation of these metrics:

Distribution	*Images*	*MAE*	*Correlation*
	Tikhonov	0.020 ± 0.009	0.859 ± 0.342
First	Prediction from Model 1	0.015 ± 0.008	0.982 ± 0.386
	Prediction from Model 2	0.013 ± 0.007	0.989 ± 0.388
	Prediction from Model 3	0.011 ± 0.005	0.992 ± 0.389
Second	Prediction from Model 1	0.015 ± 0.008	0.982 ± 0.386
	Prediction from Model 2	0.015 ± 0.006	0.990 ± 0.389
	Prediction from Model 3	0.009 ± 0.004	0.993 ± 0.389
Third	Prediction from Model 1	0.015 ± 0.008	0.980 ± 0.385
	Prediction from Model 2	0.011 ± 0.006	0.989 ± 0.389
	Prediction from Model 3	0.010 ± 0.004	0.991 ± 0.389

Table 4.1: Comparison of the obtained MAE and correlation values for the first, second and third aleatory distribution of the training datasets.

The results show a constant decrement of the MAE and an increment in the correlation when the model gets more robust. To better visualize these differences between the architectures, a violin plot is presented, which is a statistical graph that can show the "entire distribution of data" [48]. In this case, the mean and median of the values are also represented:

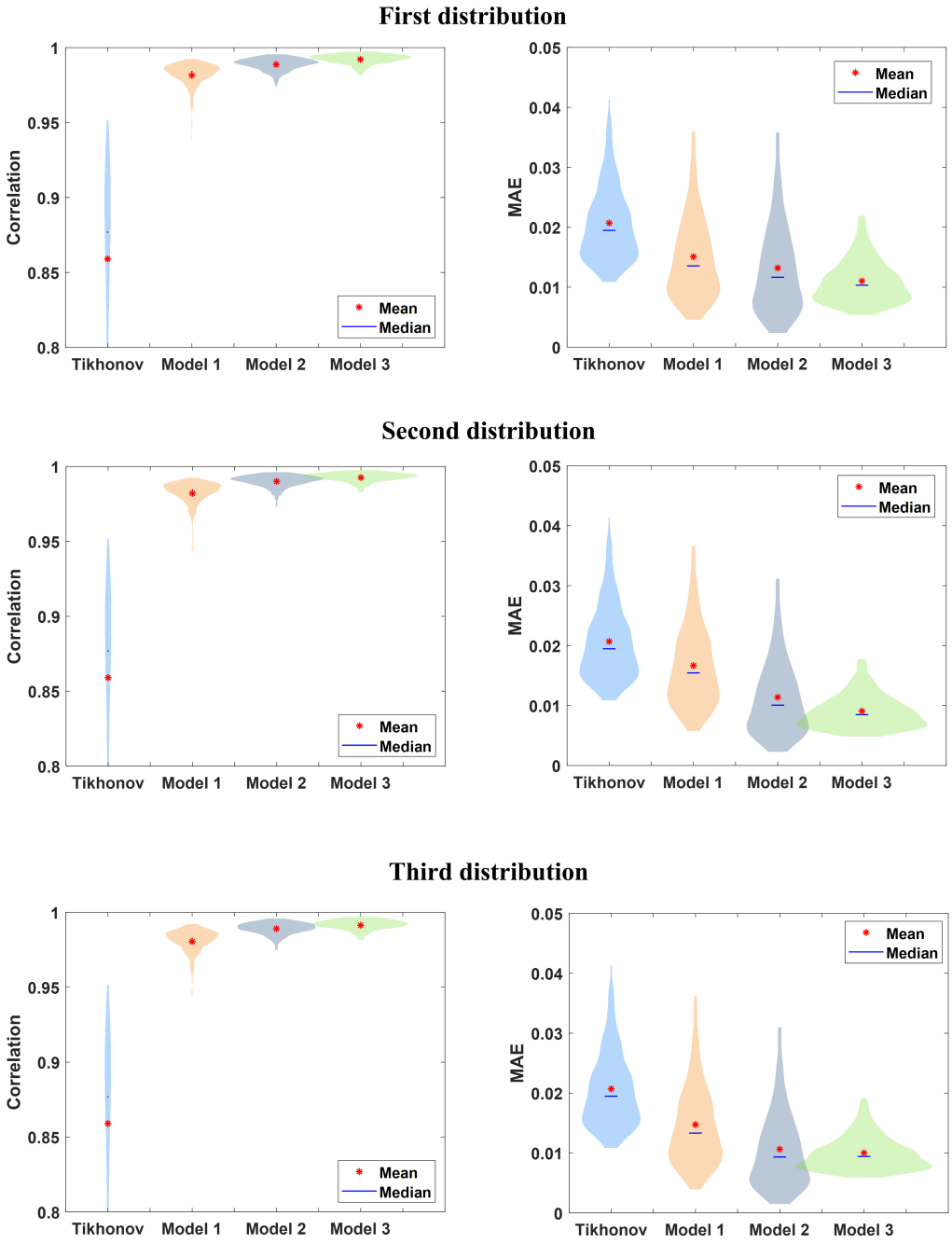

Figure 4.3: Comparison of the violin plots representing the correlation and MAE for Tikhonov and each architecture proposed.

Although the general performance is better than the reference (Tikhonov), the third model prevails as the best of the proposed ones. This one had a more complex architecture: including more Batch Normalization layers, Average Pooling instead of Max Pooling layers (which is used in the first architecture), and the use of skip connections to disconnect some layers and then connect them to deeper ones, leading to a further extraction of features.

On the other hand, the violin plots (4.3) have shown a small difference between the predictions from the three distributions of the training dataset. This could be occurring because the database does not have enough variety, which can be improved with data augmentation techniques or by increasing the number of generated cases. When creating the dipoles, a generator of random numbers from *MATLAB* was used, which probably ended up not being as aleatory as expected. Thus, to avoid a biased conclusion, the results with a "medium" response were selected, i.e., the ones obtained from the first distribution.

Considering that the proposed solution, which will be analyzed from now on, is based on the performance of the third architecture with the first training dataset distribution, it is relevant to mention how the results have been improved by reducing the MAE in a 46.7%, and increasing the correlation in a 15.5%, compared to the Tikhonov solution (4.1).

4.3 Representation of the predictions

A more visual way of studying the predictions is by comparing the images in 2D and 3D. From the 1.000 cases of the test dataset, three have been chosen: one of the best, average, and worst predictions (based on the correlation).

The following table (4.2) shows the MAE and correlation obtained for each case:

Selected case	***Images***	***MAE***	***Correlation***
Best case	Tikhonov	0.043 ± 0.009	0.909
	Prediction	0.028 ± 0.006	0.990
Average case	Tikhonov	0.027 ± 0.014	0.916
	Prediction	0.015 ± 0.007	0.995
Worst case	Tikhonov	0.028 ± 0.009	0.911
	Prediction	0.015 ± 0.005	0.984

Table 4.2: Comparison of the obtained MAE and correlation values for the three selected predictions.

The analysis of the 3D representations can be highly illustrative. To be precise, this could help confirm, or not, whether the shape of the ellipsoid is an important factor when reconstructing the image. The figures will be presented in their front and back view and will be compared to the analytic and Tikhonov ones:

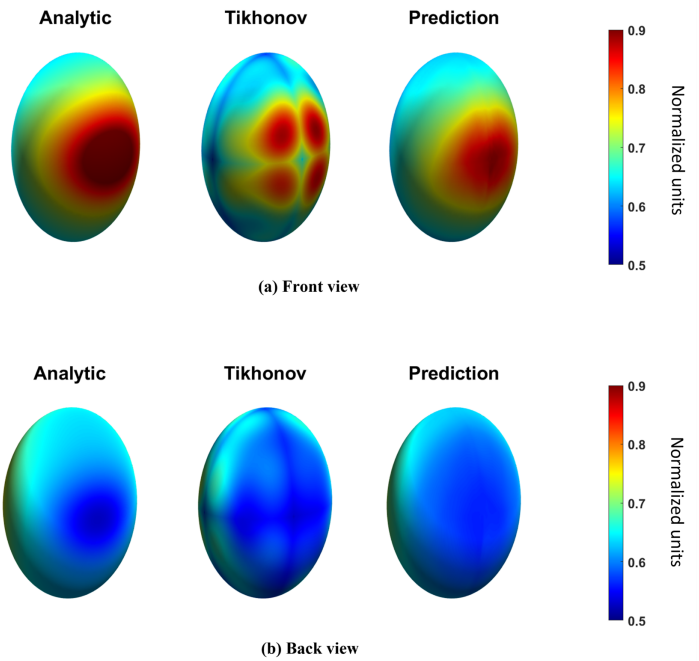

Figure 4.4: Comparison between analytic, Tikhonov and predicted 3D ellipsoids from the selected best case.

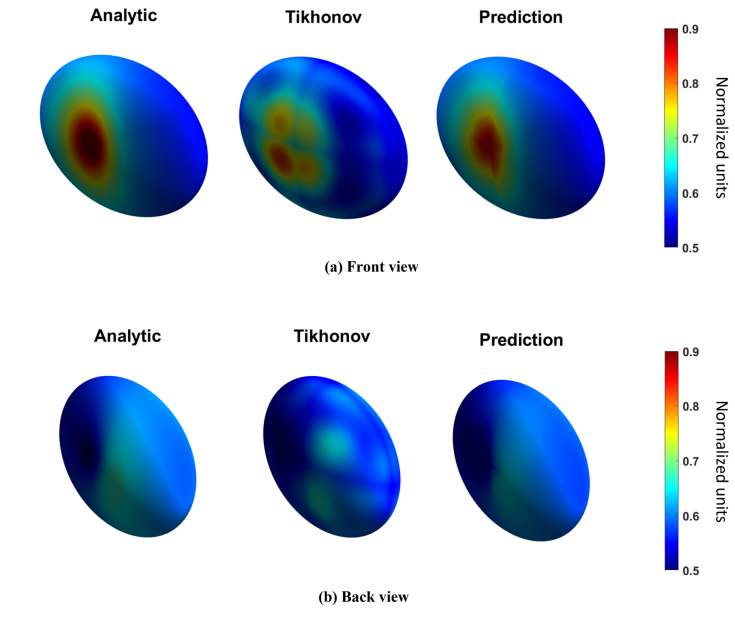

Figure 4.5: Comparison between analytic, Tikhonov and predicted 3D ellipsoids from the selected average case.

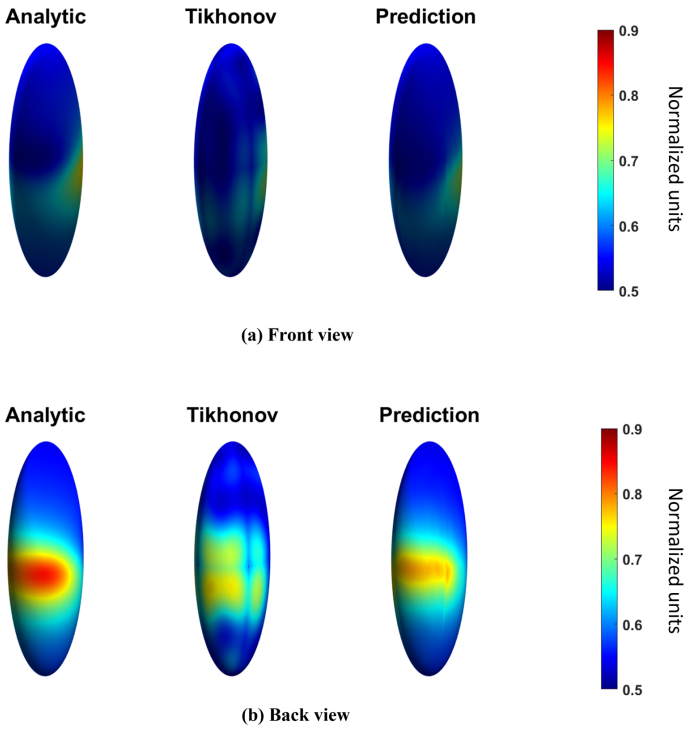

Figure 4.6: Comparison between analytic, Tikhonov and predicted 3D ellipsoids from the selected worst case.

The main objective is to visually evaluate how the neural network is performing with a variety of test cases. Thus, the figures compare the *Analytic* (original), *Tikhonov* and predicted images from the selected cases. Both predictions, from the best (4.4) and average cases, have a relatively "standard" ellipsoid shape, without extreme elongations. Nevertheless, the ellipsoids from the worst case (4.6) are far more deformed. Still, the prediction is better than the Tikhonov solution, with a 8.02% increment of the correlation and a reduction of the 48.43% in the MAE.

To better evaluate the reconstruction of the patterns and amplitudes of the predictions, the corresponding images (2D) of the presented ellipsoids are as follows:

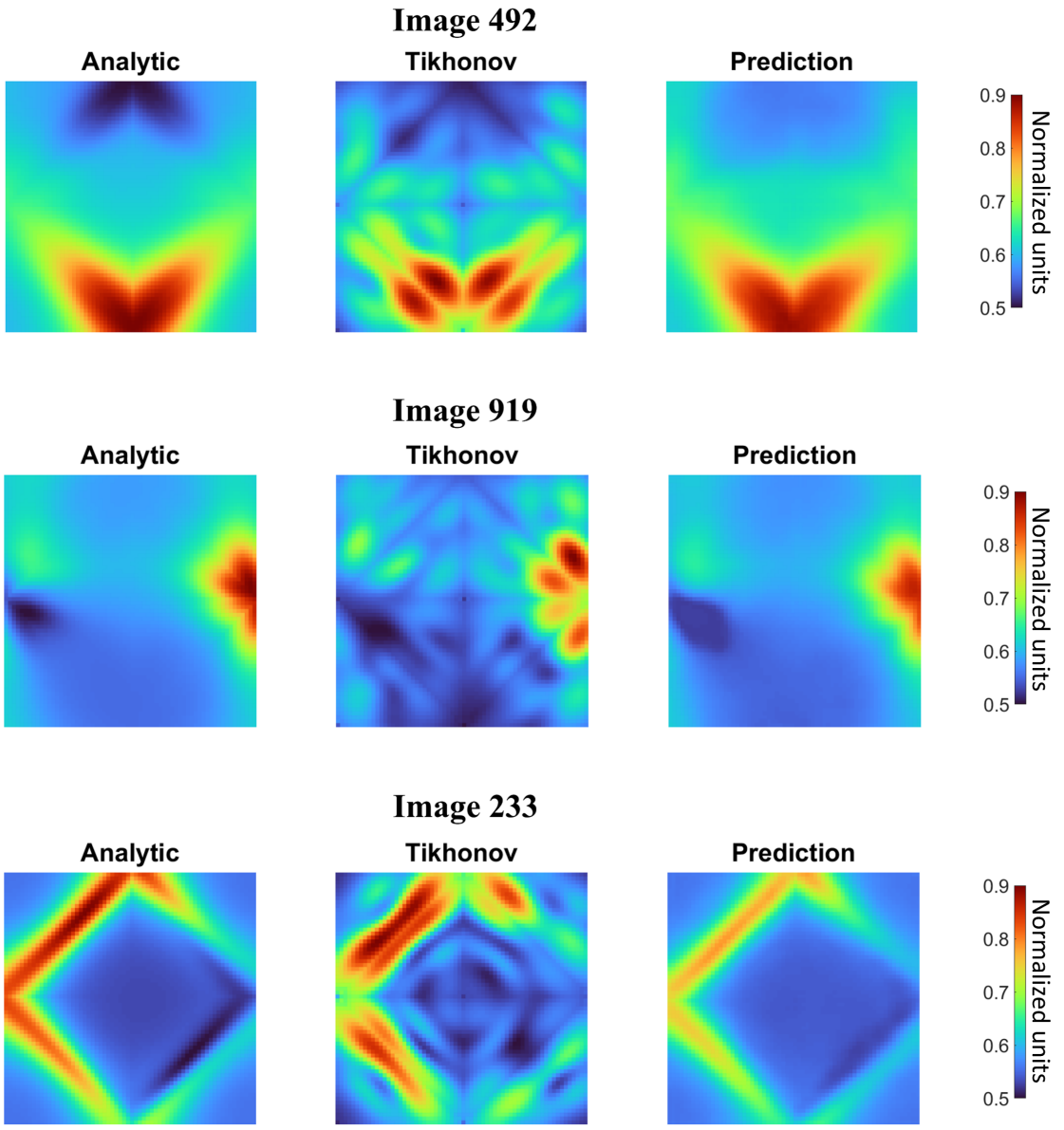

Figure 4.7: Comparison between analytic, Tikhonov and predicted 2D images for three different cases.

As a matter of fact, the NN is not only capable of reconstructing the pattern, but the amplitude as well, compared to the Tikhonov solution. A notorious characteristic in the Tikhonov images is the present geometric noise, which resembles a diamond shape. This could have potentially represented a problem for the network when denoising due to the difficulty of avoiding noise that looks like possible features. However, that was not the outcome of the predictions of the proposed neural network.

A good example of the previous statement is present in the image from the worst case, whose pat-

tern coincidentally follows the distribution of the noise. However, the model has acted as a useful tool to smooth the noise to the point it is almost unrecognizable in the predictions, without losing the essential information. This smoothness is also helpful to eliminate possible inconveniences with the generated meshes.

Another important aspect to consider is how well the Tikhonov images are, in contrast to the analytic ones. There is always a presence of noise, but the best and average cases have a more accurate amplitude representation, whereas the worst one is not that precise. Indeed, the predictions depend on the behavior of the Tikhonov solution, which can explain why the correlation for this case is worse.

Chapter 5

Discussion

As was previously presented, all deep learning architectures significantly improved the Tikhonov solution. To be precise, with the proposed network, a correlation of 0.99 and a MAE of 0.011 were obtained, which can be compared to the results from the inverse problem: a correlation of 0.86 and a MAE of 0.021 (4.1). Indeed, the use of denoising techniques has been proven as a good approach to upgrade the Tikhonov solution.

These results show how the use of denoising NNs can considerably improve the data obtained from regularization methods (precisely Tikhonov), which is not always as accurate as expected. Findings concerning the development of AI algorithms in inverse problems mostly focus on other applications, such as X-Rays or CT scans [49], which usually have more available resources and databases (something that is not common in cardiac research). Nevertheless, the focal point of a few studies has also been the evaluation of improving the current methods to increase the accuracy measured. The work of Chen et. al [50] is a good example of this. They achieve a correlation coefficient of around 0.74 by using a simple CNN to solve the ECGi with a database of real signals from pigs.

Other studies evaluate the performance of autoencoders for image denoising applications, such as Mao et. al [51], where symmetric skip connections are also evaluated and have shown how helpful they can be when cleaning the image (which has also improved the performance of this thesis proposed solution).

This proposal enables an a posteriori follow-up analysis of the extent to which the implementation of CNNs can improve the obtained results from the Tikhonov inverse problem. The study also revealed a data dependency when training, which directly affects the predictions; a matter that needs to be addressed, by using data augmentation techniques for a bigger database or by applying K-fold techniques to determine which combination of data is better for training, Moreover, the hyperparameters could be tuned to enhance the performance of the network as well.

Considering the high performance obtained, it would be interesting to widen the reach of this thesis, by evaluating the behavior of this type of NNs for data gotten from other regularization methods and datasets. In fact, this could lead to further research in the improvement of diagnosis methods of arrhythmias (ECGi), which is in growing need of modernization to guarantee a better treatment for patients.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this thesis, different deep learning algorithms have been implemented to develop denoising neural networks, with the purpose of evaluating the possibility of having a better behavior by post-processing the image that resulted from the Tikhonov regularization method.

First, a literature review about the inverse problem of electrocardiography theory, fundamentals of deep learning, and denoising as a sub-field of computer vision was done. A database was created with concentric ellipsoids, representing the heart and body surfaces, defined by the behavior of an aleatory dipole orientation. Later on, the type of NN used was the autoencoder, and the distribution of the data while training was also contemplated; considering three types of them.

The evaluation of the results has shown that the autoencoder could offer a remarkable performance when denoising the Tikhonov solution: increasing the correlation in a 15.5002% and reducing the MAE a 46.7168% (4.1). Besides, apart from the results of the metrics, the predictions were also represented in 2D and 3D, to clearly visualize the achieved upgrades.

Concluding, then, that autoencoders could be later used to denoise the images obtained from the ECGi could potentially represent a high improvement of the standard methods used currently.

The previous analysis led to the following specific conclusions:

- Using average pooling instead of max pooling layers made the prediction smoother, allowing it to eliminate more noise.
- The performance of the neural network improved when the architecture became more complex, i.e., when the number of layers increased.
- It is better to not only include batch normalization layers in between the structure, but at the end of it.
- The distribution of the training dataset is important, to the point it can alter the final performance of the neural network. This emphasizes the relevance of creating a bigger database with enough variety in its cases to mitigate the dependency on data.

- The architecture achieved a selective denoising, making it possible to detect and maintain patterns that could even be similar to the noise distribution.

6.2 Future work

Although this thesis was mainly focused on the implementation of denoising algorithms to upgrade the current results from the Tikhonov regularization, it could be extrapolated to every other generic inverse problem.

Following this idea, it is possible to consider the results obtained from a regularization method (not only referring to Tikhonov) as noisy data that can be refined with very simple denoising neural networks a posteriori. Moreover, the proposed solution of the thesis could be modified and applied to real ECGi data from patients.

Thus, future work concerns the publication of an article from this thesis in the Inverse Problem Journal, focusing on the idea of broadening its reach; considering the use of denoising algorithms as a post-processing technique for images acquired from regularization methods, which have been applied to solve inverse problems.

References

[1] K. Mc Namara, H. Alzubaidi, and J. K. Jackson. "Cardiovascular disease as a leading cause of death: how are pharmacists getting involved?" In: *Integrated Pharmacy Research and Practice* (2019), pp. 1–11. DOI: 10.2147/IPRP.S133088. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6366352/.

[2] R. J. Martis, U. R. Acharya, and H. Adeli. "Current methods in electrocardiogram characterization". In: *Computers in Biology and Medicine* 48 (2014), pp. 133–149. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2014.02.012. URL: https://www.sciencedirect.com/science/article/pii/S0010482514000432.

[3] IBM. *How is artificial intelligence used in medicine?* n.d. URL: https://www.ibm.com/topics/artificial-intelligence-healthcare#:~:text=Clinical%20operations%20and%20data%20managers%20executing%20clinical%20trials%20can%20use,amend%2C%20and%20manage%20clinical%20studies. (visited on 06/02/2023).

[4] MathWorks. *Deep Learning for Image Processing*. n.d. URL: https://www.mathworks.com/help/images/deep-learning.html?category=deep-learning&s_tid=CRUX_topnav (visited on 06/02/2023).

[5] M. Bertero, P. Boccacci, and C. De Mol. *Introduction to Inverse Problems in Imaging*. CRC Press, 2009, p. 2.

[6] *10 Lectures on Inverse Problems and Imaging*. 2023. URL: https://tristanvanleeuwen.github.io/IP_and_Im_Lectures/intro.html (visited on 06/06/2023).

[7] *Physics-embedded inverse analysis with algorithmic differentiation for the earth's subsurface*. 2023. URL: https://www.nature.com/articles/s41598-022-26898-1 (visited on 06/30/2023).

[8] J. Salinet et al. "Electrocardiographic Imaging for Atrial Fibrillation: A Perspective From Computer Models and Animal Experiments to Clinical Value". In: *Frontiers in Physiology* 12 (2021). ISSN: 1664-042X. URL: https://www.frontiersin.org/articles/10.3389/fphys.2021.653013 (visited on 06/30/2023).

[9] C. E. Chávez, F. Alonzo-Atienza, and D. Alvarez. "Avoiding the inverse crime in the Inverse Problem of electrocardiography: estimating the shape and location of cardiac ischemia". In: *Computing in Cardiology 2013* (2013), pp. 687–690. ISSN: 0213-9111. URL: https://ieeexplore.ieee.org/document/6713470.

[10] D. Farina. "Forward and inverse problems of electrocardiography : clinical investigations". PhD thesis. 2008. 196 pp. URL: https://publikationen.bibliothek.kit.edu/1000007590?.

[11] B. M. Horáček and J.C. Clements. "The inverse problem of electrocardiography: A solution in terms of single- and double-layer sources on the epicardial surface". In: *Mathematical Biosciences* 144.2 (1997), pp. 119–154. ISSN: 0025-5564. URL: https://www.sciencedirect.com/science/article/pii/S0025556497000242.

[12] D. Calvetti et al. "Tikhonov regularization and the L-curve for large discrete ill-posed problems". In: *Journal of Computational and Applied Mathematics* 123.1 (2000). Numerical Analysis 2000. Vol. III: Linear Algebra, pp. 423–446. ISSN: 0377-0427. DOI: https://doi.org/10.1016/S0377-0427(00)00414-3. URL: https://www.sciencedirect.com/science/article/pii/S0377042700004143.

[13] Instituto Nacional de Estadística. *Top 15 causes of death in Spain*. 2021. URL: https://public.tableau.com/views/DCM_en1/trial?:showVizHome=no&:embed=true (visited on 04/27/2023).

[14] J. Jalife et al. *Basic cardiac electrocardiography for the clinician*. Wiley-Blackwell Publishing, 2009, pp. 12–13. URL: https://books.google.es/books?hl=es&lr=&id=Ypwkqcyll3cC&oi=fnd&pg=PA2&dq=cardiac+electrophysiology&ots=nWJlt0YQuI&sig=kEs15ua2sgYlADk0bpByZAbXbq4#v=onepage&q&f=false.

[15] J. Gupta and M. Shea. *Biology of the Heart*. 2022. URL: https://www.msdmanuals.com/home/heart-and-blood-vessel-disorders/biology-of-the-heart-and-blood-vessels/biology-of-the-heart (visited on 05/22/2023).

[16] X. Wei, S. Yohannan, and J. R. Richards. *Physiology, Cardiac Repolarization Dispersion and Reserve*. 2022. URL: https://www.ncbi.nlm.nih.gov/books/NBK537194/ (visited on 05/22/2023).

[17] M. Josephson. *Clinical Cardiac Electrophysiology: Techniques and Interpretations*. Lippincott Williams and Wilkins, 2008, pp. 2–3.

[18] P. Madona, R. I. Basti, and M. M. Zain. "PQRST wave detection on ECG signals". In: *Gaceta Sanitaria* 35 (2021). The 3rd International Nursing and Health Sciences Students and Health Care Professionals Conference (INHSP), S364–S369. ISSN: 0213-9111. DOI: https://doi.org/10.1016/j.gaceta.2021.10.052. URL: https://www.sciencedirect.com/science/article/pii/S0213911121002466.

[19] RN. Ghanem et al. "Heart-surface reconstruction and ECG electrodes localization using fluoroscopy, epipolar geometry and stereovision: application to noninvasive imaging of cardiac electrical activity". In: *IEEE Trans Med Imaging* (2003), pp. 1307–18. DOI: 10.1109/TMI.2003.818263. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2034496/.

[20] D. B. Geselowitz. "On the theory of the electrocardiogram". In: *roceedings of the IEEE* 77.6 (1989), pp. 857–876. DOI: 10.1109/5.29327. URL: https://ieeexplore.ieee.org/document/29327.

[21] F. H. Fenton and E. M. Cherry and. *Cardiac arrhythmia*. 2008. URL: http://www.scholarpedia.org/article/Cardiac_arrhythmia (visited on 05/17/2023).

[22] R. J. Ghanem et al. "Noninvasive Electrocardiographic Imaging (ECGI): Comparison to intraoperative mapping in patients". In: *Heart Rhythm* 2.4 (2005), pp. 339–354. ISSN: 1547-5271. DOI: https://doi.org/10.1016/j.hrthm.2004.12.022. URL: https://www.sciencedirect.com/science/article/pii/S1547527105000056.

[23] R. S. MacLeod, B. Taccardi, and R. L. Lux. “Electrocardiographic mapping in a realistic torso tank preparation”. In: *Proceedings of 17th International Conference of the Engineering in Medicine and Biology Society* 1 (1995), pp. 245–246. DOI: 10.1109/IEMBS.1995.575092. URL: https://ieeexplore.ieee.org/document/575092.

[24] B. J. Messinger-Rapport and Y. Rudy. “The Inverse Problem in Electrocardiography: A Model Study of the Effects of Geometry and Conductivity Parameters on the Reconstruction of Epicardial Potentials”. In: *IEEE Transactions on Biomedical Engineering* BME-33.7 (1986), pp. 667–676. DOI: 10.1109/TBME.1986.325756. URL: https://ieeexplore.ieee.org/document/4122363.

[25] R. M. Gurlrajani. “The forward and inverse problems of electrocardiography”. In: *IIEEE Engineering in Medicine and Biology Magazine* 17.3 (1998), pp. 84–101. DOI: 10.1109/51.715491. URL: https://ieeexplore.ieee.org/abstract/document/715491.

[26] J. M. Norman and D. H. Hook. *Newell, Simon and Shaw Develop the First Artificial Intelligence Program*. 2023. URL: https://www.historyofinformation.com/detail.php?id=742 (visited on 05/17/2023).

[27] *AI Winter: The Highs and Lows of Artificial Intelligence*. 2021. URL: https://www.historyofdatascience.com/ai-winter-the-highs-and-lows-of-artificial-intelligence/ (visited on 05/17/2023).

[28] P. van der Made. *The Future Of Artificial Intelligence*. 2023. URL: https://www.forbes.com/sites/forbestechcouncil/2023/04/10/the-future-of-artificial-intelligence/ (visited on 05/17/2023).

[29] Z. H. Zhou. *Machine Learning*. Trans. by S. W. Liu. Springer Singapore, 2016, p. 2. DOI: https://doi.org/10.1007/978-981-15-1967-3. URL: https://link.springer.com/book/10.1007/978-981-15-1967-3.

[30] Y. LeCun, Y. Bengio, and G. Hinton. “Deep Learning”. In: *Nature* 521 (2015), pp. 436–444. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539#citeas.

[31] J. Torres. *Deep Learning, Introducción práctica con Keras (Primera Parte)*. Kindle Direct Publishing, 2018, pp. 77–83.

[32] L. Hardesty. *Explained: Neural networks*. 2017. URL: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414 (visited on 05/16/2023).

[33] T. Wood. *Backpropagation*. 2020. URL: https://deepai.org/machine-learning-glossary-and-terms/backpropagation (visited on 05/16/2023).

[34] B. Kröse and P. van der Smagt. *An introduction to Neural Networks*. The University of Amsterdam, 1996, p. 15.

[35] S. Sharma. *Activation Functions in Neural Networks*. 2017. URL: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (visited on 05/16/2023).

[36] G. Nanos. *Computer vision: Differences between Low-Level and High-Level Features*. 2023. URL: https://www.baeldung.com/cs/cv-low-vs-high-level-features (visited on 05/17/2023).

[37] Z. Li et al. “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2022), pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827. URL: https://ieeexplore.ieee.org/document/9451544.

[38] R. Qayyum. *Introduction To Pooling Layers In CNN*. 2022. URL: https://towardsai.net/p/l/introduction-to-pooling-layers-in-cnn (visited on 05/18/2023).

[39] J. Brownlee. *How to use the UpSampling2D and Conv2DTranspose Layers in Keras*. 2019. URL: https://machinelearningmastery.com/upsampling-and-transpose-convolution-layers-for-generative-adversarial-networks/ (visited on 05/18/2023).

[40] Keras Team. *Keras Documentation: BatchNormalization layer*. n.d. URL: https://keras.io/api/layers/normalization_layers/batch_normalization/ (visited on 05/19/2023).

[41] J. Brownlee. *How to use Learning Curves to Diagnose Machine Learning Model Performance*. 2019. URL: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/ (visited on 05/16/2023).

[42] S. Solomon. *Image Denoising using Deep Learning*. 2021. URL: https://medium.com/analytics-vidhya/image-denoising-using-deep-learning-dc2b19a3fd54 (visited on 05/16/2023).

[43] D. Sharma. *Top 5 Interview Questions on Autoencoders*. 2022. URL: https://www.analyticsvidhya.com/blog/2022/11/top-5-interview-questions-on-autoencoders/ (visited on 05/19/2023).

[44] *Electric Dipole: Introduction, Dipole Moment, Formulas, Videos, Examples*. 2020. URL: https://www.toppr.com/guides/physics/electric-charges-and-fields/electric-dipole/ (visited on 06/06/2023).

[45] D. J. Griffiths. *Introduction to Electrodynamics*. 4th ed. Cambridge University Press, 2017, p. 456. DOI: 10.1017/9781108333511. URL: https://www.cambridge.org/highereducation/books/introduction-to-electrodynamics/3AB220820DBB628E5A43D52C4B011ED4#overview.

[46] F. Chollet. *Building Autoencoders in Keras*. 2016. URL: https://blog.keras.io/building-autoencoders-in-keras.html (visited on 06/06/2023).

[47] Mathworks. *Correlation Coefficients - MATLAB corrcoef*. n.d. URL: https://es.mathworks.com/help/matlab/ref/corrcoef.html (visited on 06/06/2023).

[48] E. Lewinson. *What Are Violin Plots and How Do You Use Them?* 2023. URL: https://builtin.com/data-science/violin-plots (visited on 06/19/2023).

[49] M. T. McCann, K. H. Jin, and M. Unser. “Convolutional Neural Networks for Inverse Problems in Imaging: A Review”. In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 85–95. DOI: 10.1109/MSP.2017.2739299. URL: https://ieeexplore.ieee.org/document/8103129.

[50] K. W. Chen, L. Bear, and C. W. Lin. “Solving Inverse Electrocardiographic Mapping Using Machine Learning and Deep Learning Frameworks”. In: *Sensors* 22.6 (2022). ISSN: 1424-8220. DOI: 10.3390/s22062331. URL: https://www.mdpi.com/1424-8220/22/6/2331.

[51] X. J. Mao, C. Shen, and Yang Y. B. “Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections”. In: *CoRR* abs/1606.08921 (2022). URL: http://arxiv.org/abs/1606.08921.

Part II

Appendices

Appendix A

Detailed structure of the network

The following figures (A.1, A.2) illustrate the complete architecture of the third model proposed (in sub-section 3.2.3), including the type of layer, and the dimensions of the input and output for each one of them.

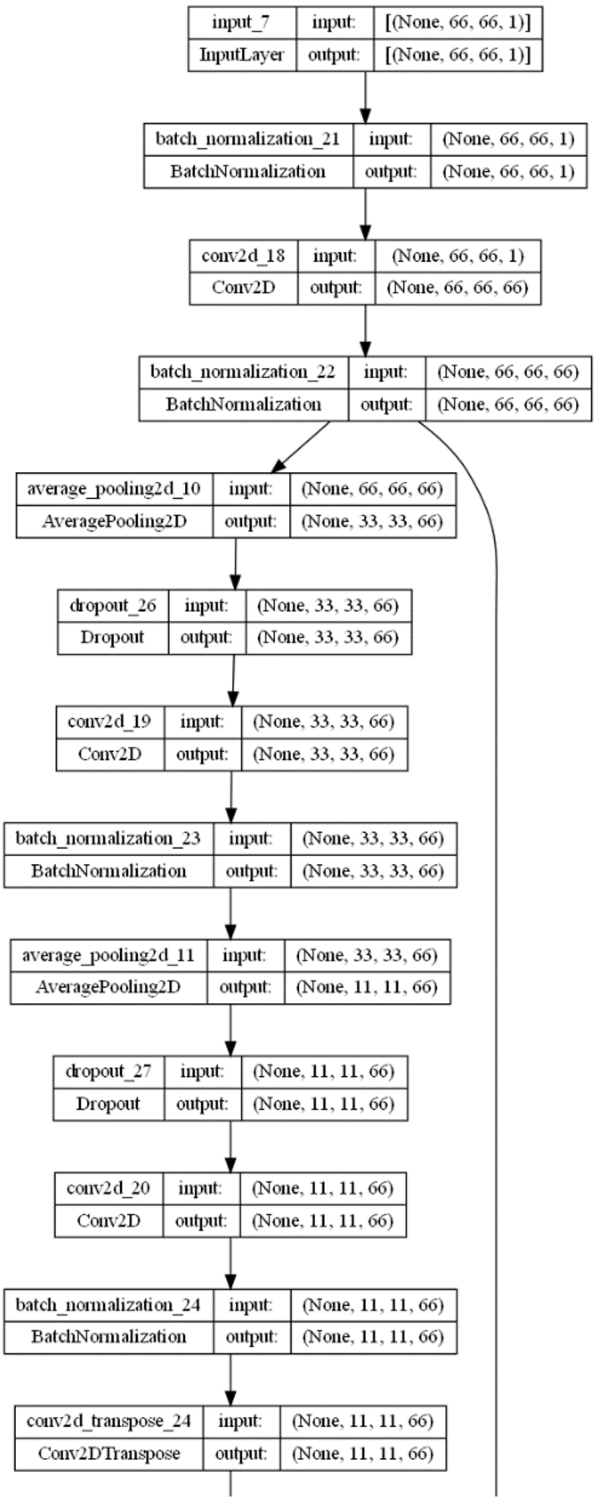

Figure A.1: First part of the architecture.

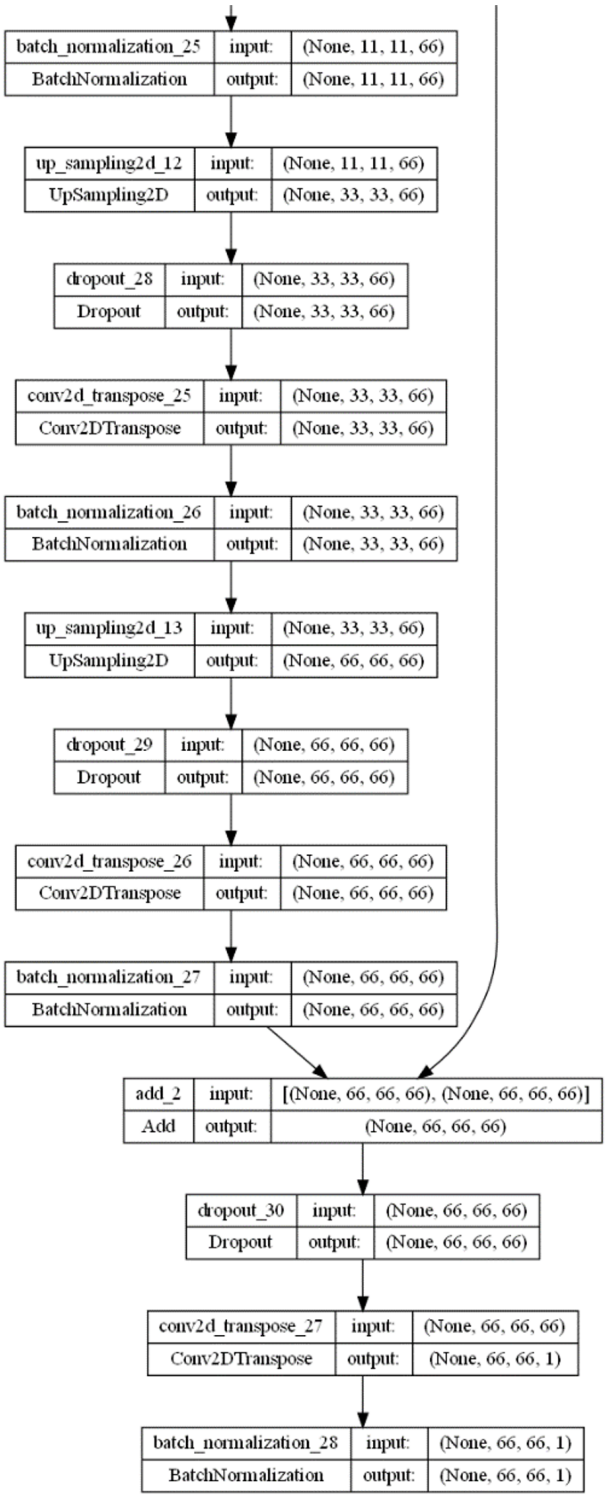

Figure A.2: Second part of the architecture.