UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Multi-modal deep learning: Application to integrative modelling of electrocardiography and cardiac imaging

Author:
Nikolaos ATHANASOPOULOS

Supervisor:
Víctor M. CAMPELLO
Karim LEKADIR

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 30, 2023

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Multi-modal deep learning:
Application to integrative modelling of electrocardiography and cardiac imaging

by Nikolaos ATHANASOPOULOS

The field of machine learning has undergone significant advancements and developments in medicine over the past few years. Strong deep learning techniques have been developed for the purpose of processing complicated medical data types as time-series data (such as electrocardiography), genetic data or medical imaging. However, these developments have often taken place in specialised ways, resulting in a lack of deep learning implementations that allow to seamlessly integrate medical data from different types. This diploma thesis investigates the potential of deep learning models to combine electrocardiography (ECG) and magnetic resonance imaging (MRI) data for improved cardiac analysis. The research aims to overcome the costs, complexity and processing time constraints of MRI, by developing a deep learning model capable of predicting cardiac structural dynamics based on ECG signals. The interest of this project relies on the potential of generating from a single ECG signal a pseudo-MRI Image, which corresponds to the same patient, providing significantly more detailed information related to the condition of the heart compared to the ECG alone. In this approach, we will benefit from both the advantages ECG processing has over MRI and the improved understanding of the heart provided by the generated pseudo-MRI. The study utilizes a large collection of ECG and cardiac MRI pairs dataset derived from multiple patients, available in the UK Biobank, in order to build connections between these two modalities. By successfully integrating multimodal data, the proposed model can create opportunities for novel applications in anomaly detection, diagnosis and clinical decision-making. This innovative approach has the potential to transform the field of medical imaging by providing a more affordable, effective and accessible way to analyze cardiac health and disease.

Acknowledgements

In first place, special thanks to my supervisor Victor M. Campello for sharing his knowledge with me and Polyxeni Gontra for the introduction to the project.

I would like to acknowledge Universitat de Barcelona, and more specifically the BCN-AIM Team, for providing a GPU which significantly reduced the computational time and hence we were able to complete the experiments on time.

Last but not least, i would like to express my deepest thanks to my flatmate Nikolaos Andriotis for always listening and advising me.

Contents

Chapter 1

Introduction

Medical imaging plays a crucial role in the diagnosis, treatment, and monitoring of various diseases and conditions. One of the most powerful imaging modalities utilized in the field of medical imaging is Magnetic Resonance Imaging (MRI). An MRI scan is a test that creates clear images of the structures inside the body, providing this way thorough anatomical details. However, MRI processing is often associated with significant challenges, making it an expensive, difficult and time-consuming process.

The fundamental nature of MRI technology itself is one of the main causes of the processing complexity. Strong magnetic fields and radio waves are used in MRI to provide precise images of the inside organs and tissues of the body. The acquisition process requires specialized equipment and skilled personnel to operate the MRI scanner, ensuring patient safety and optimizing image quality. Additionally, MRI datasets usually require various computational resources for storage and processing. In this direction, analyzing MRI data requires a number of difficult tasks. To enhance the quality of captured images, pre-processing methods like noise reduction, artifact removal and image enhancement are frequently required. Algorithms for image reconstruction are also used to convert raw MRI data into comprehensible images. These algorithms make use of complex mathematical models and calculation methods, which increase the computational load and processing time.

Electrocardiography (ECG) offers a considerably more accessible method of monitoring heart activity, in comparison to the difficulties encountered by MRI processing. The heart's electrical signals are measured using an ECG, which can provide important details about the heart's rhythm and operation. ECG recordings are frequently utilized in clinical practice because they are non-invasive, affordable, and easily accessible.

Though, despite the drawbacks we mentioned earlier regarding the processing of MRIs, they proved to be extremely helpful for the analysis of the heart, due to the fact that MRI provide more detailed information compared to ECG. To be more specific, MRI can accurately visualize the size, shape, and position of the heart, as well as even more detailed information regarding the surrounding structures, such as the lungs, blood vessels and adjacent organs, which can be important in evaluating complex cardiac cases. Therefore, while ECG focuses primarily on the electrical activity of the heart, MRI offers a more comprehensive overview of the heart's structure, making MRI a valuable tool for diagnosing and monitoring cardiac conditions.

1.1 Motivation

Machine learning methods have observed an impressive expansion over the past decade, thanks to the advancement of potent deep learning methods [14] that have been utilized across diverse fields, including the medical domain. Deep learning models, in particular, have shown tremendous promise for handling complex medical data types like medical imaging, genetic data, and time-series data like ECG. Nevertheless, because these developments have mainly taken place in specialized settings, there aren't many deep learning applications that can easily combine medical data from various sources. While significant advancements have been made separately in deep learning for cardiac MRI and ECG signals, there remains a significant gap in exploring the integration of cardiac electrical activity (ECG) and cardiac structure (cardiac MRI) using deep learning methodologies. This gap represents a valuable opportunity to develop innovative approaches, that leverage the complementary information from both modalities to gain a more comprehensive understanding of cardiac function and pathology.

The motivation behind this research stems from the need to bridge the gap between deep learning applications in cardiac MRI and ECG signals. By integrating these modalities using advanced deep learning techniques, we aim to develop models that can provide a richer understanding of cardiac structure-function relationships. The successful development of such models holds the potential to contribute to various downstream tasks, including anomaly detection and accurate diagnosis, ultimately leading to improved patient care and outcomes in cardiology. At the same time, the potential to greatly reduce data processing costs by using the ECG rather than the MRI consists of a great importance.

1.2 Contribution

The current diploma thesis seeks to investigate the possibility of deep learning models to close the gap between these two modalities, given the benefits of ECG processing over MRI. The main idea is based on eventually being able to generate the patient's MRI image, while processing the ECG data, in order to obtain more precise information about the heart. The interest of this project relies on the potential of generating from a single ECG the corresponding pseudo-MRI, that offers significantly more information in comparison with just an ECG alone. In this way, we will take use of both the benefits that ECG processing offers over MRI and the enhanced knowledge about the heart that is offered by the created pseudo-MRI.

This study was carried out utilizing a sizable collection of ECG and cardiac MRI datasets from the UK Biobank [26], within the context of the H2020 euCanSHare project (`http://www.eucanshare.eu`), which is under the coordination of the University of Barcelona. This novel strategy aims to overcome the costs, complexity, and processing time constraints of MRI processing by developing a deep learning model capable of processing MRI data through ECG signals. The proposed model seeks to predict cardiac structural dynamics over a heartbeat, based on the electrocardiographic signals, by building links between these two modalities. Upstream applications like anomaly detection and diagnosis are made possible by the integration of multimodal data, which shows potential for improving clinical decision-making.

This project is based upon the work by Radhakrishnan et al. [23], that proposes a methodology in order to generate a pseudo-MRI from a single ECG heartbeat of the same subject. Not only did we attempt to replicate the authors results, but also we went a step further by attempting to do an exploration of the latent space. The representations of the latent space might provide some useful information, that can be used to distinguish abnormal patterns.

1.3 Thesis Organization

This thesis is organised as follows:

- **Chapter 2 - Background and Related Work**: Explanation of the theoretical foundation of the thesis, state of the art and additional terminologies.
- **Chapter 3 - Dataset**: Analysis of our dataset and a thorough explanation of the data pre-processing process.
- **Chapter 4 - Methodology**: Detailed explanation of the different architectures that we used for every model, as well as the process of training.
- **Chapter 5 - Results**: Analysis of the pre-Training, Training, Validation and Evaluation pipelines, presentation of the final model and the results obtained from the experiments. Visualisation of the Latent Space will be provided as well, while observations regarding the outcomes will be made.
- **Chapter 6 - Conclusions and Future Work**: Assessment of the work that has been done and suggestions for further improvement.

Chapter 2

Background and Related Work

In this section we will provide a comprehensive overview of the key background concepts and related work, relevant to the development of deep learning models for integrating ECG and MRI data. The chapter begins by introducing ECG and MRI, providing definitions and general information about these modalities. Subsequently, it explores the concept of Autoencoders, which will play a significant role in the development of the models proposed in this master thesis. Finally, the chapter explores the field of Multi-modal and Cross-modal deep learning models, highlighting their significance and potential applications in the field of medical imaging.

2.1 Electrocardiogram

A diagnostic procedure that monitors the electrical activity of the heart is called an electrocardiogram or ECG [13]. Each heartbeat generates electrical activity, which is distributed throughout the body and can be detected on the skin. The ECG machine uses electrodes placed on the skin to pick up these signals and displays them graphically. During an ECG, temporary electrodes are attached to the chest and limbs using 10 electrical cables. The electrical activity of the heart, which regulates heartbeats, is observed and recorded by these electrodes. The signals detected by the electrodes are transmitted to an electrocardiograph machine. The electrocardiograph records the electrical signals as a series of waves, representing different phases of the cardiac cycle. This graphical representation, called an ECG tracing, offers important details regarding the electrical conduction and rhythm of the heart. The recorded ECG data is then processed by a computer, which analyzes the signals and translates them into a visual waveform pattern. Healthcare professionals can use this pattern to diagnose numerous cardiac problems, identify anomalies, find arrhythmias, assess the heart's conduction system and more. Using electrodes on the limbs and chest, a 12-lead ECG offers a thorough view of the electrical activity of the heart. It gathers data from several cardiac planes, enabling a more thorough study. In Figure 2.1, one can see an example of an ECG signal.

2.2 Magnetic Resonance Imaging

Magnetic resonance imaging or MRI [22] is a non-invasive imaging technology that produces three dimensional detailed anatomical images. It is often used for disease detection, diagnosis and treatment monitoring. It is based on sophisticated

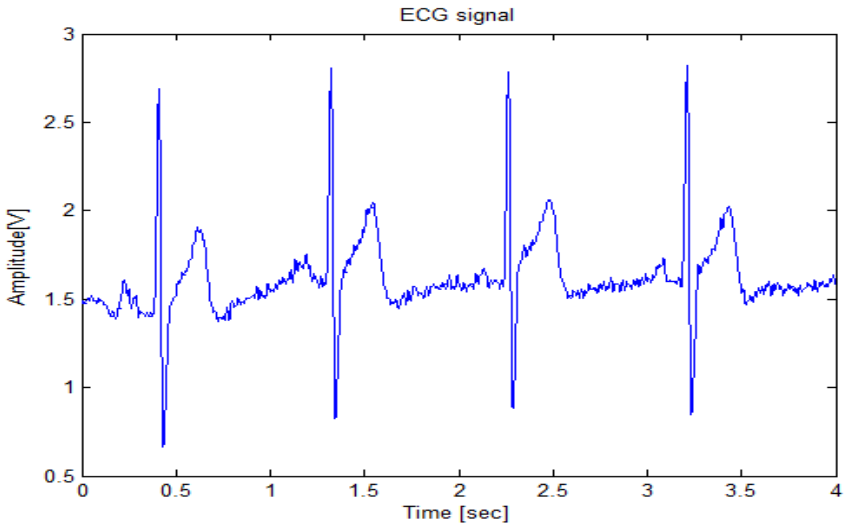


FIGURE 2.1: Example of an ECG signal [9]

technology that detects the change in the direction of the rotational axis of protons found in the water that makes up living tissues. Building upon the basic concept of MRI, Cardiac Magnetic Resonance Imaging (CMRI) specifically focuses on imaging the heart and its surrounding structures. It offers detailed anatomical images of the heart's chambers, valves, blood vessels and the surrounding tissues. The primary use of CMRI is to assess the structure, function and blood flow of the heart. It provides valuable information for diagnosing and monitoring various cardiac conditions. MRIs are widely used in Healthcare in order to detect cardiomyopathies, valvular heart disease, cardiac tumors and masses etc. In Figure 2.2, we present an example of an MRI image of the heart.

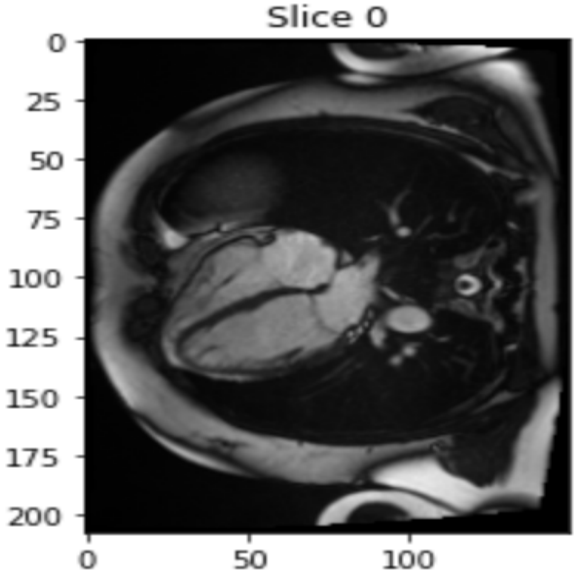


FIGURE 2.2: Example of a cardiac MRI Image

2.3 Autoencoders

What is an Autoencoder?

Autoencoders [18], an essential component of machine learning, are simple learning circuits that aim to transform inputs into outputs with minimal distortion. They are a type of artificial neural network utilized for various unsupervised learning tasks, including dimensionality reduction, feature extraction and data compression. An autoencoder comprises two primary components: an encoder and a decoder. The encoder maps the input data to a lower-dimensional latent space, while the decoder aims to reconstruct the original input data from the learned latent representation. These neural networks have been widely employed in diverse domains, such as computer vision and natural language processing. In the context of this thesis, autoencoders play a crucial role in reconstructing both ECG and MRI modalities, as all the models designed for this project are based on the Autoencoder architecture. The combination of ECG and MRI modalities may result in useful insights by utilizing the capabilities of autoencoders, improving medical diagnosis and therapies.

How Autoencoders Work

The autoencoder [1] aims to reconstruct the original data, as accurately as possible, by learning a lower-dimensional representation of the input data in the Latent Space. This lower-dimensional representation is then utilised in order to generate the output data. The objective is for the desired output to match the input data as closely as possible, by minimizing the reconstruction error between the input and the reconstructed output. The error can be measured by using different loss functions, such as Mean Squared Error(MSE) or cross-entropy loss. Autoencoders, as can be seen in Figure 2.3, consist of three main components: Encoder, Latent Space (Bottleneck) and Decoder [18].

Encoder: The Encoder is a neural network that compresses the input data into a lower-dimensional latent representation. Therefore, the output of the Encoder block consists the Latent Space representation.

Latent Space or Bottleneck: The compressed or lower-dimensional representation of the input data is called Latent Space or Bottleneck. This block contains the encoded values of the input data, capturing the most important features of them in the Latent Space in a compressed form. Hence, the visualisation of the lower-dimensional space (2D or 3D) can significantly contribute not only to the exploration of the data, but also to the extraction of meaningful patterns.

Decoder: The Decoder is another neural network that reconstructs the input data from the Latent Space representation. The input of the Decoder are the encoded data captured in the Latent Space, which the Decoder utilises in order to generate the reconstructed data. The goal of the Decoder is to minimize the reconstruction error between the original input data and the reconstructed data. Regarding the structure, the Decoder usually mirrors the structure of the encoder.

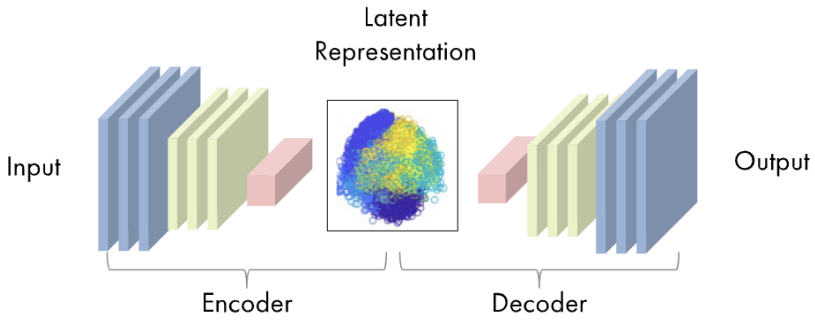


FIGURE 2.3: Basic structure of an Autoencoder for images [17]

Different types of Autoencoders

To easily distinguish the differences between some fundamental Autoencoder types, we are going to provide in this subsection a brief explanation for each type.

- **Simple Autoencoder [1991] [29]:** The Simple Autoencoder is a basic autoencoder architecture that consists of an encoder network and a decoder network. The encoder maps the input data to a lower-dimensional latent representation, while the decoder reconstructs the input data from the latent space. Both the encoder and decoder networks are typically fully connected neural networks, which will be explained in detail in the following paragraph. The hidden layers in the encoder and decoder include fewer neurons than the input and output layers, respectively. By learning a compressed representation of the input data, Simple Autoencoders can be used for tasks such as dimensionality reduction, feature learning and data denoising.

- **Fully Connected Autoencoder [2006] [11]:** The Fully Connected Autoencoder is an autoencoder architecture where each neuron in one layer is connected to every neuron in the subsequent layer. It is a generic form of autoencoder with fully connected (dense) layers. The input data is passed through multiple fully connected layers in the encoder, reducing its dimensionality, and then the decoder uses fully connected layers to reconstruct the data. Fully Connected Autoencoders are flexible and can be used for a wide range of applications. They are particularly useful when working with structured data or as a baseline for more complex autoencoder variations.

- **Convolutional Autoencoder (CAE) [2010] [29]:** The Convolutional Autoencoder is specifically designed for processing structured data, particularly images. Convolutional layers are used by CAEs in both the encoder and decoder components. The encoder applies convolutional filters to extract local features and downsamples the spatial dimensions of the input, gradually reducing its dimensionality. The decoder uses transposed convolutions (also known as deconvolutions or upsampling) to reconstruct the data back to its original shape. CAEs capture spatial dependencies and learn hierarchical representations of images, making them effective for tasks such as image generation, denoising and inpainting.

Works including ECG and MRI Autoencoders

Autoencoder models have been used in various projects, that also included the use of ECG or MRI data. For example, Ojha, Wadhwani, and Wadhwani [20] proposed a method for automatically detecting arrhythmias from ECG signals. Arrhythmias are abnormal activity of the heart functioning and some arrhythmias are harmful to the heart, causing in many occasions sudden death. In this implementation, the ECG was analysed for the diagnosis of arrhythmia beats. The authors developed a classification model that includes a one-dimensional convolutional neural network (1D-CNN) model, based on a Convolutional Autoencoder, that learns the most important values from an ECG signal in order to improve the categorization of arrhythmias. The four different forms of arrhythmic beats were then detected using a classifier. [25] consists another paper that makes use of Autoencoders. This study focuses on using advanced deep learning techniques to detect and classify brain tumors using 512x512 MRI images. The authors proposed a 2D Convolutional Autoencoder neural network to classify three types of brain tumors and healthy brains, using a large dataset of MRI images. The aforementioned model, follows the typical architecture of a U-Net [24] convolutional network, which was the baseline as well for the design of our model that reconstructs the MRI images.

2.4 Multi-modal and Cross-modal DL

Nowadays, increasing amounts of data are being generated and collected at an unprecedented speed. A single modality [7] in the context of Deep Learning in general can refer to any type of input data, such as audio, signals, images or videos. In the field of Medical Imaging in particular, imaging modalities [6] refer to different types of imaging techniques used in medical imaging to visualize the internal structures of the human body. Some examples of medical imaging modalities consist of X-ray, Ultrasound, Computed Tomography and MRI. In some cases, imaging modalities can be combined with other non-imaging modalities such as ECG, clinical history data or genomics data in order to improve diagnosis. One of the latest and most promising trends in Deep Learning research is Multi-modal Deep Learning. In this section we are going to analyse in depth not only Multi-modal, but also Cross-modal Deep learning approach. Additionally, we will also provide some works related to each approach in the field of Medical Imaging.

What is Multi-modal Deep Learning

Multi-modal Deep Learning [19] refers to the utilization of deep learning techniques for processing data from multiple modalities or sources of data simultaneously. The multi-modality refers to the ability to combine sources. The modalities can be of different types, such as signals, images or audio. The model learns to extract features and information from each modality and then combine them in order to make predictions. The main goal of Multi-modal Deep Learning is to extract features from the information gathered from the different available modalities, combining them afterwards in order to enhance the overall performance of the model and improve even further the learning process of the model.

What is Cross-modal Deep Learning

Cross-modal Deep Learning [27] goes a step further in comparison with Multi-modal Deep Learning, by explicitly addressing the relationships and correlations between different modalities. The models are designed to map data from one modality to another, enabling the transfer of knowledge and information across modalities. In other words, Cross-modality refers to the ability of a model to use information from one modality to improve performance in another modality. The main difference between multimodal and crossmodal learning is that crossmodal requires sharing characteristics of different modalities to compensate for the lack of information towards enabling the ability to use data of one modality to predict data of another modality [4].

In order to have a better understanding of how these two notions differentiate, let's think about a scenario: A model is tasked with predicting the emotion of a person in a video. A Multi-modal model would process both the visual frames (images) and the audio simultaneously and use the information from both modalities to make predictions. On the other hand, a Cross-modal model may learn to generate textual descriptions of facial expressions from images and then predict the emotion based on the textual description. While the Multi-modal model used simultaneously the information provided by the different modalities in order to make predictions, the Cross-modal model used the information from one modality (images) in order to improve the performance in another modality (text) and then predicted the emotion of a person based on the text modality.

Regarding the heart, by combining information coming from different modalities, such as ECG and MRI, a more comprehensive understanding of cardiac health is feasible and different aspects of patients anatomy could be revealed. In this direction, we have to bear in mind the fact that Multi-modal systems are based on the idea that different modalities have the potential to provide unique information, significantly contributing to a more holistic understanding. The development of Multi-modal systems in the medical health sector is driven by the goal of improving patient outcomes, enhancing clinical decision-making and accelerating medical research. For instance, these systems can offer a greater understanding of a patient's condition by combining imaging data with clinical notes, enabling more accurate diagnosis and customised treatment strategies. Additionally, they can help with early disease detection, treatment outcome prediction and disease progression monitoring.

The past few years, a lot of research on the exploration of Multi-modalities has been made, in order to benefit from the advantages offered by Multi-modality Deep Learning models. The method proposed by Hammad, Liu, and Wang [10] presents a Multi-modal biometric authentication system, that combines two different biometric modalities: Electrocardiogram (ECG) and fingerprint. The goal of the study is to create a more secure and reliable authentication system, by fusing these two biometric modalities at different levels using Convolution Neural Network. In other words, the objective was to enhance security and improve the overall performance compared to traditional unimodal systems, while at the same time reducing the chances of security threads.

Another innovative approach in Cross-modal integrations consisted [23], which was

the starting point of our project, due to the fact that the authors effectively generated pseudo-cardiac MRI from an ECG input. Looking in more detail, a Cross-modal autoencoder framework was developed in order to process data from two different modalities: ECG and MRI. In this way, a more comprehensive representation of the cardiac state is feasible. The combination of ECG and cardiac MRI data contributed to the demonstration of the advantages of Cross-modal representations through three different applications. Firstly, they improved the prediction of cardiovascular phenotypes using ECGs alone, which is particularly useful considering the abundance of ECG data compared to MRI. Secondly, they successfully imputed cardiac MRIs from ECGs, making it possible to acquire challenging modalities using easily obtainable ones. Lastly, they identified associations between genotypes and general cardiovascular traits. Notably, the authors found that increasing the number of unlabelled ECG and MRI pairs had a more significant impact than increasing the number of labeled MRI data. The study also highlighted the ability of Cross-modal autoencoders to capture confounding factors, as evidenced by the effective prediction of sex and age using MRI Cross-modal embeddings. Overall, this work represents a significant advancement in utilizing Cross-modal modalities for a better understanding of physiological states in the context of cardiovascular health.

Now, let's take a pause and summarize what we have seen so far in Chapter 2. This chapter has provided a comprehensive overview of the fundamental concepts of ECG and MRI modalities, introduced the concept of Autoencoders and discussed the significance of Multi-modality and Cross-Modality in Deep Learning. Therefore, the reader should be equipped by now, with thorough understanding regarding the aforementioned notions. The knowledge gained from this chapter will serve as a solid foundation for the subsequent chapters, where the proposed models will be developed and evaluated in order to achieve the objectives of this master thesis.

Chapter 3

Dataset

For the implementation of this project we needed a large database, which we had access to thanks to the H2020 euCanSHare project (www.eucanshare.eu), coordinated by the Universitat de Barcelona. Within the project, we had access to a large patient cohort from the UK Biobank, a large-scale biomedical database being acquired in the United Kingdom that contains multiple sources of genetic and health data from around 500,000 participants. The database is continually expanded with new information and is globally accessible to approved researchers. It has greatly contributed to the advancement of modern medicine and has enabled several scientific discoveries that improve human health.

To be more specific, the euCanSHare project is a collaborative effort between European and Canadian institutions. It seeks to create a platform for collaboration that enables the fusion of various data types related to cardiovascular health, such as sociodemographic data, biosamples, cardiac imaging and clinical outcomes. Researchers are able to learn more about various facets of cardiovascular health, thanks to this extensive dataset. In other words, the euCanSHare project acts as a platform for collaboration that combines cardiovascular data from several cohorts. The University of Barcelona was involved in coordinating the project, and therefore we eventually had access to a great number of ECG signals and cardiac MRI datasets, accounting for more than 40,000 participants. For this project, we decided to limit the sample size to 10,000 participants to allow for a faster implementation and training. We have to underline that regarding the structure of our project, we created a folder named **UKBB**, inside which we had two more folders: **ecg**, where all the ECG signals were stored and the **NIFTI** folder, where all the cardiac MRI Images were stored.

It is of a great importance to underline the fact that this thesis was based on pair modalities, and more specifically pairs of ECG signals and MRI images. Therefore, this dataset is quite useful for us, due to the fact that for the majority of the patients we had available pairs of ECG signals and the corresponding cardiac MRIs. However, for the patients that an ECG-MRI pair didn't exist on the dataset, because either the ECG or the MRI was missing, we removed those patients from the dataset using the unique identifier. Eventually, we filtered out the common Ids that existed not only in the ecg folder, but also in the NIFTI folder, so that the final dataset will only include ECG-MRI pairs. Moreover, we had available other information related to the characteristics of the subjects of the dataset in a csv file. The mapping between the Dataset and the csv file, was feasible by using again the unique Id for every patient. The most significant features of each subject, among others, can be observed on the following list:

- **feid**: The unique Id number of each patient in the dataset.
- **age**
- **height**
- **weight**
- **sex**: This column records the sex (male or female) of the individuals.
- **bmi**: This column stores the Body Mass Index values of the individuals.
- **date_first_scan**: Contains the dates when the first scan was conducted.
- **date_second_scan**: This column is related to the dates that the second scan took place.
- **Acute MI**: This column indicates whether the patient had an acute myocardial infarction (MI). An acute MI is a serious condition that occurs when the blood supply to part of the heart is blocked, usually by a blood clot.
- **Ventr. Tachy.**: This column indicates whether the patient had ventricular tachycardia at some point, which is a type of abnormal heart rhythm.
- **LV fail.**: This column indicates whether the patient had heart failure with reduced ejection fraction.

In terms of sex, our dataset had a balanced number of Females (23510) and Males (21844) patients. In the following sections, we are going to analyse how we pre-processed the ECGs and the MRIs that we had available for each individual subject.

3.1 Electrocardiogram

To begin with, we are going to display an ECG signal stored from the dataset in Figure 3.1. As you can see, the amplitude of the ECG has approximately range [-60, 200]. Furthermore, the full ECG signal has 5000 samples (x-axis) and some noise also is present. The pre-processing of the ECG mainly concerns the aforementioned features of the signal.

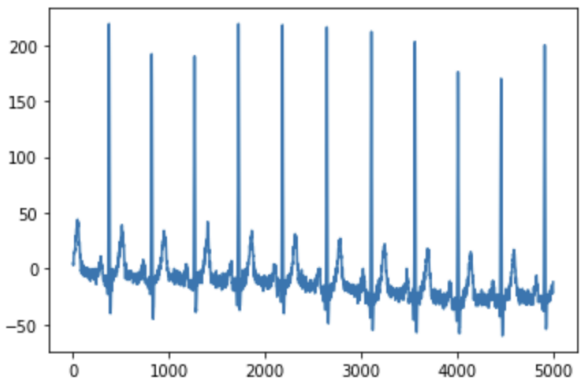
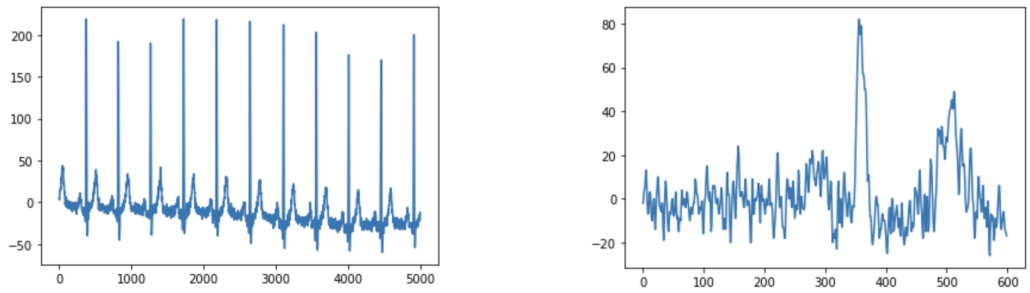


FIGURE 3.1: Example of an ECG Signal

ECG Preprocessing

- **Segmentation**

  Following the implementation by Radhakrishnan et al. [23], we selected the first 600 samples from the full ECG signal, which correspond to the 1.2-second 600-voltage median waveforms derived from the full 10-second ECG. You can notice in Figure 3.2, that after the segmentation the x-axis has a range of [0, 600].



(A) (B)

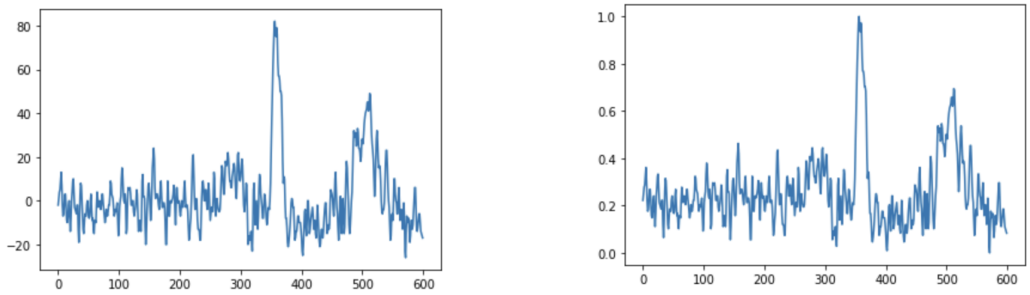
FIGURE 3.2: (A) Before Segmentation (B) After Segmentation

- **Re-scaling**

  The next step is to normalise the ECG signals. For this purpose, we utilised the following formula:

$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x}) + 1e^{-}4} \tag{3.1}$$

  where **x** refers to the ECG signal. The factor 1e-4 was added, because some signals of the dataset had the same min and max value, and therefore we wanted to avoid divisions by zero. After the Normalization process, we can observe that the ECG signal has an amplitude value between 0 and 1. In Figure 3.3, we present the ECG before and after normalization.



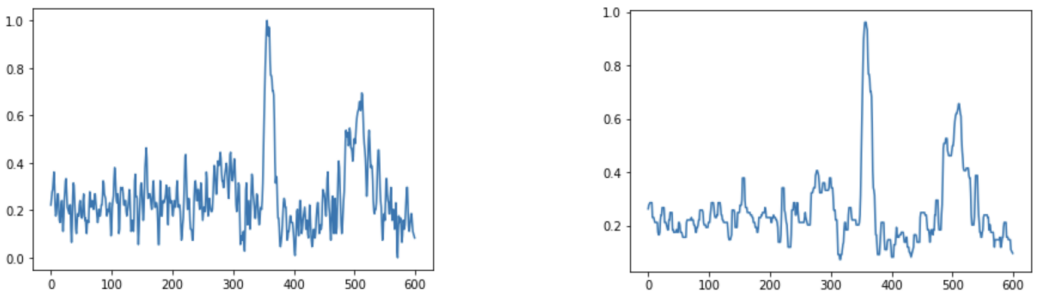
(A) (B)

FIGURE 3.3: (A) Before Normalization (B) After Normalization

- **Denoise**

  Last but not least, we applied a Denoising technique in order to cancel the noise that the ECG signals have. In this way, we expect to be able to acquire

better results. To accomplish that, we created a function named median_filter and we got the results shown in 3.4. It is obvious that the final ECG signal is more clean that the initial one.



(A) (B)

FIGURE 3.4: (A) Before Denoise (B) After Denoise

3.2 Magnetic Resonance Imaging

As far as it concerns the cardiac MRI Images, due to the fact that the computational effort and time needed for the completion of the tasks were extremely big, we decided to select a single slice from the entire 50-slice MRI Image. In that way, not only the data given as an input to the models were extremely reduced, but also the number of parameters of the models as well. Therefore, the computational time was significantly reduced, which was extremely important for us due to time limitations. Apart from that, because MRI images in the dataset had various sizes, we resized all images to 128x128 pixels. In Figure 3.5, you can see an example of an MRI Image after the pre-processing that we applied.

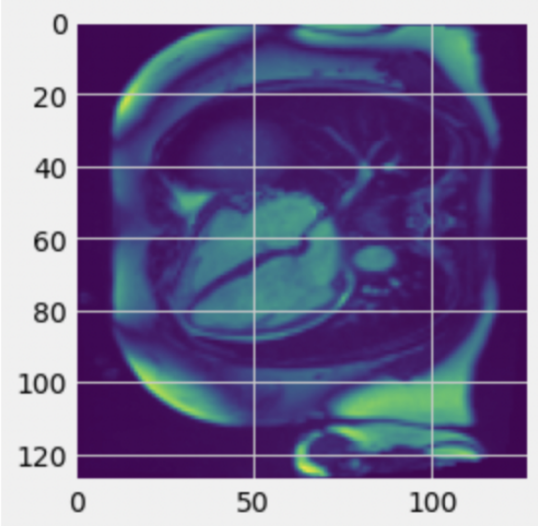


FIGURE 3.5: An example of an 128x128 MRI Image after pre-processing

Chapter 4

Methodology

In this section we are going to describe the methodology employed during this work. To be more specific, we will examine the various models we have developed, including details on their architecture and the loss functions we used. We implemented a Multi-modality autoencoder that had different encoding paths for the different data sources. For ECG data we used a Fully Connected-based autoencoder, while for MRI data we used a 2D Convolutional autoencoder. More details will be given next regarding the specific architecture and training details.

4.1 Fully Connected Autoencoder for ECGs

The first model that we are going to analyse consists the first out of the two models used in order to form the Multi-Modality Autoencoder. This model is a Fully Connected Autoencoder and is responsible for reconstructing ECG signals. The pipeline can be seen in Figure 4.1. Having a batch size of 64, each input ECG signal has a shape of (batch size=64, 600). This model also consists of an encoder and a decoder, which are defined as separate sequential modules. The encoder is responsible for transforming the input ECG signal into a lower-dimensional latent representation. The encoder comprises several linear (fully connected) layers followed by ReLU activation functions and each layer reduces the dimensionality of the input gradually, going from 600 to 256. Therefore, after the ECG Encoder the dimensionality will be reduced from 600 to a 256 feature vector. The decoder takes the latent representation and reconstructs the original ECG signal. In our implementation, the decoder mirrors the structure of the encoder in reverse order, consisting of linear layers and ReLU activations. The decoder layers progressively increase the dimensionality, ultimately reconstructing the ECG signal of size (batch size=64, 600). The final layer of the decoder uses a sigmoid activation function to squash the output values between 0 and 1, which is typical for signal reconstruction tasks. This model was initially pretrained using a group of 5000 ECG signals. More details on the pretraining process of the Fully-Connected Autoencoder will be provided in Chapter 5.

Number of Parameters

In Figure 4.2, one can observe the model parameters that the Fully Connected Autoencoder includes in every layer of the autoencoder as well as the total number of parameters.

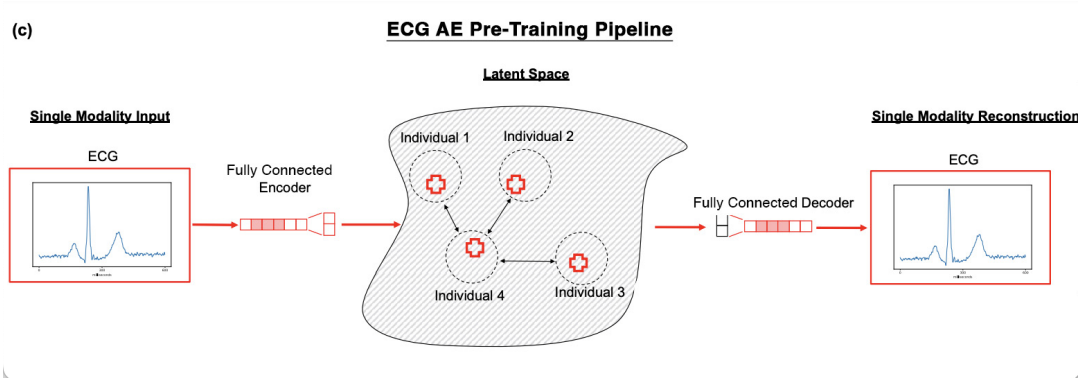


FIGURE 4.1: Fully Connected ECG Autoencoder Pipeline

```
Number of parameters for each layer of my model:
--------------------------------
encoder.0 - 307712
encoder.2 - 205200
encoder.4 - 120300
encoder.6 - 77056
decoder.0 - 77100
decoder.2 - 120400
decoder.4 - 205312
decoder.6 - 307800
```

```
: pytorch_total_params = sum(p.numel() for p in ecg_autoencoder.parameters() if p.requires_grad)
  print("Total number of parameters for my model: ", pytorch_total_params)
```

```
Total number of parameters for my model:  1420880
```

FIGURE 4.2: ECG Fully Connected Autoencoder number of parameters overview

4.2 2D Convolutional Autoencoder for MRIs

The basis for the implementation of our 2D Convolutional Autoencoder model consisted the simple and successful U-Net model, which was introduced in 2015 [24]. The UNet architecture is a convolutional neural network that is widely used for image segmentation tasks, particularly in the field of biomedical image analysis. It was specifically designed to handle the challenges of limited training data and class imbalance often encountered in such applications.

4.2.1 Architecture

The U-Net architecture consists of an encoder path and a corresponding decoder path. The encoder path follows a typical convolutional neural network (CNN) structure, consisting of multiple convolutional and pooling layers that progressively downsample the input image to capture context and extract high-level features. The decoder path, on the other hand, performs upsampling and uses skip connections to concatenate feature maps from the encoder path with the corresponding upscaled feature maps to recover spatial resolution and localize features. It is of a crucial importance to underline that in our implementation we removed the skip connections, which are visualised with the gray arrows, due to the fact that we do not desire a connection between the Encoder and the Decoder of our model. In Figure 4.3 an overview of the classic UNET model architecture is presented.

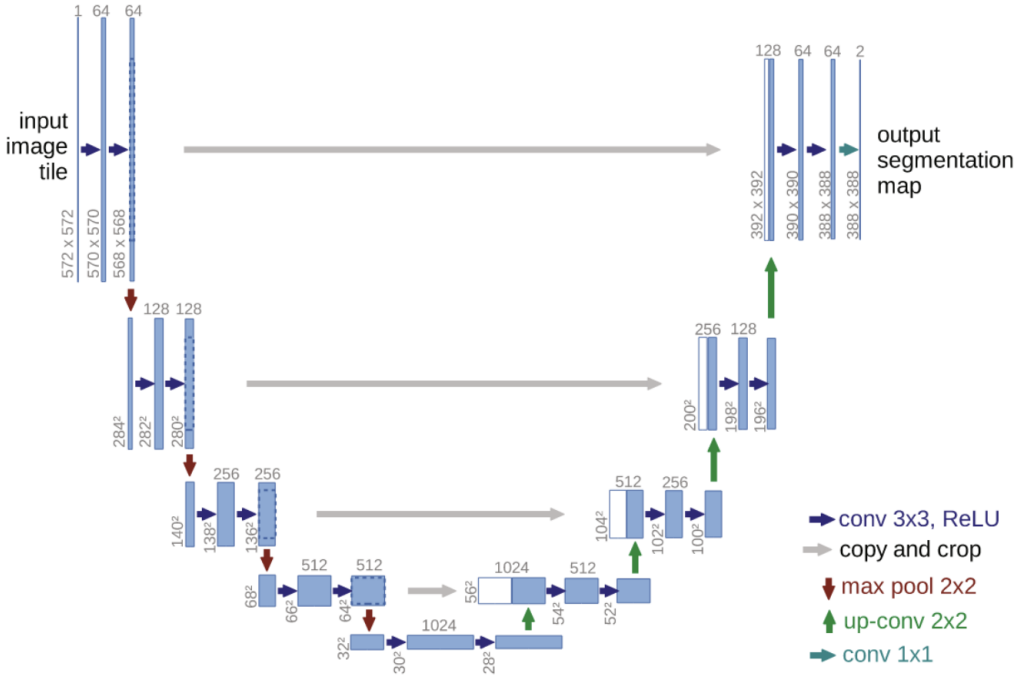


FIGURE 4.3: UNET Model Architecture [24]

We will now try to break down the different blocks of the UNET model. The basic building block of the UNet model, shown in Figure 4.4, consists of three main components: a convolutional operation, a normalization layer, and an activation function. This building block is typically repeated multiple times(x2) in the model to create a deep network architecture.

- **Convolutional Operation**: The fundamental part of the building block is the convolutional operation. In order to extract relevant features, it applies a series of trainable filters to the input data. By sliding over the input, each filter carries out a convolution operation by computing the dot product between the filter weights and the corresponding input values. This operation helps to capture spatial patterns and features in the data.
- **Normalization Layer**: The output of the convolutional operation is normalized by the normalization layer. By ensuring that the input values are within an acceptable range, it contributes to the stabilization and improvement of the training process. Batch Normalization is a frequently used normalization layer that adjusts the output by dividing by the mini-batch's standard deviation and subtracting the mean.
- **Activation Function**: The building block becomes non-linear due to the activation function. The output is element-wise subjected to the activation function following the convolutional operation and normalization. It makes the model more expressive and enables the model to learn complex relationships. ReLU (Rectified Linear Unit) and its variants, such as Leaky ReLU or PReLU, are frequently used activation functions that hold positive values constant while setting negative values to zero.

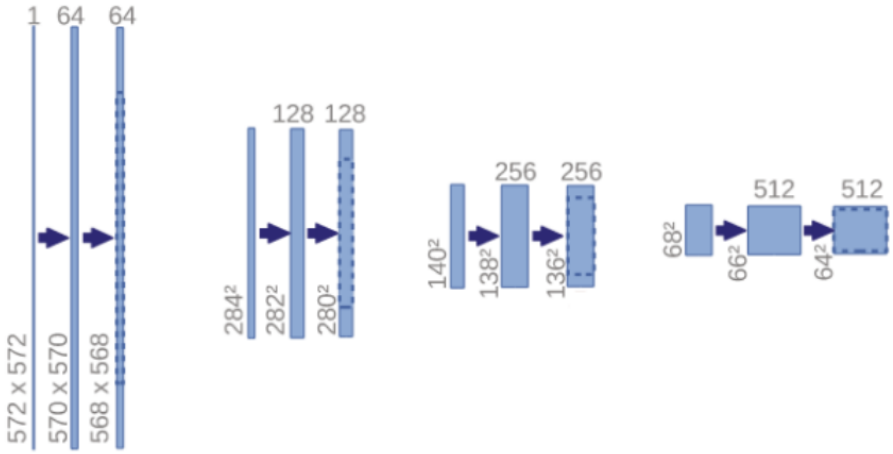
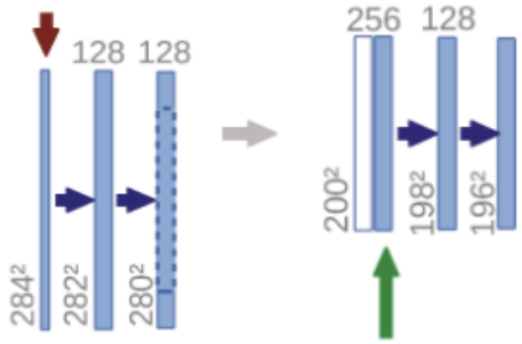


FIGURE 4.4: Building Block Architecture

A downsampling operation, presented in Figure 4.5, is applied to reduce the spatial resolution of the feature maps. The most common downsampling operations used in the UNet model are max pooling and strided convolution. Max pooling divides the feature maps into non-overlapping regions and selects the maximum value within each region, effectively reducing the spatial dimensions. Strided convolution, on the other hand, performs convolution with a larger stride, resulting in a smaller output size. By stacking multiple downsampling blocks together, the UNet model gradually reduces the spatial resolution while increasing the number of channels, allowing it to capture both local and global features at different scales. This downsampling process helps to extract and encode the hierarchical information in the input data, which can later be used for upsampling and reconstructing the original input in the decoder part of the UNet model. In our case we have 3 consecutive downsampling blocks, reducing the spatial dimensions from 128x128 that are the input cardiac MRI Images to 16x16. After that, a Max Pooling operation with Kernel size 16 is applied, reducing even more the spatial dimensions from 16x16 to a 1x1 vector. We have also to mention the fact, that while downsampling, the channels increased from 64 to 256 channles on the feature space. On the other hand, we have the Upsampling block, displayed in Figure 4.5, which increases the the spatial dimensions and starts the procedure of reconstructing the 128x128 image. In our implementation, we used 7 Upsampling blocks followed at the end from a 1x1 Convolution layer, upsampling eventually the initial 1x1 vector to a 128x128 reconstructed MRI Image. In this case, the number of channels was reduced from 256 to 64. The updated UNET model has a total number of 27338497 parameters.

4.2.2 Pipeline

In Figure 4.6 we present the Pipeline for the 2D Convolutional Autoencoder for the cardiac MRI Image reconstruction. As you can see, the input is a 128x128 MRI Image, that Preprocessing techniques are applied, as mentioned in Chapter 3. Then, this processed 128x128 Image is encoded in a 1x1 vector which is represented with blue colour in the latent space. Subsequently, the 1x1 vector is Decoded and the output is the generated 128x128 pseudo-MRI Image. This model was separately pretrained as well, like we did with the Fully-Connected Autoencoder, but this time



(A) (B)

FIGURE 4.5: (A) Downsampling Block (B) Upsampling Block

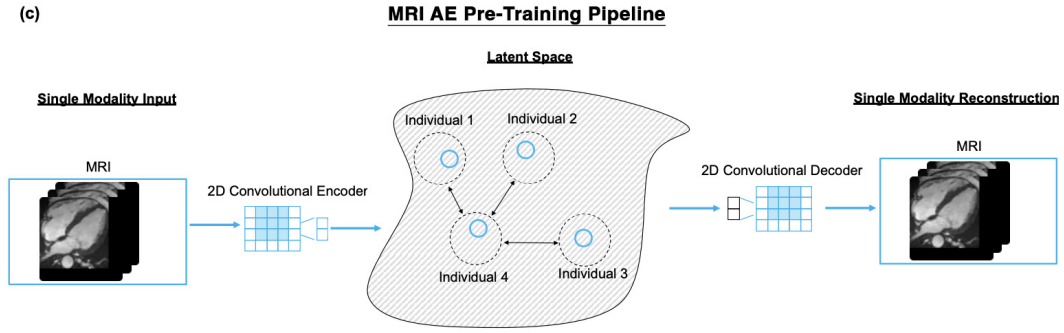


FIGURE 4.6: 2D Convolutional MRI Autoencoder Pipeline

using 1000 MRIs. Detailed information related to the pretraining process of the 2D Convolutional Autoencoder will be provided in Chapter 5.

4.3 Multi-Modality Autoencoder

In this section we are going to explore the Multi-modality Autoencoder. At first, paired modalities will be explained, a detailed pipeline that we used for the training process will be provided as well as the architecture of the model.

4.3.1 Training Pipeline

To begin with, we are going to analyse the pipeline that we used in order to train the Multi-modal Autoencoder. An overview of our Multi-modal autoencoder framework can be seen in Figure 4.7. Having a general overview of the Autoencoder model, we notice that the basic elements are:

- Input Modalities (ECG and cardiac MRI pairs)
- Encoder
- Latent Space

- Decoder
- Output Modalities (Reconstructed ECG and MRI for the same individual)

For our implementation we used two different modalities as inputs to our Multi-modal Autoencoder model: ECG signals and MRI Images. To be more specific, we have trained our model on ECG and cardiac MRI pairs from the UK biobank. We have to underline the fact that the ECG-MRI pairs correspond to a single individual. The model is trained on paired-modalities, meaning the ECG-MRI pairs of each subject. By combining information coming from different modalities in our case, a more comprehensive understanding of cardiac health might be feasible and different aspects of patients anatomy could be revealed.

The first step of our Training pipeline consists the Preprocessing of our inputs (both ECG and MRI), as we previously explained in Chapter 3. After that, each of the inputs will go through the Encoder part. The ECG signals will go through the Fully Connected ECG Encoder and the MRI Images through the 2D Convolutional Encoder. Hence, different data sources have different encoding paths. Modality specific encoders map data modalities into a shared latent space in which a contrastive loss is used to enforce the constraint that paired samples are embedded nearby and further apart from other samples. Contrastive Loss will be further explained later in this Chapter. For both inputs, the output of the encoder would be a 256 feature vector in the Latent Space. The Latent Space is represented with grey colour in Figure 4.7, where with a red cross is the 256 feature vector of the ECG encoded signal and with a blue circle is the 256 feature vector of the MRI encoded image for each subject.

Then, the next step in our Training pipeline is the Decoder part of each model. Modality specific decoders are then used to reconstruct modalities from points in the latent space. In more detail, in this step the 256 feature vectors are transformed again to ECG signals or MRI cardiac images, contributing to the successful reconstruction of the ECG as well as the MRI. Overall, the final outputs of the model will eventually be: the Reconstructed ECG signal and the Reconstructed MRI cardiac Image.

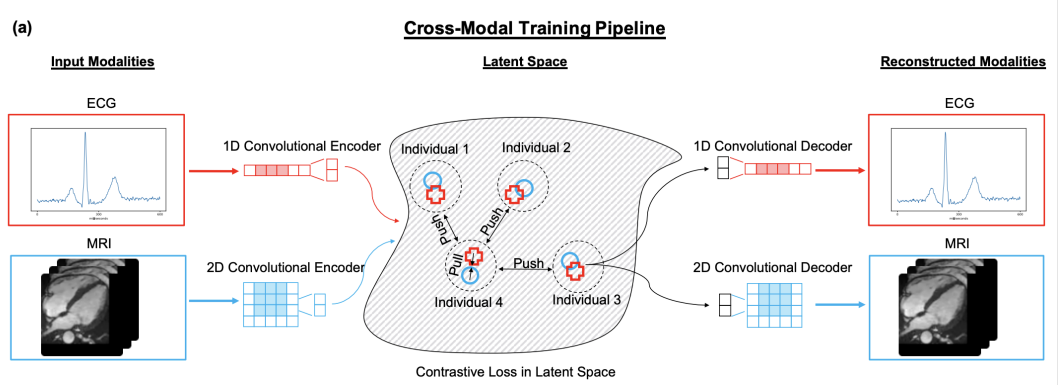


FIGURE 4.7: An overview of our Multi-modal autoencoder framework, which also consists a visualization of our training pipeline. [23]

4.3.2 Architecture

As we observe in Figure 4.7, the Multi-Modal Autoencoder is a combination of two separate autoencoder models, that were explained in detail previously in the sections 4.1 and 4.2 accordingly: a Fully Connected Autoencoder for processing ECG signals, which is shown with red colour, and a 2D Convolutional Autoencoder, which is responsible for processing MRI images and is shown with blue colour at the pipeline.

The Multi-Modality Autoencoder model combines the capabilities of an ECG autoencoder and an MRI autoencoder to jointly learn representations from both modalities and reconstruct the input data. Moreover, we combine the ECG encoder and decoder with the MRI encoder and decoder. In further detail, the model takes an ECG signal and an MRI image as inputs. To obtain the encoded representation, the ECG signal is processed through the ECG encoder. The ECG signal is then regenerated using the encoded ECG representation. Similarly, the MRI image is passed through the MRI encoder to obtain the encoded representation, which is then decoded to reconstruct the MRI image. The forward method returns the encoded representations of both the ECG and MRI inputs, as well as the reconstructed ECG signal and MRI image.

In Figure 4.8, we are able to see the number of parameters of the model for every layer. We have to underline the fact that the total number of parameters for our Multi-Modality Autoencoder is: 11981905.

```
Number of parameters in the ECG Encoder: 710268
Number of parameters in the ECG Decoder: 710612
Number of parameters in the MRI Encoder: 7051968
Number of parameters in the MRI Decoder: 3509057
-------------------------------------------------
Number of layers in the model: 180

Total number of parameters for my model:  11981905
```

FIGURE 4.8: Multi-Modality Autoencoder number of parameters overview

4.3.3 Training Procedure

In our implementation, the above pipeline consisted the Training pipeline, as we used 4627 ECG-MRI pairs for different individuals provided by the UK biobank dataset as inputs to the model. For this Multi-modal model we used the following parameters:

- **Loss Function**: The L1 loss function was used, also known as the Mean Absolute Error, which is used to measure the average absolute difference between predicted and true values.
- **AdamW Optimizer [15]**: AdamW is an optimization algorithm derived from Adam, a popular optimizer in deep learning. It combines the benefits of adaptive learning rates with weight decay, an L2 regularization technique. By adding a penalty term based on the magnitudes of model weights, weight decay helps

control model complexity and prevents overfitting. This regularization encourages smaller weight values, leading to smoother and more generalizable models.

- **Learning rate**: The learning rate that we used for training is 0.001.
- **Batch size**: A batch size of 64 was used for Training, which means that the model is processing 64 training examples at a time.

Regarding the Train-Validation split of our dataset, we have to mention the fact that the initial 5000 dataset was split with a proportion of:

- **80% for Training** : The training dataset is composed of 3701 ECG-MRI pair modalities.
- **20% for Validation**: The validation dataset, on the other side is composed of 926 ECG-MRI pair modalities.

The model has been trained for 30 epochs and the model parameters after Training have been saved, so that they could be loaded afterwards for the final Evaluation of the model. For every epoch, after the training of our training dataset has been completed, a Validation starts to take place on the validation set, which contains unseen data compared to the training set. In this way, we are allowed to assess how well our trained model performs on unseen data in every epoch and determine the models generalization potential by assessing it on a different dataset. In other words, it gives us insights on the model's effectiveness and potential problems like overfitting or underfitting by demonstrating how the model is expected to behave when new data are provided.

4.4 Loss Functions

The training methodology in this project involves the use of a multi-modal autoencoder to reconstruct original modalities and ensures that the representations of modalities corresponding to the same sample are embedded nearby in the latent space. In the following section we are going to discuss about the Reconstruction, Contrastive and Total Loss. To begin with, the total loss function used for training consists of a linear combination of two components: a reconstruction loss (LRec) and a contrastive loss (LContrastive). For our experiments we will use: $\lambda = 0.1$.

$$L(\{X(j), f_j, g_j\}) = L_{\text{Contrastive}}(\{X(j), f_j\}) + \lambda L_{\text{Rec}}(\{X(j), f_j, g_j\})$$

Reconstruction Loss (LRec)

The Reconstruction Loss [23] is responsible for reconstructing the original modalities. It measures the difference between the original input samples and the reconstructed samples obtained from the autoencoder. The reconstruction loss (LRec) is calculated as the sum of squared Euclidean distances between the original and reconstructed samples.

$$L_{\text{Rec}}(\{X(j), f_j, g_j\}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| x^{(i,j)} - g_j(f_j(x^{(i,j)})) \right\|^2$$

Contrastive Loss (LContrastive)

Contrastive loss [23, 28, 3, 5] is a commonly used loss function in deep learning for tasks involving paired modalities, such as cross-modal retrieval. In the context of paired modalities, we have multiple data modalities available for the same sample, and the goal is to learn a shared representation space that captures the relationships between these modalities. The contrastive loss aims to ensure that the representations of modalities corresponding to the same sample are embedded nearby in the latent space, while pulling apart the representations of different samples or modalities. This loss encourages similar samples or modalities to have smaller distances in the learned representation space, promoting better alignment. The contrastive loss is typically computed by comparing pairs of samples or modalities within a batch. It encourages the positive pairs (pairs of modalities from the same sample) to have small distances and negative pairs (pairs of modalities from different samples) to have large distances. This is achieved by using similarity measures, such as dot products or cosine similarities, and applying a softmax function to obtain probabilities that represent the similarity between pairs. In our implementation we used the cosine similarity.

To develop a multi-objective loss function for Multi-modal autoencoder training, the contrastive loss can be paired with additional loss functions, such as reconstruction loss. Hyperparameters like lambda can be used to regulate the weights of these different loss components to balance their contributions during training. The Reconstruction Loss is responsible for reconstructing the original modalities. It measures the difference between the original input samples and the reconstructed samples obtained from the autoencoder. The LRec is calculated as the sum of squared Euclidean distances between the original and reconstructed samples. Overall, the Contrastive Loss ensures that the representations of modalities corresponding to the same sample are embedded nearby in the latent space, by using a contrastive loss formulation to push similar samples closer together and separate dissimilar samples. The contrastive loss is calculated as follows:

$$L_{\text{Contrastive}}(\{X(j), f_j\}) = \sum_{j_1=1}^{m} \sum_{j_2=1}^{m} \mathbf{I}_b \log \sigma_{j_1 j_2}(S_{j_1 j_2} e^t) + \mathbf{I}_b \log \sigma_{j_2 j_1}(S_{j_1 j_2} e^t)^2$$

Chapter 5

Results

In this section, the different experiments and results from the different pipelines, that we have explained in Chapter 4, will be exposed. Additionaly, the Final Cross-modal Autoencoder pipeline will be presented. To be more specific, we will present the results for the following procedures:

- Pretraining of the ECG Autoencoder
- Pretraining of the MRI Autoencoder
- Training of the Multi-Modality Autoencoder
- Evaluation of the Final Cross-modal Autoencoder

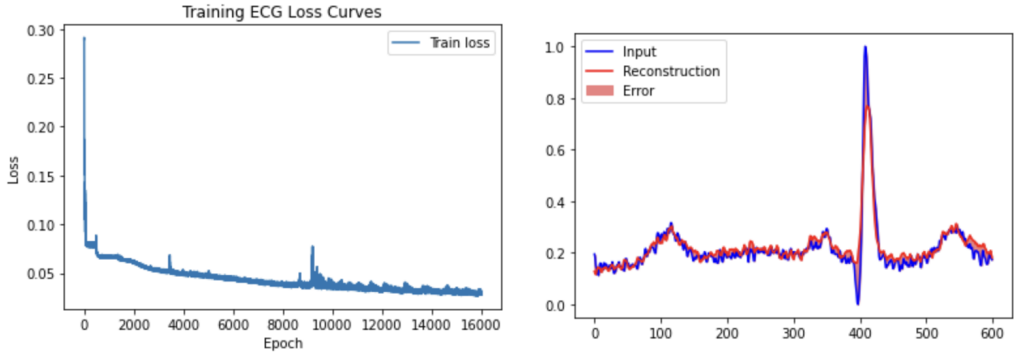
It is of a great importance to underline the fact that for our experiments, the initial implementation was edited in order to use a GPU (Graphics Processing Unit), provided by the University of Barcelona, instead of a CPU (Central Processing Unit). This was possible, due to the fact that Python's framework Pytorch, which was used for all the implementations, gives the user the opportunity to speed up deep learning computations by using GPUs [21]. GPUs are specifically designed for parallel computations and boast a large number of cores, making them highly efficient for the computationally intensive matrix calculations involved in training neural networks. This is especially advantageous when dealing with extensive datasets, such as images, which tend to be memory-intensive and require complex operations. By utilizing a GPU's power, we can achieve remarkable speed-ups in the training and evaluation process of deep learning models. GPUs excel at parallel processing, allowing multiple operations to be executed simultaneously. This capability significantly accelerates the computations required for tasks like matrix multiplications and convolutions, which are fundamental operations in neural network training. Consequently, the training time for deep learning models is substantially reduced compared to running them solely on a CPU.

5.1 ECG Signal Autoencoding

In the beginning, we will analyse the results that we have got after pre-training the Fully Connected Autoencoder model (4.1), that is responsible for reconstructing the ECG signals.

Pre-Training

In the pre-train process, which obviously took place before the Training of the combined Multi-Modality Autoencoder, after preprocessing the ECG data as explained in Chapter 3, we used 5000 ECG signals as inputs in order to pre-train our model for the ECGs. Regarding the parameters (loss function, optimizer, learning rate), we used exactly the same parameters with the parameters we used for the Training of the Cross-Modal Autoencoder as presented in 4.3.3, except from the batch size which instead of being 64 it was 512. In Figure 5.1a, we can observe the Loss Function curve for the ECG pretraining process. From the plot of the Loss Function we notice that our ECG model is learning very fast and this is reflected from the steepness of the curve, which means that the loss is reducing dramatically during the first epochs. Afterwards, the loss continues to decrease ,but at a slower rate , until finally getting stable, which means than our model is not improving anymore. We have to mention the fact, that this is a desirable Loss function plot, because it means that our model is learning extremely fast, significantly reducing the loss eventually. Regarding the reconstructed ECG signal, we can see in Figure 5.1b, that with the blue colour is presented the initial ECG signal and with the red colour is displayed the reconstructed ECG signal. The reconstructed ECG signal is considered to be very accurate, following the same pattern as the original ECG signal, even the peak of the ECG waveform. After the pre-training process is over, we save the pretrained ECG Fully Connected Autoencoder model parameters. These ECG model parameters will be loaded before starting the Training of the Multi-Modality Autoencoder model.



(A) Loss Function Plot for ECG pre-Training

(B) Input and Reconstructed ECG signal after pre-Training

FIGURE 5.1: ECG Pre-Training Results

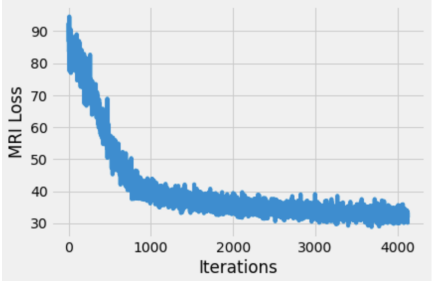
5.2 MRI Image Autoencoding

After presenting the results of ECG Signal Autoencoding, let's take a look at the outcomes obtained from the MRI Autoencoder model (4.2). We have to remind, that the aforementioned model is responsible for regenerating the MRI.

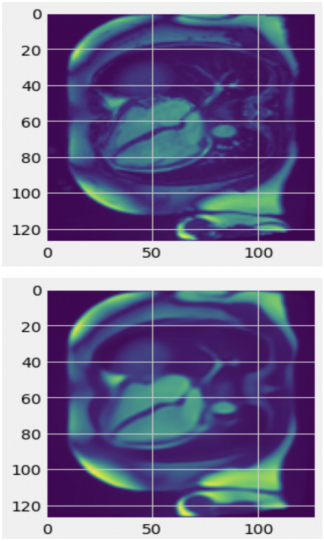
Pre-Training

Pre-training was applied not only to the ECG, as we have seen previously, but also to the MRI Images. To be more specific, the parameters (loss function, optimizer, learning rate) used for this procedure were exactly the same with the ones that we used for the Training of the Multi-Modality Autoencoder, apart from the batch size, that we gave the value of 32. For the pre-training we used 1000 MRI Images as an input to the model, which was trained for 40 epochs. It is worth noticing that during the MRI pre-training we shuffle the data to avoid memorization and thus improve generalizability of the model. In more detail, shuffling the data refers to randomly rearranging the order of the samples in our dataset. Memorization occurs when a model learns to map specific inputs to specific outputs, without truly understanding the underlying relationships. By shuffling the data, we ensure that the model is exposed to a variety of samples in different orders during training. This helps the model to learn the underlying patterns in the data, rather than relying on the samples particular sequence. The dataset's inherent biases and ordering are disturbed by shuffling, contributing this way to better generalization and adaptability of the model to unseen data. In Figure 5.2a we display the Training Loss curve that we got after pre-training, which shows that the 2D Convolutional Autoencoder shows a significant improvement as the loss is reducing rapidly during some iterations. We also observe that at the last iterations the Loss curve is stable, which means that the model is not improving anymore and we have to stop the pre-train process in order to avoid overfitting.

Finally, we present an example of an original MRI Image, accompanied by the Reconstructed MRI cardiac Image in Figure 5.2b. As one can see, the reconstruction of the MRI image is quite satisfying as the shape of the original MRI image has been accurately reconstructed and, in general, the reconstructed image is very similar to the original one. However, finer details tend to be blurred in the reconstructed image.



(A) MRI Loss Function



(B) Original(Up), Reconstructed(Down) MRI after pre-Training

FIGURE 5.2

5.3 Multi-Modality Autoencoding

In the following section, we are going to present the results that we obtained after following the Methodology 4.3 for the Multi-Modality Autoencoder. We have to bear in mind the fact that this model has 2 inputs (Original ECG Signal and MRI Image) and 2 outputs (Reconstructed ECG Signal and MRI Image).

Training

The following results were obtained after the training process was complete. Regarding the ECG signals reconstruction, we present you in the same plots in Figure 5.3, some examples of the Original ECG signals accompanied by the Reconstructed ECG signals. As you can observe, the Reconstructions of the ECG precisely replicate the original ECG waveforms.

Apart from the Reconstructed ECG, we are going to present the Reconstructed MRI's as well. In 5.5a, we have captured an example of a Reconstructed MRI Image that we obtained after Training, compared to the Original MRI Image of the subject. We observe that the shape of the MRI Image has been correctly recreated, while also some details of the heart are depicted with various tonalities.

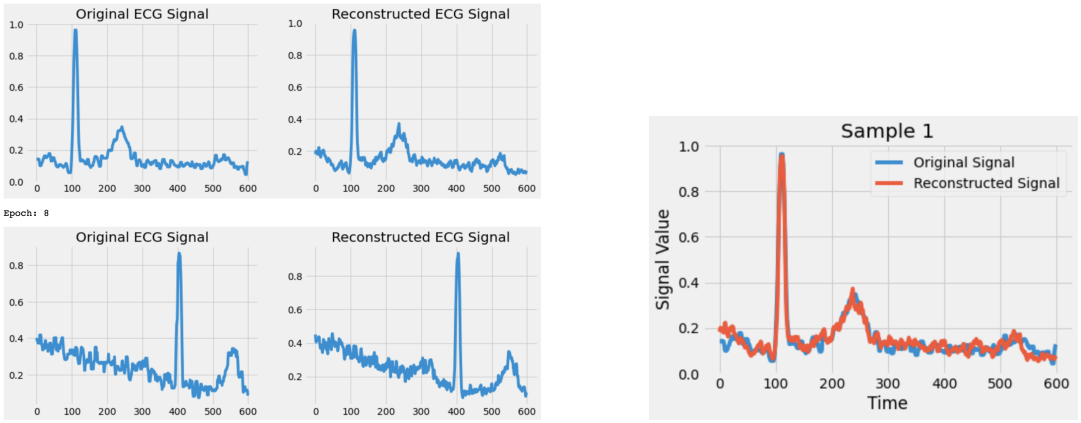


FIGURE 5.3: ECG Reconstruction Results

Training Loss Function

In Figure 5.4 one can see the waveform of the Training Loss Function. According to the preceding explanation at section 4.4, this Loss Function was the result of adding the Reconstruction (LRec) and Contrastive (LContrastive) Losses. Additionally, the loss function for training seems to first decline quickly before remaining stable for the last iterations. The significance of this plot is focused on visualizing the point at which the model stopped becoming better, which is also the point at which we terminated the training procedure.

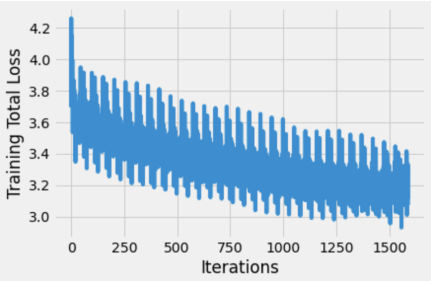
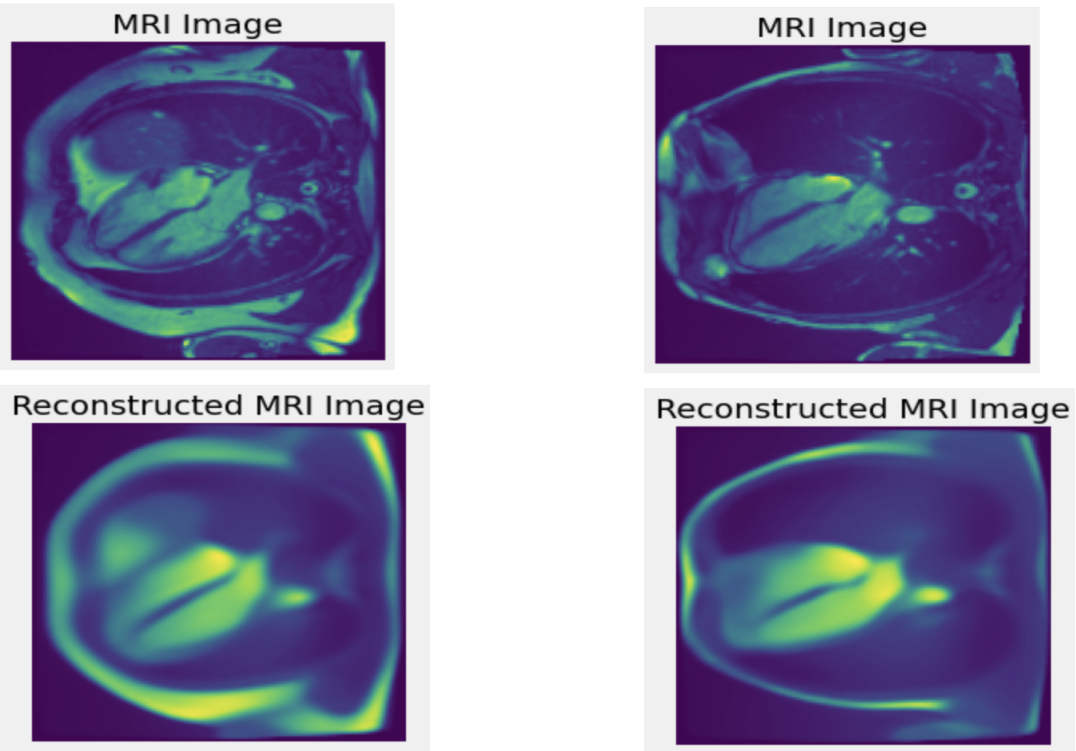


FIGURE 5.4: Training Loss Function

Validation

For each epoch, after training is completed, a Validation step is performed. During validation, the model is evaluated on a separate validation dataset that is not used for training. Validation has various positive aspects, such as evaluating models performance. More specifically, through validation we can evaluate how effectively the model generalizes to new unseen data. On the validation set, we may estimate the model's performance and determine if it is overfitting or underfitting by calculating various metrics or loss values. In Figure 5.5b a Reconstructed MRI Image over the Validation set is displayed. The reconstruction is still blurry but is similar to the reconstruction for a training sample, underlying the fact that the model is learning.



(A) Training (B) Validation

FIGURE 5.5: MRI Reconstruction Results

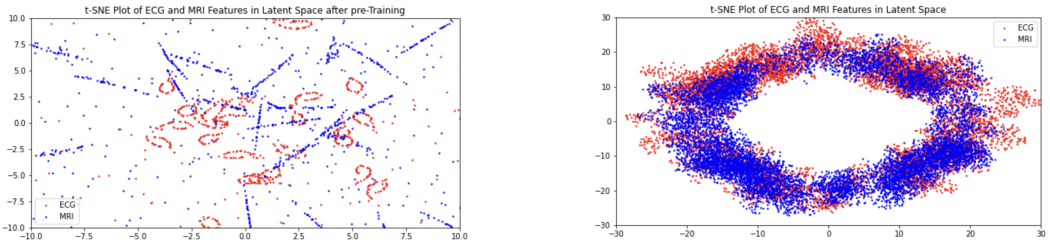
5.4 Visualisation of the Latent Space

As previously explained in Section 2.3, the latent space in our case refers to the compressed representation of the ECG signals and MRI images. The encoder part of the autoencoder learns to map the high-dimensional input data (ECG and MRI) to this lower-dimensional latent space, while the decoder part learns to reconstruct the original input data from the compressed latent representations. In order to have a better understanding of our model's behaviour, we are going to visualise the Latent Space of the encoded ECG and MRI. To achieve this, we used the t-distributed stochastic neighbor embedding, also called t-SNE [16], Latent Space Representation. T-SNE is a dimensionality reduction technique that is particularly well-suited for visualizing high-dimensional data into two or three dimensions. The t-SNE latent space representation refers to the visualization of the latent space using the t-SNE technique and is extremely useful for understanding the organization of the data points in the latent space and how the autoencoder learns to separate or group these data points based on their inherent structure or relationships. By applying t-SNE to the encoded ECG signals and MRI images, we are able to create a 2D or 3D representation of the latent space that can be easily visualized and analyzed. In our case, we implemented a 2D Visualisation of the Latent Space.

In more detail, we wanted to visualise the Latent Space of the Encoded ECG Signals and MRI Images before and after the Training process. Visualizing the latent space using t-SNE before training the autoencoder can help you understand the initial organization of the data points. At this stage, the latent representations may be randomly distributed without any clear structure or grouping, as the autoencoder has not yet learned to encode the input data effectively. On the other hand, by visualizing the t-SNE representation of the latent space after training, we will be able to observe how the autoencoder has learned to organize the data points and bring ECG signals and MRI images that refer to the same subject closer together. This is especially useful when using a loss function that incorporates both Reconstruction and Contrastive Losses, as it promotes the grouping of similar and the separation of dissimilar data points, respectively. In this way, useful conclusions could be extracted especially regarding the functionality of our Contrastive Loss Function, which was presented in 4.4. In general, visualizing the t-SNE representation of the latent space before and after training, can offer insightful information about the autoencoder's learning process and the way it arranges the input data in the latent space. The effectiveness of our model, the selected loss function, and perhaps the identification of possible areas for further improvement are all clarified in this way.

Therefore, focusing on our implementation, we present in Figure 5.6 the t-SNE Latent Space Representations that we got before (5.6a) and after (5.6b) the Training of our Multi-Modality Autoencoder. We have to underline the fact that the ECG are displayed with red colour, while the MRI are presented with blue colour. As can be observed, it is obvious that before Training, the ECG and MRI encoded points are randomly distributed in the Latent Space. Therefore, we can come to the conclusion that in general the ECG and MRI of the same subject were quite distanced to each other for every patient. In addition, after training the Multi-Modality Autoencoder, we notice that the Contrastive Loss significantly contributed to reduce the distance between the ECG and the MRI for all the patients, bringing the corresponding points in the Latent Space closer. We can also say that both ECG and MRI points, are not anymore randomly distributed in the space, creating a distribution which reminds

us of a circle with a gap in the center. Overall, the plots that we have are giving us the indication that the Contrastive Loss that we created is successfully working, grouping together the pairs of ECG and MRI while training the Autoencoder. In this way, t-SNE underlines the improvement and the fact that our Autoencoder model is learning during training.



(A) Before Training (B) After Training

FIGURE 5.6: Latent Space t-SNE Visualisation of the encoded ECG's and MRI's

5.5 MRI Image Reconstruction from ECG signal

The ultimate target of this project was actually to be able to have a clear understanding over the heart in general and more specifically the MRI Images, given only the ECG Signals. Therefore, we present the final Cross-modal model, which is capable of producing an MRI Image of a subject, given a single ECG signal as an input of the model. In Figure 5.7 we present the Pipeline of this final Cross-modal Autoencoder Model, which aims from an ECG signal to create the corresponding MRI. For this purpose, we combined the Encoder quantity of the ECG Fully Connected Autoencoder with the Decoder quantity of the 2D Convolutional MRI Autoencoder.

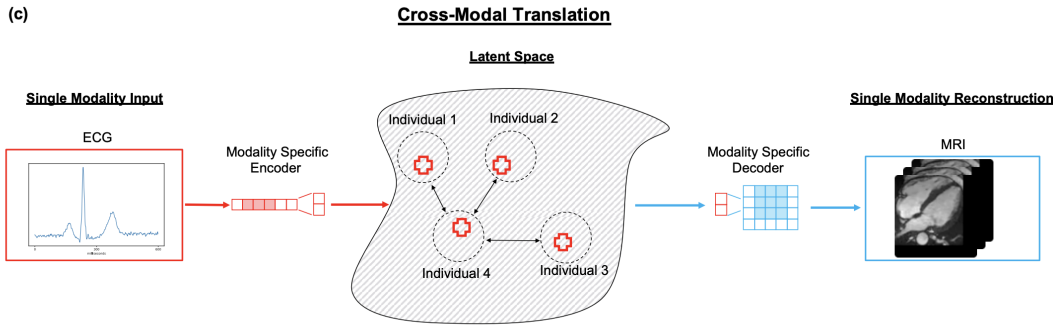


FIGURE 5.7: Final Cross-modal Model Pipeline [23]

Evaluation

In Figure 5.8 we can see the number of parameters for this combined model, which in total are: 4219325.

```
Number of parameters for each layer of my model:
--------------------------------
ecg_encoder.0 - 307712
ecg_encoder.2 - 205200
ecg_encoder.4 - 120300
ecg_encoder.6 - 77056

pytorch_total_params = sum(p.numel() for p in model.parameters() if p.requires_grad)
print("Total number of parameters for my model: ", pytorch_total_params)

Total number of parameters for my model:  4219325
```

FIGURE 5.8: Final Model Number of Parameters

For the evaluation of this final model, we utilised completely new and unseen data, which were approximately the 20% of the Training Data. In Figure 5.9 we present you an example of an ECG signal from our Evaluation dataset, which was given as an input to this model.

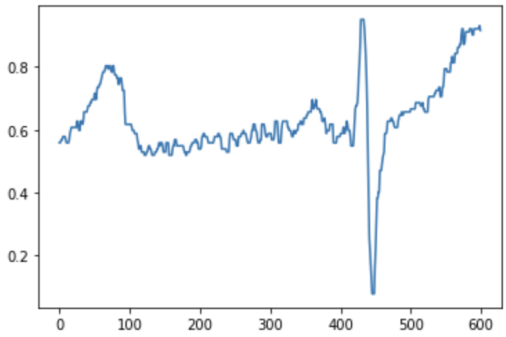
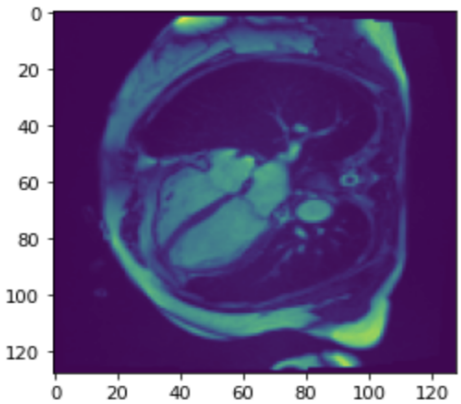
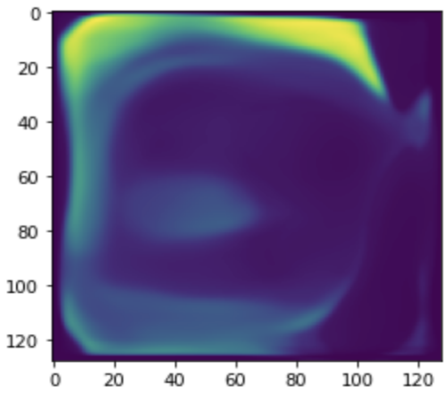


FIGURE 5.9: ECG Input to the model

We display in Figure 5.10b the pseudo-MRI Image that was generated by our model and was the final output of the model. Furthermore, in Figure 5.10a we present for the same subject, the Original MRI Image that we have available in our dataset and constitutes the real MRI Image. Hence, we tried the Reconstructed Image to be as close as possible to the Original Image. As we notice, the reconstruction is not very accurate, despite the fact that we can observe a very abstract representation of the heart and it seems like the model is trying to recreate the shape of the original MRI.



(A) Original MRI



(B) Reconstructed MRI

FIGURE 5.10: MRI Reconstructed Evaluation Results

Chapter 6

Conclusions and Future Work

This project deals with developing a deep learning implementation that can combine medical data from two different sources. Time series data, and more specifically ECG, were combined with MRI in order to develop models that will provide a richer understanding of the relationships between those two modalities. The general goal of this project was eventually to be able to predict cardiac structural dynamics over a heartbeat based on the information extracted from the ECG signals. This is of a great importance, due to the fact that MRI processing is a quite expensive, difficult and time-consuming process in addition to ECG. Apart from that, the possibility of generating from an ECG the corresponding pseudo-MRI, , which provides considerably more information than an ECG alone, is what makes this study interesting. In order to achieve this goal, our objective was to built several deep learning models using different architectures and creating various pipelines. The main steps of this project consisted of pre-Training, Training, Validation and Evaluation. While the first three steps gave us satisfying results, during the Evaluation process we observed some limitations.

Limitations

The reconstruction of the MRI images, that the final Cross-modal model produced, was not accurate comparing to the original MRI images of the dataset. Moreover, the model seemed to be unable to generate diverse outputs for different signal inputs. The generated pseudo-MRI images looked in general quite similar to each other. Therefore, further investigation needs to be done in order to figure out whether we are experiencing mode collapse [8], [2]. When we talk about mode collapse, in the context of machine learning, we are referring to a situation where a trained model fails to capture and express the desired complexity or diversity of the data that it has been trained on. This term is widely used also in Generative Adversarial Networks (GANs) apart from the Autoencoders. Similarly, mode collapse refers to the situation where a GAN is able to generate only few modes(variations) of the input data distribution even though input data contains many modes. In contrast, the model converges to a simpler representation that does not fully take advantage of the variety of the training data. The model produces outputs that lack diversity and sometimes exhibit repetitive patterns or modes. Mode collapse can lead to a lack of variability in the latent space representation. Looking in more detail, the encoder network fails to capture the full complexity of the data distribution, leading to collapsed representations where similar inputs are mapped to nearby points in the latent space. As a result, the decoder network fails to generate diverse outputs.

Future Work

The most important factor that contributed significantly to the limitations regarding the regeneration of the pseudo-MRI, is the fact that in order to reduce the computational time we extremely reduced the sample sizes, not only for the ECG but also for the MRI. More specifically, from the initial 12-lead ECG of our dataset, we only utilised 1-lead for every ECG of our subjects, which contributed to the simplification of the Training data and resulted the lack of more diverse information. Moreover, regarding the MRI, we selected for every image only 1 frame from the initial 50 frame cardiac MRI. This decision had as a side effect the simplification of the MRI data as well, despite the fact that the computational time was drastically reduced. Hence, if we used the complete ECG and MRI information (12-lead ECG and 50 frame MRI) for our Training, instead of slicing the data, the results would be for sure more promising.

Another factor that possibly contributed to the aforementioned problems, consist the limitations regarding the expressivity of the model that we constructed. In order to overcome this problem we should use more complex network architectures. To be more specific, a more complicated model that we would suggest for future work would be a UNET model consisting of Residual and Attention Blocks [12]. We believe that such a model, would significantly contribute to the acquisition of more accurate results and we would be capable of overcoming mode collapse. In our case, the development of a second and improved model wasn't possible due to time restrictions.

Further research in the future could be made, focusing on the representations of the Latent Space. More specifically, our UK Biobank dataset contained information regarding some features of the subjects, such as sex, bmi, Acute MI, Ventr. Tachy, LV fail. Useful conclusions might be extracted if someone tries to group the points represented in the Latent Space based on some characteristics of the patients, once we have available various information. Patterns might be revealed for each modality (ECG or MRI) separately for different group of people and may help to detect and diagnose various cardiovascular diseases more effectively. In this direction, we separated the points in the Latent Space based on the patients sex (Male or Female) as one can see in Figure 6.1. Though, the points don't seem to be gathered in a specific space for Males or Females, without contributing to the extraction of a pattern at first glance.

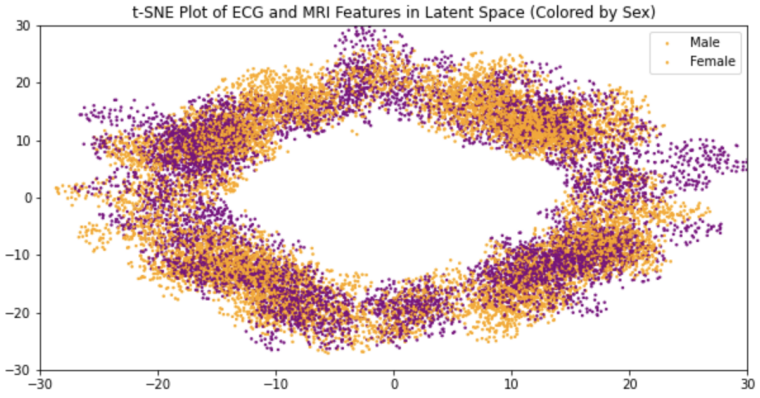
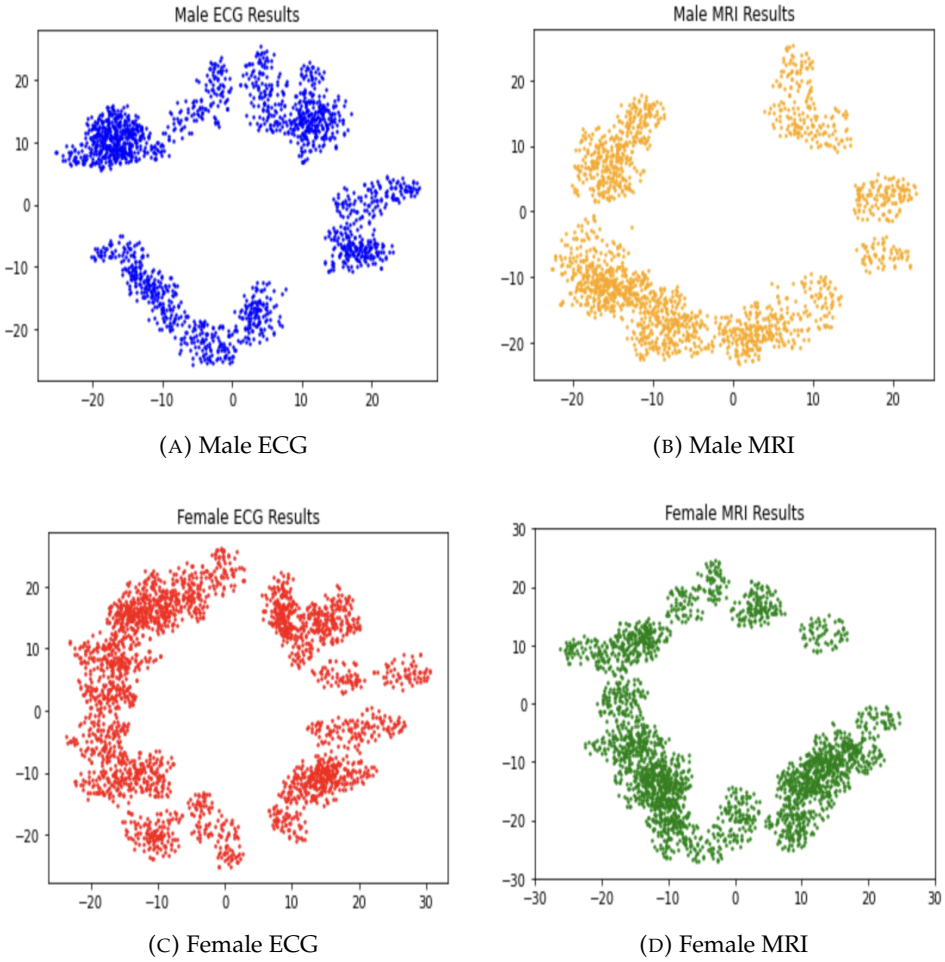


FIGURE 6.1: t-SNE Latent Space Representation grouped by sex

Taking a step further into this research, we grouped separately the points for each modality (ECG and MRI) and sex (Males and Females) shown in Figure 6.2. At this point, one can observe that in the latent space we have different groups like clusters. Hence, it would be anatomically interesting if these groups are expressed in the signals or images somehow.

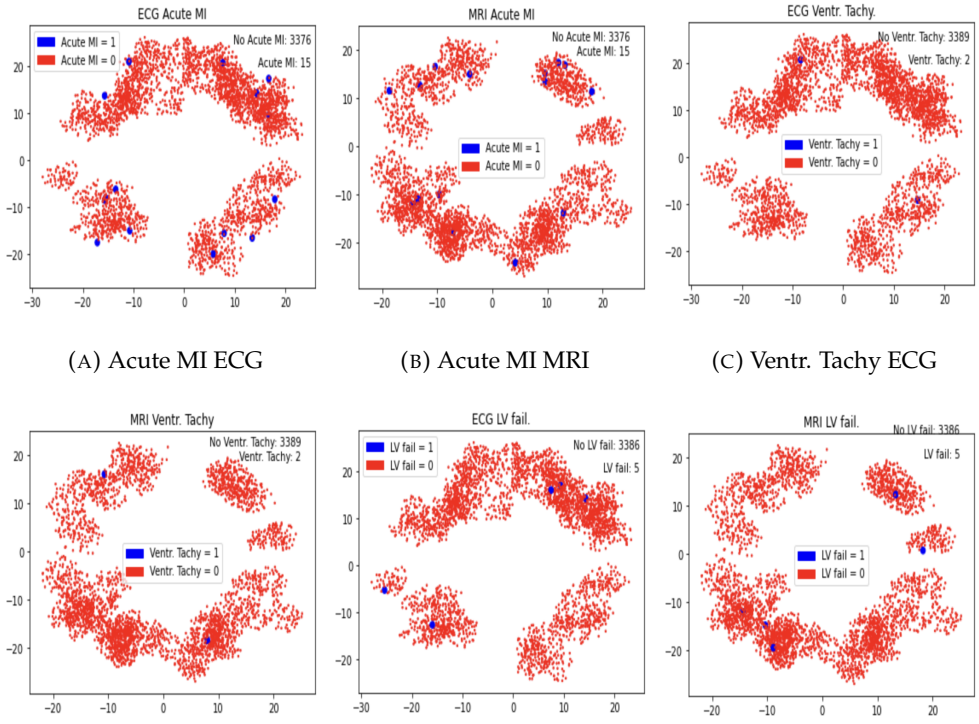


(A) Male ECG (B) Male MRI

(C) Female ECG (D) Female MRI

FIGURE 6.2: t-SNE Latent Space Representation for Female and Male Modalities

Last but not least, we tried to experiment a little bit with some diseases that some patients have suffered, like Acute Myocardial Infarction, Ventricular Tachycardia and Left Ventricular Failure. In Figure 6.3, with red colour are dislpayed the subjects that have not suffered from the disease in question and with blue colour are showed the patients who suffered. In this case as well, we are not able to extract useful conclusions, due to the fact that there is a small number of individuals in our training dataset, that have suffered from the aforementioned diseases. It is possible if someone examines a wider range of the Dataset, by utilising for Training a significantly greater number of patients data, to come to more generalizable conclusions. We have to underline the fact, that the available Dataset includes a great number of other features, that we have not included to our grouped Latent Space representations, therefore providing a solid ground for further research in this field.



(A) Acute MI ECG (B) Acute MI MRI (C) Ventr. Tachy ECG

(D) Ventr. Tachy MRI (E) LV fail. ECG (F) LV fail. MRI

FIGURE 6.3: t-SNE Latent Space Representation grouped by Acute MI, Ventr. Tachy and LV fail.

Appendix A

Master thesis source code

The source code for this master thesis can be found at the following github link: https://github.com/nickathans/master-thesis

Bibliography

[1] Pierre Baldi. "Autoencoders, Unsupervised Learning and Deep Architectures". In: *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*. UTLW'11. Washington, USA: JMLR.org, 2011, 37–50.

[2] Duhyeon Bang and Hyunjung Shim. *MGGAN: Solving Mode Collapse using Manifold Guided Training*. 2018. arXiv: 1804.04391 [cs.CV].

[3] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG].

[4] Minh-Son Dao. *Multimodal and Crossmodal AI for Smart Data Analysis*. 2022. arXiv: 2209.01308 [cs.AI].

[5] Nathaniel Diamant et al. "Patient contrastive learning: A performant, expressive, and practical approach to electrocardiogram modeling". In: *PLOS Computational Biology* 18.2 (2022). Ed. by Roger Dimitri Kouyos, e1009862. DOI: 10.1371/journal.pcbi.1009862. URL: https://doi.org/10.1371%2Fjournal.pcbi.1009862.

[6] Aarthipoornima Elangovan and T. Jeyaseelan. "Medical imaging modalities: A survey". In: *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*. 2016, pp. 1–4. DOI: 10.1109/ICETETS.2016.7603066.

[7] Jing Gao et al. "A Survey on Deep Learning for Multimodal Data Fusion". In: *Neural Computation* 32.5 (May 2020), pp. 829–864. ISSN: 0899-7667. DOI: 10.1162/neco_a_01273. eprint: https://direct.mit.edu/neco/article-pdf/32/5/829/1865303/neco_a_01273.pdf. URL: https://doi.org/10.1162/neco_a_01273.

[8] Benyamin Ghojogh et al. *Generative Adversarial Networks and Adversarial Autoencoders: Tutorial and Survey*. 2021. arXiv: 2111.13282 [cs.LG].

[9] José Guerreiro et al. "BITalino: A Multimodal Platform for Physiological Computing". In: vol. 1. July 2013.

[10] Mohamed Hammad, Yashu Liu, and Kuanquan Wang. "Multimodal Biometric Authentication Systems Using Convolution Neural Network Based on Different Level Fusion of ECG and Fingerprint". In: *IEEE Access* 7 (2019), pp. 26527–26542. DOI: 10.1109/ACCESS.2018.2886573.

[11] G. E. Hinton and R. R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (2006), pp. 504–507. DOI: 10.1126/science.1127647. eprint: https://www.science.org/doi/pdf/10.1126/science.1127647. URL: https://www.science.org/doi/abs/10.1126/science.1127647.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *arXiv preprint arxiv:2006.11239* (2020).

[13] ANAND JOSHI, ARUN TOMAR, and MANGESH TOMAR. "A Review Paper on Analysis of Electrocardiograph (ECG) Signal for the Detection of Arrhythmia Abnormalities". In: *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 03 (Oct. 2014), pp. 12466–12475. DOI: 10.15662/ijareeie.2014.0310028.

[14] Yann LeCun, Y. Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539.

[15] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[16] Laurens van der Maaten and Geoffrey Hinton. "Viualizing data using t-SNE". In: *Journal of Machine Learning Research* 9 (Nov. 2008), pp. 2579–2605.

[17] MathWorks. *Autoencoder*. https://es.mathworks.com/discovery/autoencoder.html. Accessed on 2023-06-16.

[18] Umberto Michelucci. *An Introduction to Autoencoders*. 2022. arXiv: 2201.03898 [cs.LG].

[19] Jiquan Ngiam et al. "Multimodal Deep Learning". In: Jan. 2011, pp. 689–696.

[20] Manoj Ojha, Sulochna Wadhwani, and Arun Wadhwani. "Automatic Detection of Arrhythmias From An ECG Signal Using An Auto-Encoder And SVM Classifier". In: (Oct. 2021). DOI: 10.21203/rs.3.rs-981164/v1.

[21] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *ArXiv* abs/1912.01703 (2019).

[22] Bhakti Patel and Amgad N. Makaryus. "Artificial Intelligence Advances in the World of Cardiovascular Imaging". In: *Healthcare* 10.1 (2022). ISSN: 2227-9032. DOI: 10.3390/healthcare10010154. URL: https://www.mdpi.com/2227-9032/10/1/154.

[23] Adityanarayanan Radhakrishnan et al. "Cross-modal autoencoder framework learns holistic representations of cardiovascular state". In: *Nature Communications* 14.1 (2023), p. 2436. DOI: 10.1038/s41467-023-38125-0. URL: https://doi.org/10.1038/s41467-023-38125-0.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[25] Soheila Saeedi et al. "MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques". In: *BMC Medical Informatics and Decision Making* 23 (Jan. 2023). DOI: 10.1186/s12911-023-02114-6.

[26] Cathie Sudlow et al. "UK Biobank: An Open Access Resource for Identifying the Causes of a Wide Range of Complex Diseases of Middle and Old Age". In: *PLOS Medicine* 12.3 (Mar. 2015), pp. 1–10. DOI: 10.1371/journal.pmed.1001779. URL: https://doi.org/10.1371/journal.pmed.1001779.

[27] Devin Taylor, Simeon E. Spasov, and Pietro Lio'. "Co-Attentive Cross-Modal Deep Learning for Medical Evidence Synthesis and Decision Making". In: *ArXiv* abs/1909.06442 (2019).

[28] Yonglong Tian, Dilip Krishnan, and Phillip Isola. *Contrastive Multiview Coding*. 2020. arXiv: 1906.05849 [cs.CV].

[29] Yifei Zhang. "A Better Autoencoder for Image: Convolutional Autoencoder". In: 2018.