FEDERAL UNIVERSITY OF ALAGOAS

COMPUTING INSTITUTE

MASTER'S IN INFORMATICS PROGRAM

MASTER THESIS

CLASSIFYING HEARTBEATS FROM ELECTROCARDIOGRAM SIGNALS USING A SIAMESE CONVOLUTIONAL NEURAL NETWORK

MASTER'S STUDENT

EDUARDO MORAES DE MIRANDA VASCONCELLOS

ADVISOR

THIAGO DAMASCENO CORDEIRO, DR.

MACEIÓ, AL

FEBRUARY - 2022

EDUARDO MORAES DE MIRANDA VASCONCELLOS

CLASSIFYING HEARTBEATS FROM ELECTROCARDIOGRAM SIGNALS USING A SIAMESE CONVOLUTIONAL NEURAL NETWORK

Master thesis presented as a partial requirement to obtain a Master Degree by the Master's in Informatics Program of the Computing institute at Federal University Of Alagoas
Advisor: Thiago Damasceno Cordeiro, Dr.

MACEIÓ, AL
FEBRUARY - 2022

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

V331c Vasconcellos, Eduardo Moraes de Miranda.
Classifying heartbeats from electrocardiogram signals using a siamese convolutional neural network / Eduardo Moraes de Miranda Vasconcellos. – 2022.
34 f. : il.

Orientador: Thiago Damasceno Cordeiro.
Dissertação (mestrado em informática) - Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2022.
Texto em inglês.

Bibliografia: f. 29-34.

1. Eletrocardiografia. 2. Aprendizagem de máquina. 3. Few Shot Learning. 4. Redes neurais de computação. 5. Batimento cardíaco - Classificação. I. Título.

CDU: 004.81:159.953.5:616.12-073.97

UNIVERSIDADE FEDERAL DE ALAGOAS
UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Informática – PPGI
Instituto de Computação/UFAL
Campus A. C. Simões BR 104-Norte Km 14 BL 12 Tabuleiro do Martins
Maceió/AL - Brasil CEP: 57.072-970 | Telefone: (082) 3214-1401

Folha de Aprovação

EDUARDO MORAES DE MIRANDA VASCONCELLOS

CLASSIFICAÇÃO DOS BATIMENTOS CARDÍACOS A PARTIR DE SINAIS DE ELETROCARDIOGRAMA USANDO UMA REDE NEURAL CONVOLUCIONAL SIAMESA

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas e aprovada em 24 de fevereiro de 2022.

Banca Examinadora:

Documento assinado digitalmente
gov.br
Thiago Damasceno Cordeiro
Data: 05/03/2022 00:35:18-0300
Verifique em https://verificador.iti.br

Prof. Dr. THIAGO DAMASCENO CORDEIRO
UFAL – Instituto de Computação
Orientador

Prof. Dr. BALDOINO FONSECA DOS SANTOS NETO
UFAL – Instituto de Computação
Examinador Interno

Documento assinado digitalmente
gov.br
FILIPE ROLIM CORDEIRO
Data: 05/03/2022 13:06:59-0300
Verifique em https://verificador.iti.br

Prof. Dr. FILIPE ROLIM CORDEIRO
UFRPE – Universidade Federal Rural de Pernambuco
Examinador Externo

ABSTRACT

The Electrocardiogram (ECG) is a low-cost exam commonly used to diagnose abnormalities in the cardiac cycle such as arrhythmias and problems in the heart's muscle. With the advance of machine learning (ML) techniques in recent years, the automatic classification of ECG signals garnered interest in the scientific community. However, the process of annotating large and diverse datasets to support the training of ML techniques is still very time-consuming and error-prone. Thus, ML techniques whose training does not require a large, well-annotated datasets are becoming even more prominent. This means that underrepresented data in ECG datasets, like rare cardiologic disturbs can still be properly identified and classified. In this work, the use of Siamese Convolutional Neural Networks, popular in imaging classification problems, to classify 12-Lead ECG heartbeats is investigated. The early results indicate accuracy of up to 95% in a public dataset by using models composed of different combinations of similarity and loss functions. The class by class classification results are also compared with those of similar methods found in the literature, obtaining metrics on par and even exceeding them in the classification of some classes.

Keywords: Electrocardiogram, Machine Learning, Few-Shot Learning, Siamese Neural Networks, Heartbeat Classification

RESUMO

O Eletrocardiograma (ECG) é um exame de baixo custo comumente usado para diagnosticar anormalidades no ciclo cardíaco, tais como arritmias e problemas no músculo do coração. Com o avanço das técnicas de aprendizagem de máquinas (ML) nos últimos anos, a classificação automática de ECG está obtendo um interesse crescente na comunidade científica. Entretanto, o processo de anotar grandes e diversos conjuntos de dados para serem usados no treinamento de técnicas ML ainda é muito demorado e propenso a erros. Assim, técnicas ML cujo treinamento não requer um grande e bem anotado conjunto de dados estão se tornando cada vez mais proeminentes. Isto significa que os dados subrepresentados nos conjuntos de dados ECG, como raros distúrbios cardiológicos, ainda podem ser devidamente identificados e classificados. Neste trabalho, é investigado o uso de Redes Neurais Convolucionais Siamêsas, populares em problemas de classificação de imagens, para classificar batimentos cardíacos de 12 derivações em sinais de ECG. Os primeiros resultados indicam uma precisão de até 95% em um conjunto de dados públicos, utilizando modelos compostos de diferentes combinações de funções de similaridade e perda. Os resultados da classificação classe por classe também são comparados com os de métodos similares encontrados na literatura, obtendo-se métricas ao par e até mesmo excedendo-as na classificação de algumas classes.

Palavras-Chave: Eletrocardiograma, Aprendizagem de Máquina, *Few Shot Learning*, Redes Neurais Siamesas, Classificação dos Batimentos Cardíacos

List of Figures

List of Tables

Acronyms

ANN Artificial neural network.

CNN Convolutional Neural Network.

DWT DiscreteWavelet Transform.

ECG Electrocardiogram.

INCART St. Petersburg Institute of CardiologicalTechnics 12-lead Arrhythmia.

ML Machine Learning.

MSE Mean Squared Error.

NIH National Institutes of Health.

ReLU Rectified Linear Unit.

RMSE Root Mean Squared Error.

RTMG Telehealth Network of Minas Gerais.

SCNN Siamese Convolutional Neural Network.

SNN Siamese Neural Network.

Contents

1 Introduction

The healthcare system in Brazil has many deficiencies due to low investment and poor distribution of doctors among the country's regions. According to the latest medical, demographic survey [RSS18], there is a ratio of 2.18 doctors per 1,000 inhabitants in the national territory. However, the northeastern region has a ratio of only 1.42 doctors per 1,000 inhabitants. According to a 2019 study by the Association of American Medical Colleges, there is a ratio of 353 people per physician in the United States, and only 2.4% are specialists in cardiology [AAM20].

Cardiovascular diseases are the most common cause of death in the world [Org20]. In Brazil, they represent the leading cause of disability retirement and hospitalization expenses. However, only 4.1% of medical specialists in Brazil are cardiologists, and this scarcity compromises the analysis simple tests such as the electrocardiogram (ECG) [RSS18]. Moreover, regular visits to a cardiologist can help reverse this situation since cardiovascular diseases could be diagnosed prematurely through ECG tracings, avoiding stroke and heart attack complications.

The rest ECG is a simple, non-invasive, and inexpensive test that records the heart's electrical activity over a short period (approximately 10 seconds). The recording can be done by 12 leads, combining the position of electrodes located in the limb region and on the front of the chest. The differences in shapes and frequency of the ECG waves allow identifying different cardiovascular diseases such as cardiac arrhythmias or heart muscle problems.

Aiming to speed up the triage process in medical centers that perform remote ECG reports, researchers have been developing a set of computational algorithms to automatically classify ECG signals as to the state of normality or abnormality in cardiac electrical activity. In the literature, several papers explore deep learning techniques to classify ECG signals from digital tracings. For example, Acharya *et al.* [AFL$^+$17] trained two eleven-layer convolutional neural networks (CNN) to classify ECG signals as normal or with coronary artery disease. In one of the networks, 95,300 2-second segments, 15,300 normal, and 80,000 altered; and in the other, 38,120 5-second segments, 6,120 normal and 37,000 altered were used for training. All signals were obtained by lead II of 40 normal patients from the Fantasia [IPM$^+$96] database. In addition to these, seven more records of patients with coronary artery disease from the St. Petersburg Institute of Cardiology Technics 12-lead arrhythmia database [GAG$^+$00] were used. The two networks had the same structure but were trained with segments of different lengths. In this study, the accuracy obtained was 95% with two-second samples and 95.1% with five-second samples.

Using a 10-layer CNN, Baloglu *et al.* [BTY$^+$19] was able to detect 10 different classes of myocardial infarction from 12-lead signals found in the PTB Diagnostic ECG [GAG$^+$00] database. A total of 148 signals with myocardial infarction and 42 healthy signals were used. The signals went through a wavelet transform-based pre-processing step for noise and baseline wander removal and then through an R-wave detector to extract a stretch of the ECG signal that corresponds to only one heartbeat. In this approach, each lead was trained separately on the neural network, resulting in average accuracy of 99.60%.

Yildirim *et al.* [YPTA18], on the other hand, used a different approach to detect 17 classes of cardiac arrhythmias. For training, 1000 10-second segments sampled from signals of 45 individuals from the MIT-BIH Arrhythmia database [GAG$^+$00] were used. This work follows the hypothesis that there is only one type of arrhythmia in each 10-second segment and uses longer traces to capture changes in signal characteristics over time. The CNN classifier developed in this work obtained an overall accuracy of 91.33%.

Ribeiro *et al.* [RRP$^+$20] used a residual neural network model with 12-lead ECG signals to identify six types of cardiac disorders: first-degree atrioventricular block, right bundle branch block, left bundle branch block, sinus bradycardia, atrial fibrillation, and sinus tachycardia. In this work, the authors used a private database obtained through the Telehealth Network of Minas Gerais (RTMG), containing more than 2 million and 300 thousand 10-second segments of ECG signals. The signal classes were obtained from medical reports using natural language processing techniques. In the study, the trained network's diagnosis was compared with the diagnosis given by pairs formed as follows: two cardiology residents, two emergency department residents, and two medical students. The network obtained a more consistent result than the results provided by all the pairs, with the F1 score being 80% and specificity above 99%.

Despite the high accuracy, most of these works resort to public databases for classifier training, such as the MIT-BIH Arrhythmia [MM01] and the St. Petersburg Institute of Cardiological Technics 12-lead Arrhythmia (INCART), available in the PhysioNet [GAG$^+$00] repository. Public databases, however, usually contain long signals from few patients, which implies a strong dependency between observations. This fact is not considered in accuracy calculations and contributes negatively to the fact that such measures tend to be too optimistic [JD18]. In addition, few bases make available the resting ECG signals in 12 leads, making it difficult to detect diseases whose diagnosis depends on the evaluation of signals in multiple leads, such as ventricular fibrillation and myocardial infarction [GGS17]. Another problem arises because more severe diseases tend to occur less frequently, thus having little representativeness in databases with few patients.

Considered an open problem in the context of deep learning with ECG signals [HZS$^+$20], class inbalancing is said to be an obstacle in developing effective deep learning models with a high amount of parameters by making the training phase harder. This problem is generally avoided using data augmentation techniques. Recently, a new approach called few-shot learning [WYKN20] has been popularized and is a standout on imaging processing problems. This approach tries to circumvent the necessity of large and diverse datasets by using prior knowledge to improve the models' convergence to an acceptable solution. The prior knowledge can be used in mainly three ways: augmenting the training dataset, restricting the solution search space, and modifying a similar task solution to fit the new problem.

This approach recently found its way into the classification of ECG signals. For example, Liu *et al.* [LYFW21] developed a few shot learning methods to detect arrhythmia in ECG signals by pre-training a model on an auxiliary dataset and using a meta-transfer learning scheme to improve the learning of the unseen classes. In Yang emphet al. [YWLD21] a Siamese Neural Network (SNN) based on the ODENet was used to classify 10 seconds segments of ECG signals into five classes. A paper similar to this work was published by Li *et al.* [LWL21]. There, a Siamese Convolution Neural Network (SCNN) was proposed to classify single lead ECG heartbeats into four classes under a limited dataset constraint.

1.1 General Objective

This work aims to develop a Siamese Convolutional Neural Network Model for the classification of heartbeats from digital tracings of ECG signals containing 12 leads in inbalanced datasets.

1.1.1 Specific Objective

To achieve the general objective of this work, the following specific objectives were contemplated:

- Search and selection of ECG signals public databases with data containing several cardiac disturbances.
- Research of filtering techniques to realize a preprocessing step in order to remove noise.
- Definition of a Siamese Neural Network architecture for 1D signal processing.
- Study of different Loss and Similarity functions.
- Validation and comparison of the trained models.
- Analysis of model results.

1.2 Document Structure

This document is split into five chapters. In chapter 2, a theoretical framework with the main themes shown in this work will be showed: ECG Signals, ECG Datasets, Multilayer Neural Networks, Loss Functions, Convolutional Neural Networks, Few-Shot Learning, and Siamese Neural Networks. In chapter 3, the proposed methodology contains data Preprocessing, model architecture, and decision process description. Chapter 4 includes the achieved results, with a discussion about the loss function/similarity function combination; and a comparison with other models found in the literature. The conclusion can be found in chapter 5, with possible next steps in researching the use of SCNN in the classification of ECG signals.

2 Theoretical framework

2.1 Electrocardiogram Signals

The electrocardiogram (ECG) signal is a time-voltage graph that represents the heart's electrical activity from combinations of different electrodes called leads. It is one of the main tests used in the diagnosis of heart disease identifying comorbidities such as myocardial infarction and ischemia, arrhythmias, and cardiomyopathies [GGS17].

A standard ECG signal is composed of five main entities: the P wave, which represents atrial depolarization; the QRS complex, which represents ventricular depolarization; the ST segment and T wave, which represents ventricular repolarization; and the U wave, which represents the final phase of ventricular repolarization (1). In studies of electrocardiography, it is common to analyze the QRS complex from the Q, R, and S waves that compose it, and that the existence of the U wave is ignored because it has a minimal amplitude, often being imperceptible in most examinations [GGS17].

In addition to the five entities, the ECG signal is interpreted through different segments and intervals. A segment is defined as the section between the end of one wave and the beginning of another. An interval is defined as a section that partitions the ECG to include at least one whole wave. There are three primary segments: PR, ST, and TP. The PR and ST segments represent the process of atrial and ventricular repolarization, respectively, and the TP segment represents a resting state between beats. It is generally used as a reference in the analysis of the other two segments. Four intervals are routinely measured: PR, QRS, QT, and RR, with the RR interval generally used to calculate instantaneous heart rate. The main characteristics of the ECG signal can also be seen in Figure 1.

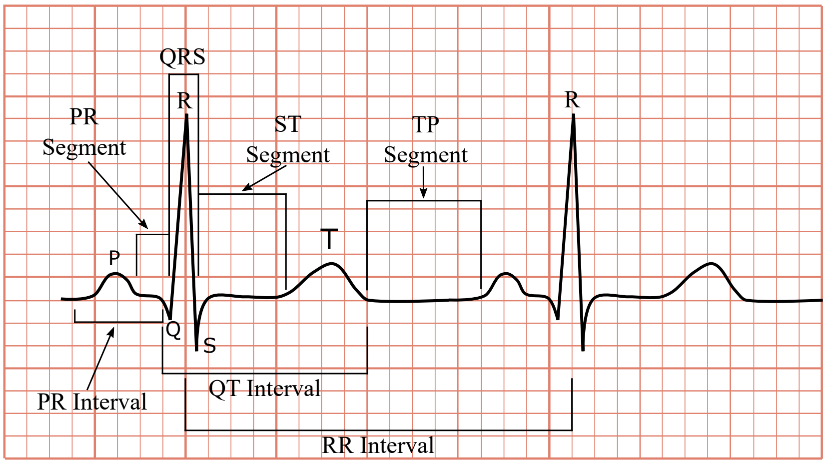

Figure 1: Main components, segments and intervals of an ECG signal.

The 12 standard leads can be divided into six peripheral leads and six precordial leads. The peripheral leads (I, II, III, aVR, aVL, aVF) are obtained employing electrodes placed on the limbs. The precordial leads (V1, V2, V3, V4, V5, V6) are obtained utilizing electrodes placed on the anterior thorax, especially on the precordium. Since the leads are positioned in different regions of the body, they record the heart activity with varying fields of view (Fig. 2). Peripheral leads provide a view from the frontal plane of the body, and precordial leads provide a view from a horizontal plane to the body. Together, they can provide a 3D dynamic view of depolarization and repolarization of the atria and ventricles [GGS17].

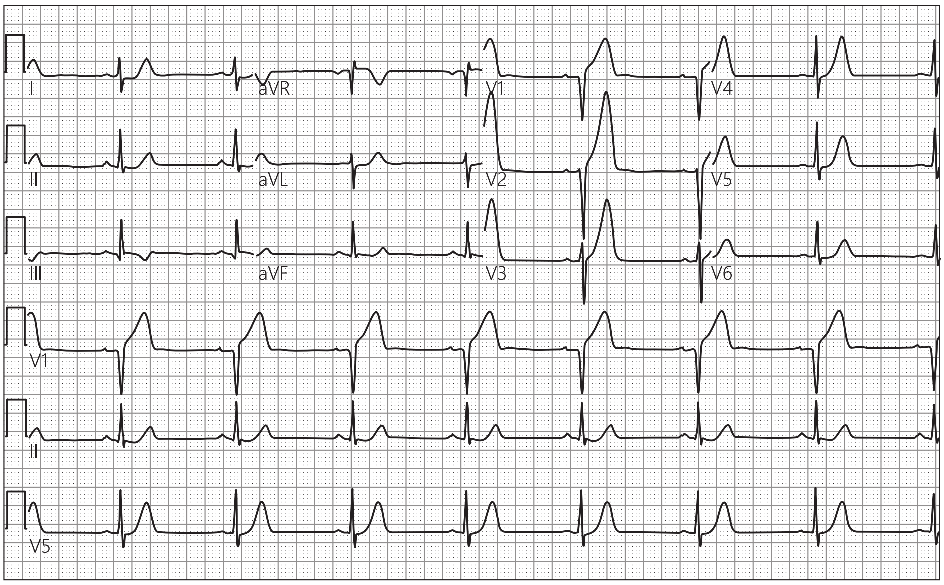

Figure 2: 12 Lead ECG Signal

2.2 ECG Signal Database

Physionet is a repository of biological signals created by the collective effort of researchers from different American universities with the support of the National Institutes of Health (NIH). Built with the goal of disseminating and cultivating research in the area of biomedical signals, Physionet contains signals from different types of tests such as electrocardiograms, electroencephalograms, and CT scans; from healthy patients and patients with different kinds of disorders such as arrhythmias, neurological disorders, sleep apnea, and aging [GAG$^+$00], all publicly available for use by the academic community.

The ECG heartbeat signals used in this work were obtained from the open-source St Petersburg INCART 12-lead Arrhythmia Database [GAG$^+$00] (INCART) on Physionet. The database consists of 75 ECG recordings extracted from 32 Holter records. In this database, each recording is 30 minutes long, containing the 12 standard leads, each sampled at 257 Hz, with over 175,000 annotated heartbeats. The beat annotations were produced automatically by an algorithm and later manually corrected.

2.3 Multilayer Neural Networks

Artificial neural networks (ANNs) are a class of supervised learning algorithms that have their construction inspired by the functioning of the human brain [Nie15]. Like other supervised learning algorithms, ANNs are used in classification and regression problems in which the input data and their respective outputs are known. The goal of the algorithm is to discover a mapping function f that from an input X leads to an output Y.

On ANNs this mapping function is obtained by combining elementary units called neurons. An artificial neuron (or perceptron) is composed of inputs (x_1, x_2, $\ldots$, x_n), weights (w_1, w_2, $\ldots$, w_n), sum function and an activation function (Fig. 3). Any input received by a perceptron is subjected to multiplication of its values by their respective weights, and the results are then summed.

To determine the output of a neuron, an activation function is applied with the goal of mapping the result of the sum of its weights within coherent bounds for its application [GBC16]. The Sigmoid function (Fig. 4a), for example, is classically used as a likelihood function for binary classification. Other common types of activation functions are the hyper-

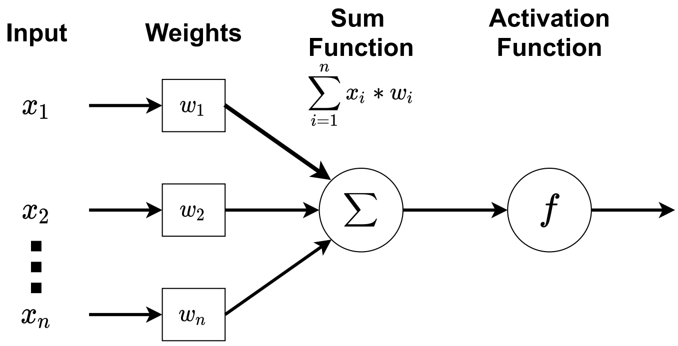

Figure 3: Exemple of a neuron with n inputs.

bolic tangent (Fig. 4b), rectified linear unit (ReLU) (Fig. 4c) and linear function (Fig. 4d). In most applications, the ReLU activation function is used because it contains a non-linear characteristic, facilitating generalization and adaptation to the data, and at the same time is computationally simple compared to the others, allowing a faster training process.

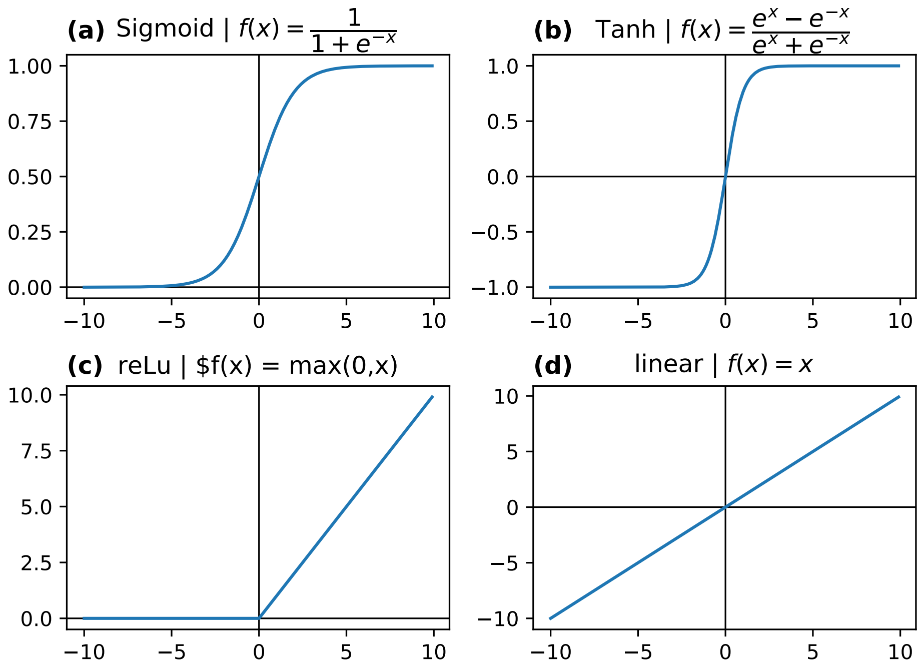

Figure 4: Activation Functions: (a) Sigmoid; (b) Hyperbolic Tangent (Tanh); (c) Rectified Linear (ReLU - Rectified Linear Unit) and; (d) Linear.

Despite being the most popular activation function, the ReLU may suffer from a problem called the dying ReLU problem. When using the ReLU, neurons can, under certain conditions, enter in a state of perpetual inactivation where it gives no output for any input and produces no gradient, making it essentially "dead", as it has no contribution to the neural network anymore. To mitigate this, a variation of the ReLU can be used, known as Leaky ReLU (eq 1). There is a slight positive slope in this activation function when the neuron is inactive, making possible the recovery from a dying state.

$$\text{LeakyReLU} = \begin{cases} X, & \text{for } X \geq 0 \\ 0.01 * X, & \text{for } X < 0 \end{cases} \tag{1}$$

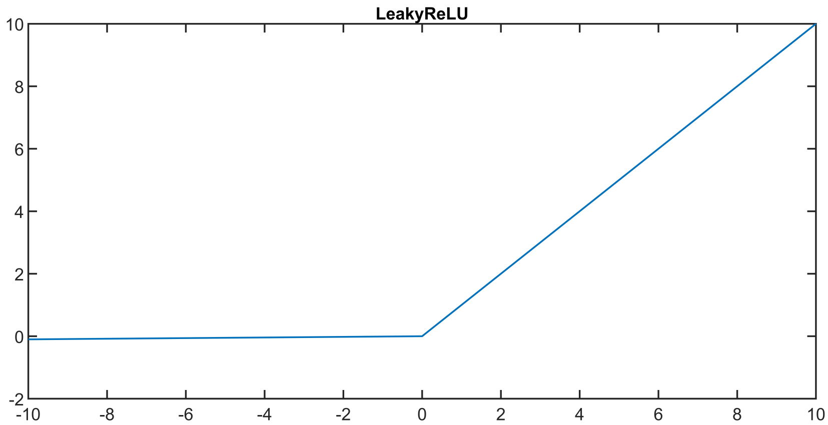

Figure 5: Leaky ReLU activation function

The multilayer perceptron artificial neural networks, also known by the term multilayer perceptron, or fully-connected, present themselves with the combination of several perceptrons organized into layers, which can be divided into input layer (represented further to the left), where data is fed into the network; output layer (further to the right), where the responses coming from the network are obtained and; hidden layers, located between the input and output layers. As mentioned earlier, each neuron has an associated weight. Thus a neural net intended for parametric estimation processes is considered trained and validated when the combination of weights of all neurons is such that the error between the estimated parameters and the actual parameters is minimal.

During the training phase, the neurons' weight values are constantly updated according to a measured network error in a process known as backpropagation. This network error is achieved by using a Loss Function that measures how good the network output is when compared with an expected output.

2.4 Loss Functions

A loss function (or cost function) is a function that estimates the cost of taking a decision or action by mapping its inputs to a real number [Bis06]. In machine learning, loss functions can be used in classification problems to estimate the quality of a prediction by adopting a high value for incorrect predictions, making the goal of the algorithms to minimize loss [Bis06]. In this works, two loss functions will be discussed: Binary Cross Entropy and Contrastive Loss [HCL06].

Binary cross entropy loss (or logarithmic loss) is a loss function typically used in binary classification. Given a machine learning model output class and an expected class, a value is calculated based on how distant they are from each other following Eq 2; where L is the loss function, y is the expected class, and d is the output of the model. Despite not being designed for metric learning problems, this loss function can still be used as a metric learning problem can easily be transformed into a classification problem by adopting two classes: one when the outputs are the same and one when the outputs are different.

$$L = -(ylog(d) + (1-y)log(1-d)) \tag{2}$$

Despite Binary cross entropy being usable in metric learning problems, a new loss function designed for this type of problem was desired. In 2006, The Contrastive Loss was proposed by Hadsell *et al.* [HCL06] as part of a method of dimensionality reduction, a concept similar to the core idea of embedded learning. This loss function aims to keep samples that are close in the source domain together and samples that are distant apart [HCL06]. Given a pair of samples, the Contrastive Loss is defined by:

$$L = yd^2 + (1-y)max(margin - d, 0)^2 \tag{3}$$

where L is the loss function, y signals if the samples are close or not in the source domain, d is the distance between the two samples in the target domain and $margin$ is a value ta limits the contribution of distant pairs to the loss function.

2.5 Convolutional Neural Networks

Convolutional neural networks (CNNs) were first used in computer vision to highlight more relevant regions in data by applying filters to generate transformed data, which can bring more relevance to a particular feature of the original data. Thus, the difference between a convolutional network to a conventional neural network is that it replaces matrix multiplication with the convolution operation [GBC16].

$$s(t) = \int x(a)w(t-a)dA \tag{4}$$

In a convolutional layer the convolution operation (Eq. 4) is applied on the source data x using a set of weights w known as kernels. The kernel is moved within the source data to a specific offset (stride), where the convolution operation is applied, obtaining a single value for each position. In the figure 6 the convolution operation is demonstrated using a 3x3 kernel and a displacement of 1, generating as a final product a new matrix with dimension 4x4 that seeks to highlight the characteristics of the original data. This type of operation works similarly on 1-dimensional signals. In this case the operation becomes similar to filtering a signal by a sliding window.

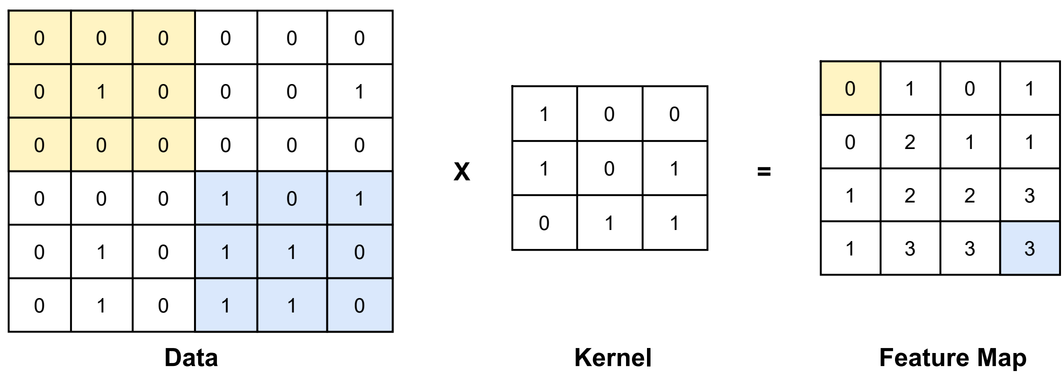

Figure 6: Simplified Example of 2D Convolution

The convolutional operation is usually followed by a clustering layer, also known pooling, which aims to simplify the feature maps generated by the convolutional operation. Among the clustering techniques, we can mention the following: Max Pooling, which returns the maximum value of each region; and Average Pooling, which produces the average value of each region. An example can be seen in Figure 7, where the two mentioned techniques are applied to the feature map generated by the convolutional operation illustrated in Figure 6 with stride 2 (two).

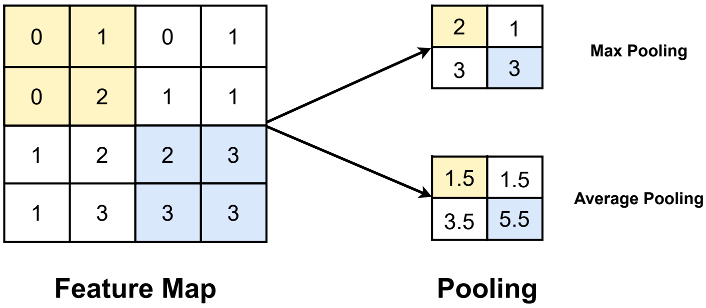

Figure 7: Example of pooling with stride 2

2.6 Few Shot Learning

Few-Shot Learning is a machine learning paradigm that aims to allow supervised learning algorithms to learn from a limited number of examples. Among its main uses, this paradigm is used when [WYKN20]:

1. The model needs to learn rare cases;
2. The cost of collecting and annotating a robust database becomes too high;
3. You need to make the machine learn like a human being.

Few Shot Learning algorithms can be divided into three categories according to the context in which prior knowledge of the problem is applied: in data, model, and algorithm. Using prior data knowledge seeks to improve the database of a model to achieve a satisfactory generalization function. To do this, one may have to convert an existing dataset into a new type of information that can facilitate the training of another model [SKS+18] [KHN16], classify unlabeled or weakly labeled samples to increase the amount of data for training [DSHJ18] [WLD+18] or generate data similar to the original database artificially [TS17] [GSZ+18].

In the model context, Few Shot Learning algorithms seek to limit the solution search space, as this facilitates convergence to a satisfiable function. Models that solve specific parts of a problem can be combined with parameter sharing to solve a more generic problem (Multitask Learning) [LZHFF17] [BW18]. The search space can also be simplified by looking for a function capable of mapping the samples to a feature space in which it is easy to differentiate the database classes using a similarity function (Embedded Learning or Metric Learning) [BHV+16] [VBL+16]. Other techniques make use of generative models and likelihood functions (Generative Modeling) [STT11].

Few Shot Learning methods are also used to guide parameter development within models. Some approaches include: adapting a series of parameters θ_0 from a model performing one type of task to parameters θ from another similar task [YGY+18][CMPT+17]; refining training parameters according to their performance [RRS+18] [FAL17]; as well as learning an optimization function to adjust model parameters during training [RL16] [ADG+16].

2.6.1 Siamese Neural Networks

Originally developed to verify handwritten signatures in images [BGL+93], a Siamese neural network (SNN) is composed of twin networks that share the same weights and architecture. Each of these twin networks accepts a different set of inputs, with the intent of producing an embedding function that maps those inputs into a d-dimensional space where the value of a similarity function f is low for inputs of the same class and high for inputs of different classes. [WYKN20]

Traditionally, neural networks are trained in a fixed number of classes, and the addition or removal of these classes is seen as a problem. In that case, the neural network must be retrained to accommodate those changes. In a SNN, this is bypassed since it learns to compare the two inputs and check whether they are similar or not. So, adding a class becomes as simple as adding another scenario to compare with the samples [KZS15]

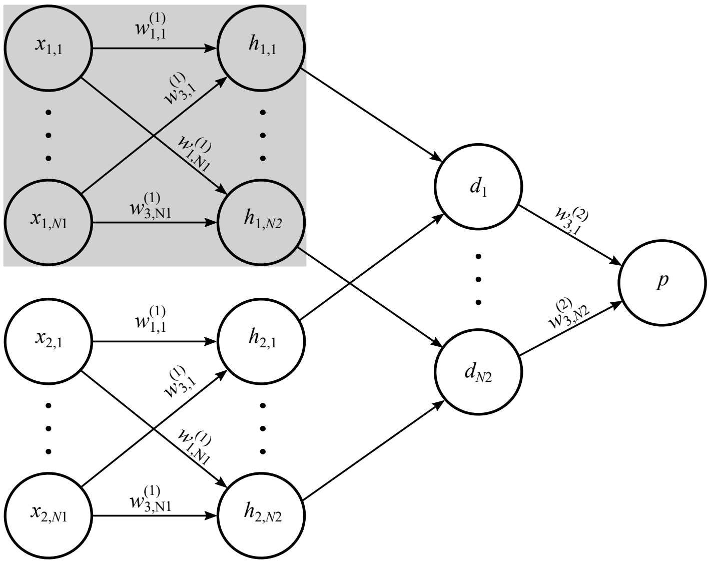

Figure 8: Example of a SNN with two twin networks, with network 1 highlighted in the figure with a gray shaded region. The features extracted by the twin networks in the hidden layers ($h_{i,j}$) have their similarity calculated in a common layer (d_j). Finally, the node p represents a logistic likelihood function. Adapted from [KZS15]

As an Embedded Learning algorithm, the network maps inputs to a feature space where it is easier to discriminate different classes. Because it is composed of a set of networks with the same parameters, it is unlikely that similar data will be mapped to very different locations in the feature space (Fig 9). With this, for a coherent mapping function, the similarity function should have low values for samples of the same class and high values for samples of a different class.

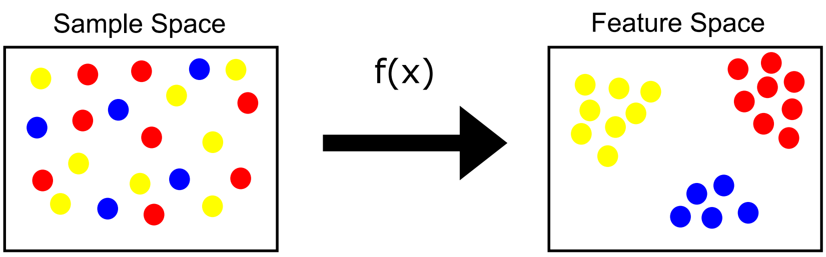

Figure 9: Mapping from sample space to feature space.

The use of convolutional layers in this type of network is particularly advantageous, as the convolution operation has filtering characteristics and can be used to enhance patterns in data segments. This way, the output of a trained convolutional layer can be used to represent important characteristics in its input and provide certain robustness to noise [KZS15].

2.7 Chapter Discussion

This chapter is a theoretical framework for understanding the rest of the text. An explanation about selected topics of interest was presented to enlighten the reader about the network architecture and electrocardiogram signals. The proposed methodology to classify ECG signals will be detailed in the next chapter.

3 Methodology

The methodology of this work will be addressed in this section, as well as the steps in the pre-processing phase, the description of siamese convolutional neural network architecture, and the experimental setup utilized.

3.1 Pre-processing

To decrease the influence of noise in the models' performance, each ECG signal on the dataset has gone through a filtering step to remove noise caused by sources such as line power, muscle movement, or poorly attached electrodes. The filtering method used aims to remove unwanted frequencies that are outside the spectrum of ECG signals frequencies that are between 0.5 and 40 Hz [BGM12] [ZAAB12]. A Discrete Wavelet Transform (DWT) approach with Daubechies 4 as the mother wavelet was used. This approach works by applying a DWT to the signal and then discarding the resulting wavelet components that represent low and high-frequency noise. As an additional step, a second-order Butterworth bandstop filter with a 50 Hz cutoff frequency was employed to reduce powerline noise.

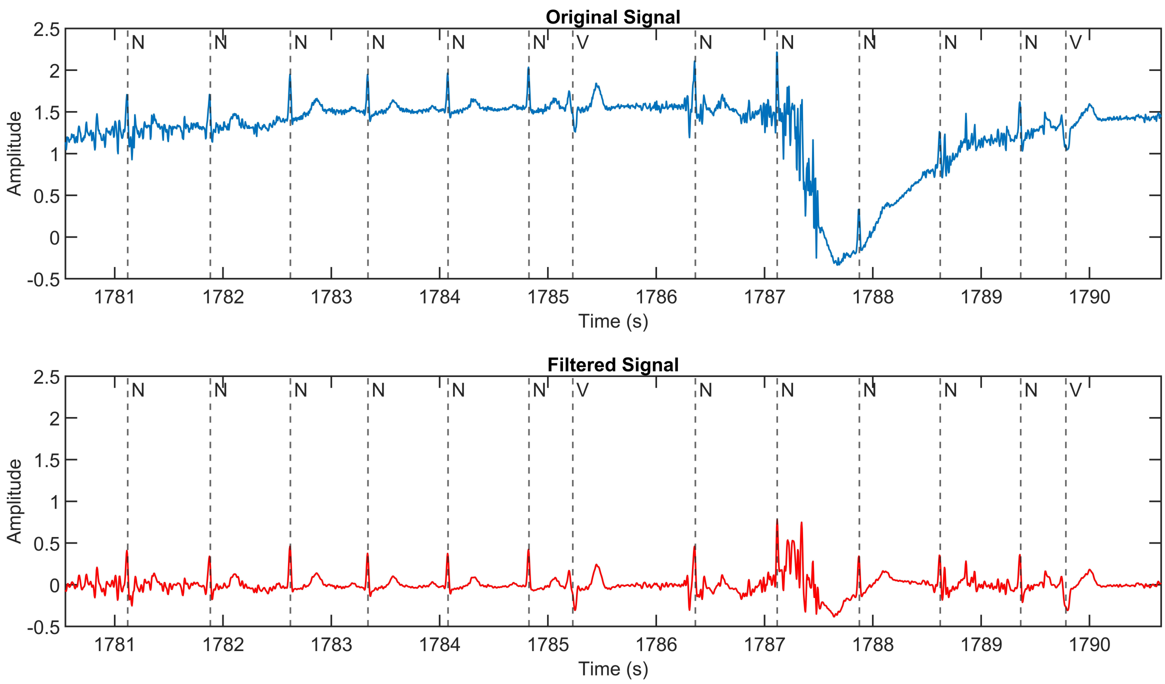

Figure 10: Original and Filtered ECG Signals

After the filtering process, each heartbeat was extracted from the signals by adapting the methodology presented in Baloglu *et al.* [BTY$^+$19] to use a lower sampling rate, i.e., the INCART base rate itself. Heartbeats samples were then collected from the filtered signals by extracting segments located around the pre-annotated R waves. The extracted segments contain from 65 samples before the R wave to 103 samples after it, totaling 169 points per heartbeat. This sample range can be recalculated for signals of other databases by adjusting its values in accordance with the database's frequency using a simple rule of three. This procedure is then done to every lead of the 12 standard leads, with the collected heartbeats from all 12 leads of each R wave annotation being concatenated into a single signal with 2028 samples of length. An example can be seen in figure 11.

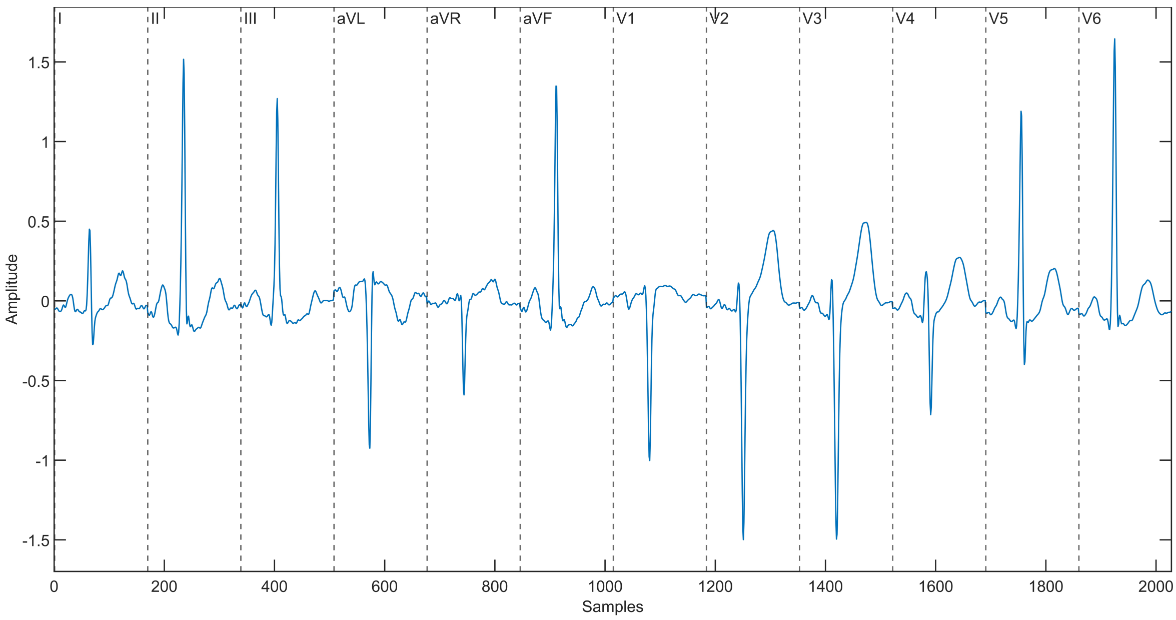

Figure 11: ECG signal with the 12 leads concatenated

In this work, heartbeats were select from 7 classes contained in the INCART database [GAG$^+$00] and split according to table 1

Heartbeat Class	Number of Heartbeats
Atrial Premature	1,943
Fusion of Ventricular and Normal	219
Nodal (Junctional) Escape	92
Normal	150,393
Premature Ventricular Contraction	20,008
Right Bundle Branch Block	3,173
Supraventricular Premature	16
Total	175,844

Table 1: Number of heartbeats for each class

3.2 Siamese Convolutional Neural Network

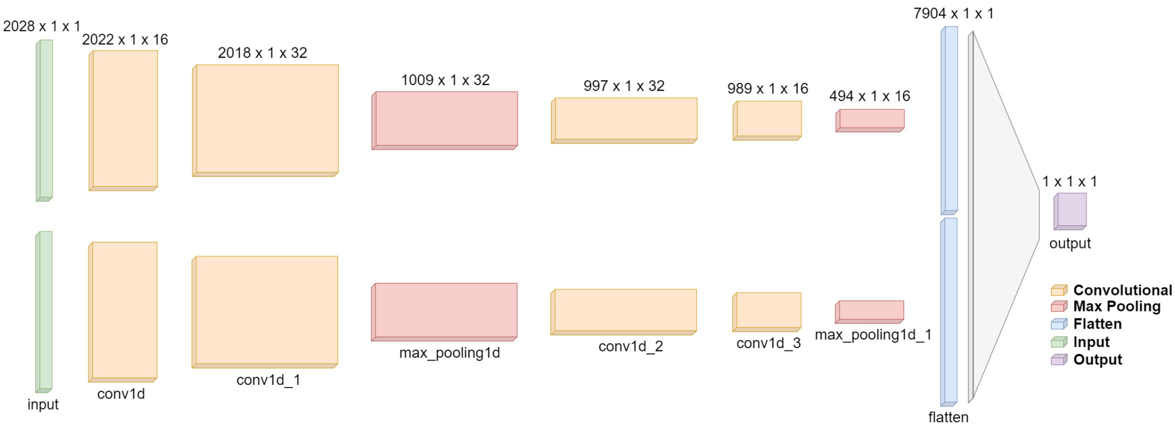

Figure 12: Proposed model architecture

The proposed SCNN is made up of 8 layers, organized as seen in figure 12. The number of layers and their disposition were obtained through empiric experimentation. LeakyReLU activation functions were employed in all convolutional layers to reduce the risk of neurons "dying" at the cost of a higher computational cost. The output layer uses a sigmoid activation function to output values between 0 and 1. The detailed parameters of each layer can be seen in table 2. To investigate its applicability in ECG signals, a combination of four similarity functions and two loss functions resulted in eight different models. With respect to the similarity functions, the L1 distance (Eq 5), L2 distance (Eq 6), Mean Squared Error (MSE) (Eq 7) and Root Mean Squared Error functions (RMSE) (Eq 8) were used.

$$L1 = \sum_{i=1}^{N} |A_i - B_i| \tag{5}$$

$$L2 = \sqrt{\sum_{i=1}^{N} (A_i - B_i)^2} \tag{6}$$

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (A_i - B_i)^2 \tag{7}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (A_i - B_i)^2} \tag{8}$$

Regarding the loss functions, two were tested: Binary Cross Entropy and Contrastive Loss [HCL06]. Those two loss functions were designed with different objectives in mind: Binary Cross Entropy was developed for classification problems and Contrastive Loss for metric-based problems. As the similarity problem can be reduced to a binary classification problem with the classes being "Same" or "Different", binary cross entropy is a commonly used loss function in this type of model, even with the existence of specialized loss functions.

Nº	Layer	Parameters	Output
1	1D Convolution	16×7, Stride=1, Input = (2028, 1), Activation = LeakyReLU	2022×16
2	1D Convolution	32×5, Stride=1, Activation = LeakyReLU	2018×32
3	MaxPooling 1D	Pool size=2, Stride=2	1009×32
4	1D Convolution	32×13, Stride=1, Activation = LeakyReLU	997×32
5	1D Convolution	16×9, Stride=1, Activation = LeakyReLU	989×16
6	MaxPooling 1D	Pool size=2, Stride=2	496×16
7	Flatten	-	7904

Table 2: Network structure and layer parameters.

In order to assign a target sample to a class, a simple decision process was employed. A sample of each class was manually selected to form a reference set. Signals were selected based on the visual format of their waves when compared with signals of the same class found in other sources. The selection of this reference sample is essential to the quality of the predictions, as it has to contain the most significant characteristics of its class. Pairs were formed by the target sample and each reference sample from the reference set and associated with its class. They were then fed into the model, with the resulting class being assigned to pair with the highest output similarity.

3.3 Experimental Setup

The dataset was split into a 75-15-10 ratio in a stratified form, as in each split has near the same proportion of samples of each class. During training, each sample in the split would produce two pairs of signals, one formed by the sample and another randomly selected sample of the same class and another created by the sample and a randomly selected sample from a different class. This way, the input to the model is equally distributed between positive (same class) and negative (different class) pairs.

The models were implemented with the Python programming language and Keras framework, running on an Nvidia RTX 2060 GPU, an Intel(R) Core(TM) i7-10875H CPU, and 32GB of ram. In the training stage, the ADAM optimizer was used with a learning rate of 0.001 and batch size of 128 samples, running for 50 epochs. Those values were also obtained after empirical experimentation. As described in the previous section, binary cross entropy and contrastive loss were used as loss functions.

3.4 Chapter Discussion

In chapter 3, a discussion about the preprocessing steps, data gathering, model architecture, and decision process was presented. The experimental setup used to achieve the results shown in the next chapter was also displayed.

4 Results

To minimize the effect of random outcomes on the results of the experiments, ten models of each combination of the loss function and similarity functions were trained. Error plots were then employed to show the accuracy and loss of the models on the validation dataset. On those plots, the lines on the graph are the average accuracy of the ten generated models, while the error bar is its standard deviation.

As seen in figure 13, the loss value of the models with Contrastive loss tends to fluctuate less than the ones with Binary Cross Entropy. All of the models' losses stagnated after close to 45 epochs, denoting that improvement from adding more training epochs could happen but is unlikely. However, fine-tuning the training parameters can still be done to reach better solutions.

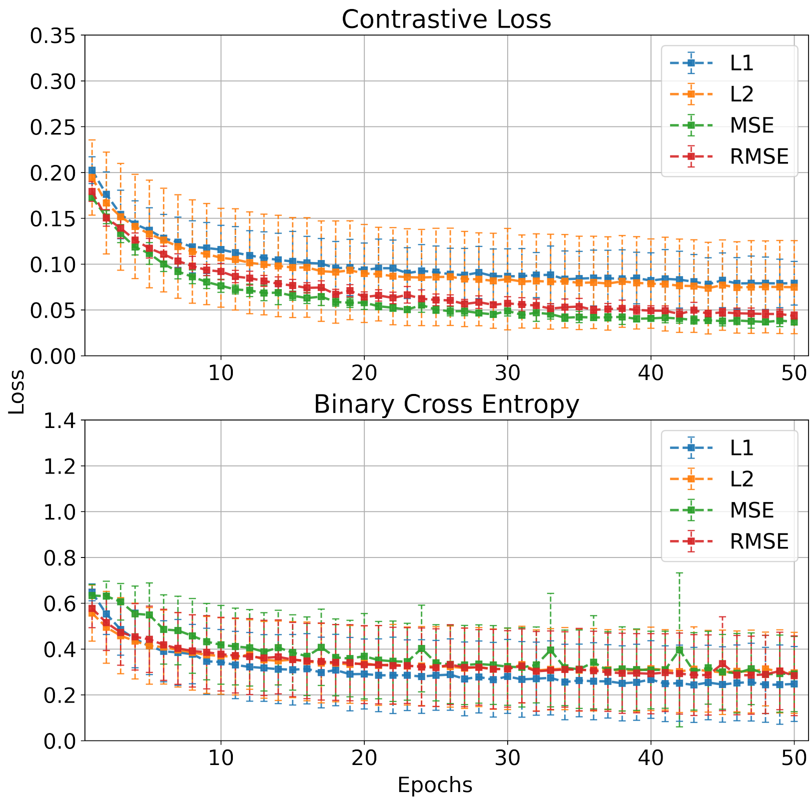

Figure 13: Error plot of the models loss after 10 executions

In figure 14, it's noticeable that the models that used Contrastive Loss as their loss function have a lower standard deviation value during the training process. In particular, this value is the lowest when the Contrastive Loss function is paired with the MSE or RMSE similarity functions. This can mean that those models achieved convergence to a similar accuracy value with a relatively high frequency. However, the models that used Binary Cross-Entropy showed a high accuracy variation. This variation was exceptionally high when paired with MSE or RMSE similarity functions, contrasting with what happened when combined with Contrastive Loss.

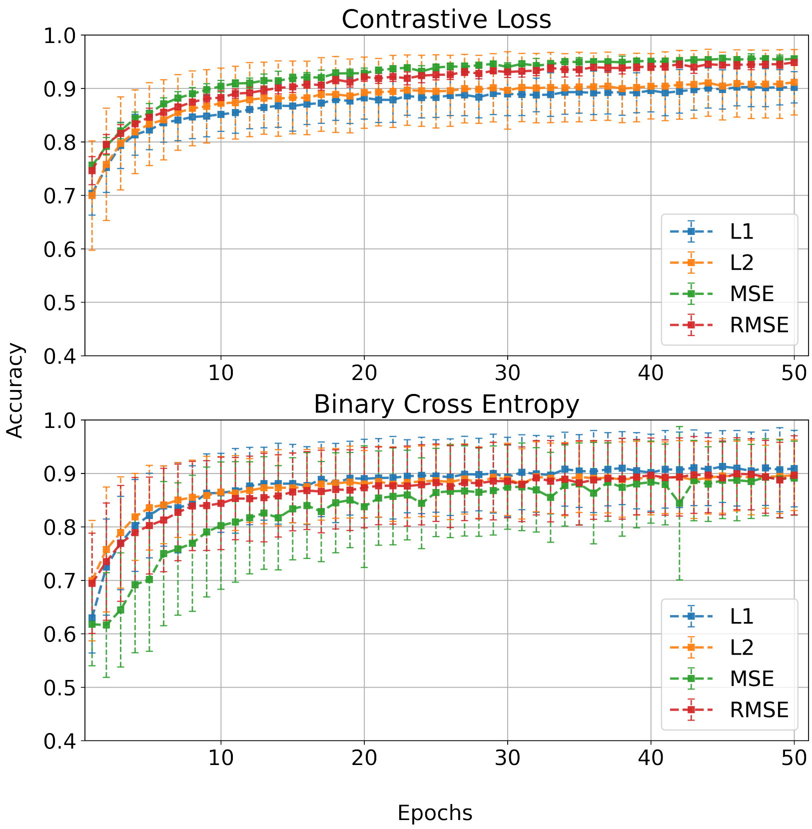

Figure 14: Error plot of each model accuracy after 10 executions

On tables 3 and 4, the average metrics after ten executions for the models with Binary Cross Entropy and Contrastive Loss can be seen. In this present scenario, the MSE and RMSE models coupled with Contrastive Loss achieved an overall better quality metrics when compared with the other researched combinations, with metrics such as 95.6% and 95.9% of accuracy and 96.1% and 94.9% of precision. The results obtained from the models that used the Binary Cross Entropy were very close to each other, with the model using the L1 distance as a similarity function obtaining slightly better results.

Table 3: Average Metrics for Models using Binary Cross Entropy

FUNC	ACC	PREC	RECALL	SPECI
L1	0.909	0.905	0.914	0.902
L2	0.896	0.889	0.906	0.885
MSE	0.890	0.878	0.905	0.874
RMSE	0.896	0.881	0.920	0.873

Table 4: Average Metrics for Models using Contrastive Loss

FUNC	ACC	PREC	RECALL	SPECI
L1	0.901	0.910	0.893	0.909
L2	0.910	0.912	0.909	0.911
MSE	0.956	0.961	0.950	0.962
RMSE	0.949	0.949	0.948	0.949

A per class analysis was also produced for the most accurate model of each loss function: a model using Binary Cross Entropy combined with the L1 similarity function; and a model using Contrastive Loss with the MSE similarity function. For the remainder of this section, those models will be referred to as the "Binary Cross Entropy Model" and the "Contrastive Loss Model".

Figure 15 shows a heatmap of the results obtained for each class using the Binary Cross Entropy model. The value of each cell represents the proportion of the predicted class in relation to the number of elements in the true class, making each line sum up to 1. According to the figure, this model achieved great results on the classes with a large number of samples (N, R, and V) and surprisingly with the low sampled j class. However, in classes with a small sample count, it was sub-par. The "F" class, for example, was often mislabeled with either the "V" or N label, and the "S" class mainly was recognized as a normal heartbeat.

The precision of the classification of the "S" class is particularly intriguing, as most of its classifications were false positives far exceeding the number of samples of that class, making its precision value plummet, as seen in table 5. The "A" class classification, on the other hand, while having a not so high recall value, achieved a low number of false positives.

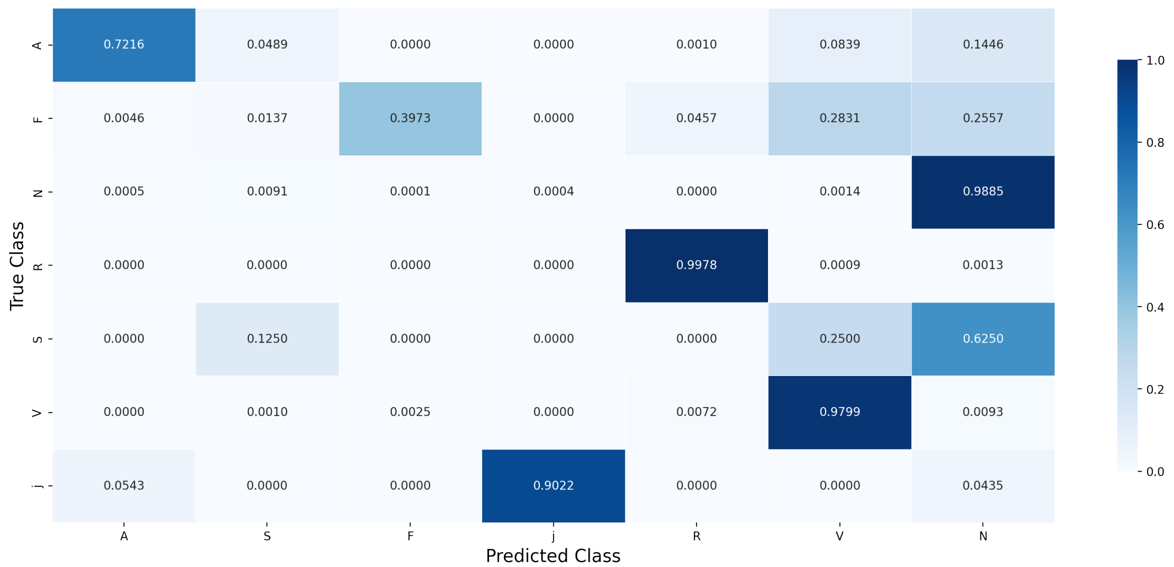

Figure 15: Heatmap of the results of the Binary Cross Entropy Model.

For the Constrastive Loss model, the results are slightly worse in comparison to the previous model when looking at normal heartbeats (N) classification, but is better everywhere else (fig 16). Gains from using Contrastive Loss as the loss function can be inferred to be smaller in classes with large sample counts, like the "V", "R", and "N" classes. However, the classification of those classes became more consistent with the reduction of the number of false positives.

In general, the classification of classes with fewer samples is considerably better. Classification of the "J" class achieves a recall value of 100%, but has a lot more false positives, especially with the misclassification of the "F" class. Still, the "F" class classification recall rose sharply compared to what was achieved using the Binary Cross Entropy Model, with a value of 68.04% versus the 39.73% shown previously. Similarly, metrics for the "S" class classification are better, with a much higher recall and precision due to a reduced number of false positives and an increased number of true positives

Figure 16: Confusion Matrix from the results of the Contrastive Loss Model

Table 5 makes a comparison of this work with some of those that can be found in the literature. With respect to the classification of the high sampled classes ("N", "R", and "V"), the proposed models achieved values that are comparable to those of other authors. A pleasant surprise was the classification of heartbeats of the "F" class, with the Contrastive Loss model achieving better results than the best model listed listed with 64.35% precision and 68.04% recall in comparison with 23.58% precision and 11.07% recall.. While not good, the classification of the "S" class heartbeats using the Contrastive Loss model is on par with the values found in other works. There were no papers found during this research that classified the heartbeats of the "j" class.

Model	A		F		N		R		S		V		j	
	Pre	Rec	*Pre*	Rec	*Pre*	Rec	*Pre*	Rec	*Pre*	Rec	*Pre*	Rec	*Pre*	Rec
Llamedo [LM12]	-	-	-	-	99%	92%	-	-	11%	89%	88%	82%	-	-
Llamedo *et al.* [LM10] Imbalanced	-	-	-	-	99%	92%	-	-	11%	85%	88%	82%	-	-
Llamedo *et al.* [LM10] Balanced	-	-	-	-	92%	92%	-	-	80%	85%	87%	82%	-	-
Llamedo *et al.* [LM10] By Recording	-	-	-	-	90%	93%	-	-	66%	64%	86%	71%	-	-
Li *et al.* [LLW+14]	-	-	-	-	-	-	-	-	-	-	66.5%	93.4%	-	-
Romdhane *et al.* [RP20] W/O Focal Loss	-	-	21.82%	9.64%	97.96%	98.59%	-	-	59.34%	53.47%	91.96%	88.78%	-	-
Romdhane *et al.* [RP20] W/ Focal Loss	-	-	23.58%	11.07%	97.98%	98.78%	-	-	64.32%	51.41%	92.71%	89.16%	-	-
Rajesh *et al.* [RD17] Linear	88.71%	86.90%	-	-	98.79%	98.60%	90.19%	96.60%	-	-	95.67%	90.70%	-	-
Rajesh *et al.* [RD17] RBF	92.67%	91.0%	-	-	99.79%	97.90%	94.55%	97.10%	-	-	92.77%	93.70%	-	-
Rajesh *et al.* [RD17] Cubic	91.78%	91.60%	-	-	99.0%	99.20%	94.64%	97.20%	-	-	95.16%	92.60%	-	-
Aziz *et al.* [AAA21]	-	-	-	-	100%	99.60%	100%	100%	-	-	99.5%	100%	-	-
Das *et al.* [DA14]	-	-	15.3%	51.8%	99.7%	93.3%	-	-	19.3%	87.0%	89.1%	94.3%	-	-
Proposed Binary Cross Entropy	94.66%	72.16%	55.41%	39.73%	99.63%	98.85%	95.16%	99.78%	0.13%	12.50%	97.81%	97.99%	60.58%	90.22%
Proposed Contrastive Loss	95.96%	77.06%	65.35%	68.04%	99.73%	99.62%	99.21%	99.62%	25.0%	62.50%	96.81%	98.81%	49.72%	100%

Table 5: Comparison Table between different heartbeat classification methods

One issue identified with the use of SCNNs combined with the proposed decision process is that its results are highly sensitive to the quality of the reference set. A reference set composed of miss-labeled, highly noisy, or ill-conditioned signals has a very negative impact on the quality metrics of the proposed models when trained in noisy datasets, as similarities can be found between the noisy reference and noisy target sample.

4.1 Chapter Discussion

This chapter shows the results obtained by employing the proposed methodology in ECG heartbeat classification. The obtained metrics were discussed, comparing the different combinations presented and results from other works found in the literature.

5 Conclusion

In this work, a Siamese Convolutional Network Model for heartbeat classification from ECG signal tracings was shown. This type of neural network learns to embed samples in a feature space by virtue of a similarity function instead of classifying; this way, it has better capabilities of handling the existence of unknown classes and classes with low sample numbers when compared to traditional neural networks.

Eight models of Siamese Neural Networks were tested. The models were built with the same layer configuration but with different loss and similarity functions. The models that used Contrastive Loss as the loss function achieved overall better results than those using Binary Cross Entropy. As a specialized loss function, the use of Contrastive Loss seemed to improve the classification results of classes containing a small number of samples like the "F", "j", and "S" classes, going from 39.73%, 90.22% and 12.50% recall to 68.04%, 100% and 62.50% respectively .

Compared with similar literature models, this work presented great results, especially when classifying heartbeats of the "F" class. This classification achieved values 65.35% precision and 68.04% recall that far exceeds values found in other works. The classification of the classes with a high number of samples was in line with what was found in other works, with precision and recall values well above the 95% mark. The classification of the "A" class, while worse than what was achieved with other methods, was still solid with 95.96% precision and 77.06%.

5.1 Future Works

Further investigation on the use of this network architecture for ECG signal classification is encouraged, as it achieved this result with a relatively simple architecture. A denser network architecture or the use of well known signal processing models combined with a more thorough tuning of its hyperparameters may improve the results significantly. A more robust preprocessing step and an automated reference signal selection could also be employed to reduce the influence of noisy signals in the network results.

Changes in the input format could also be investigated. Using a vertical stack of the 12 ECG leads instead of a horizontal concatenation would allow for the use of 2D convolutions, rendering possible interactions between the ECG leads during the convolution process that are not possible otherwise. A more complex decision process could also be employed by combining the output of a trained SCNN with other machine learning algorithms. Finally, as this type of neural network only learns the embedding, this training process can be easily used to obtain a feature extractor module that can be used in other types of neural networks, making it readily reusable.

References

[AAA21] Saira Aziz, Sajid Ahmed, and Mohamed-Slim Alouini. Ecg-based machine-learning algorithms for heartbeat classification. *Scientific reports*, 11(1):1–14, 2021.

[AAM20] 2020 physician specialty data report, 2020.

[ADG+16] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *arXiv preprint arXiv:1606.04474*, 2016.

[AFL+17] U Rajendra Acharya, Hamido Fujita, Oh Shu Lih, Muhammad Adam, Jen Hong Tan, and Chua Kuang Chua. Automated detection of coronary artery disease using different durations of ecg segments with convolutional neural network. *Knowledge-Based Systems*, 132:62–71, 2017.

[BGL+93] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6:737–744, 1993.

[BGM12] S Banerjee, R Gupta, and M Mitra. Delineation of ecg characteristic features using multiresolution wavelet analysis method. *Measurement*, 45(3):474–487, 2012.

[BHV+16] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip HS Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. *arXiv preprint arXiv:1606.05233*, 2016.

[Bis06] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.

[BTY+19] Ulas Baran Baloglu, Muhammed Talo, Ozal Yildirim, Ru San Tan, and U Rajendra Acharya. Classification of myocardial infarction with multi-lead ecg signals and deep cnn. *Pattern Recognition Letters*, 122:23–30, 2019.

[BW18] Sagie Benaim and Lior Wolf. One-shot unsupervised cross domain translation. *arXiv preprint arXiv:1806.06029*, 2018.

[CMPT+17] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

[DA14] Manab K Das and Samit Ari. Patient-specific ecg beat classification technique. *Healthcare technology letters*, 1(3):98–103, 2014.

[DSHJ18] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. Low-shot learning with large-scale diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3349–3358, 2018.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[GAG+00] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[GGS17] Ary L Goldberger, Zachary D Goldberger, and Alexei Shvilkin. *Clinical electrocardiography: a simplified approach e-book*. Elsevier Health Sciences, 2017.

[GSZ+18] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. *arXiv preprint arXiv:1810.11730*, 2018.

[HCL06] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[HZS+20] Shenda Hong, Yuxi Zhou, Junyuan Shang, Cao Xiao, and Jimeng Sun. Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review. *Computers in Biology and Medicine*, page 103801, 2020.

[IPM+96] Nikhil Iyengar, CK Peng, Raymond Morin, Ary L Goldberger, and Lewis A Lipsitz. Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 271(4):R1078–R1084, 1996.

[JD18] Linpeng Jin and Jun Dong. Normal versus abnormal ecg classification by the aid of deep learning. *Artificial Intelligence–Emerging Trends and Applications. InTech Open*, pages 295–315, 2018.

[KHN16] Roland Kwitt, Sebastian Hegenbart, and Marc Niethammer. One-shot learning of scene locations via feature trajectory transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 78–86, 2016.

[KZS15] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[LLW+14] Peng Li, Chengyu Liu, Xinpei Wang, Dingchang Zheng, Yuanyang Li, and Changchun Liu. A low-complexity data-adaptive approach for premature ventricular contraction recognition. *Signal, Image and Video Processing*, 8(1):111–120, 2014.

[LM10] Mariano Llamedo and Juan Pablo Martínez. Heartbeat classification using feature selection driven by database generalization criteria. *IEEE Transactions on Biomedical Engineering*, 58(3):616–625, 2010.

[LM12] Mariano Llamedo and Juan Pablo Martínez. An automatic patient-adapted ecg heartbeat classifier allowing expert assistance. *IEEE Transactions on Biomedical Engineering*, 59(8):2312–2320, 2012.

[LWL21] Zongjin Li, Huan Wang, and Xinwen Liu. A one-dimensional siamese few-shot learning approach for ecg classification under limited data. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021.

[LYFW21] Tianyu Liu, Yukang Yang, Wenhui Fan, and Cheng Wu. Few-shot learning for cardiac arrhythmia detection based on electrocardiogram data from wearable devices. *Digital Signal Processing*, 116:103094, 2021.

[LZHFF17] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li Fei-Fei. Label efficient learning of transferable representations across domains and tasks. *arXiv preprint arXiv:1712.00123*, 2017.

[MM01] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[Nie15] Michael A Nielsen. *Neural networks and deep learning*, volume 2018. Determination press San Francisco, CA, USA:, 2015.

[Org20] World Health Organization. The top 10 causes of death, 2020.

[RD17] Kandala NVPS Rajesh and Ravindra Dhuli. Classification of ecg heartbeats using nonlinear decomposition methods and support vector machine. *Computers in biology and medicine*, 87:271–284, 2017.

[RL16] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[RP20] Taissir Fekih Romdhane and Mohamed Atri Pr. Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss. *Computers in Biology and Medicine*, 123:103866, 2020.

[RRP+20] Antônio H. Ribeiro, Manoel Horta Ribeiro, Gabriela M. M. Paixão, Derick M. Oliveira, Paulo R. Gomes, Jéssica A. Canazart, Milton P. S. Ferreira, Carl R. Andersson, Peter W. Macfarlane, Wagner Meira Jr., Thomas B. Schön, and Antonio Luiz P. Ribeiro. Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nature Communications*, 11(1):1760, 2020.

[RRS+18] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[RSS18] Antonio Franco Ravioli, Patrícia Coelho De Soárez, and Mário César Scheffer. Modalidades de gestão de serviços no sistema único de saúde: revisão narrativa da produção científica da saúde coletiva no brasil (2005-2016). *Cadernos de Saúde Pública*, 34:e00114217, 2018.

[SKS+18] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *arXiv preprint arXiv:1806.04734*, 2018.

[STT11] Ruslan Salakhutdinov, Joshua B Tenenbaum, and Antonio Torralba. Learning to learn with compound hd models. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2061–2069, 2011.

[TS17] Yao-Hung Hubert Tsai and Ruslan Salakhutdinov. Improving one-shot learning through fusing side information. *arXiv preprint arXiv:1710.08347*, 2017.

[VBL+16] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016.

[WLD+18] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[WYKN20] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[YGY+18] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. *arXiv preprint arXiv:1805.07513*, 2018.

[YPTA18] Özal Yıldırım, Paweł Pławiak, Ru-San Tan, and U Rajendra Acharya. Arrhythmia detection using deep convolutional neural network with long duration ecg signals. *Computers in biology and medicine*, 102:411–420, 2018.

[YWLD21] Fan Yang, Guijin Wang, Chuankai Luo, and Zijian Ding. Improving automatic detection of ecg abnormality with less manual annotations using siamese network. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021.

[ZAAB12] Zahia Zidelmal, Ahmed Amirou, Mourad Adnane, and Adel Belouchrani. Qrs detection based on wavelet coefficients. *Computer methods and programs in biomedicine*, 107(3):490–496, 2012.