Image Segmentation for Heart using Deep Learning

By

Valentin Craciun

Department of Computer Science
University of Sheffield

A report submitted in fulfillment of the requirements

for the degree of BSc in Computer Science.

Supervised by Dr. Haiping Lu

COM3610

May 2022

Word count: 9000

Abstract

Cardiovascular disease is an illness that affects many individuals worldwide and this thesis focuses the study on understanding how to use the current methods to create a system that is able to output accurate heart segmentation images for interest regions in a robust and easy way to use, which speed the current processes for analyzing patient's heart data.
In recent years, neural networks became more popular and surpassed traditional methods of machine learning, finding their way in every day applications from smart-phones to self-driving cars and also in medical domain. Deep learning networks are models that rely on neuron like structures to be trained systematically on labeled input and output data in order to make good predictions.
In order to approach this problem and be able to check the heart's health of a patient, I have used a data set [41] containing 20 patients with an average of 110 MRI images each (with a resolution of 320x320 px) and its left atrium segmentation for each one of them.
I have used 85% of the data (17 patients) for training a convolution neural network, under the U-Net architecture the rest being used for testing the model. Following that, for the output of the model I have used a series of image processing techniques in order to improve the model's performance, in the end, achieving a dice score accuracy of 87.97% after 9 minutes of training.
Furthermore, using the predictions made by the model, the system can output the volume of the left atrium for each patient in millimeter cube and milliliters (therefore being able to compare it with the expected volume of a healthy person and find if it is within normal parameters), a set of images comparing the ground truth and the true label, a video for each patient's heart that underlines the prediction against the true label on the input image and in the end, a point cloud interface that the user can interact with that represents the 3d shape of left atrium for each patient that the model was tested on.
The code for the system and its instructions can be found at: "`https://github.com/19valentin99/Left-Atrium-Segmentation-Dissertation`", the data set that I have used (split in training and testing) can be found at: "`https://www.dropbox.com/home/Dataset`" and a pre-trained model for the deep learning component can be found at: "`https://www.dropbox.com/home/Pre_trained_model`".

Declaration and Acknowledgments

Finishing of my dissertation for which I researched different methods to apply deep learning and image processing methods in order to obtain accurate heart segmentation images, I want to thank you to everyone who supported and guided me.
As this thesis couldn't have been possible without a scientific guidance, I want to thank you to Dr. Haiping Lu for the guidance that I received. Furthermore, I would like to thank you for offering me the chance to have this dissertation thesis.
Furthermore, I want to thank you to my love, Le Ma, my parents, Nicoleta and Antonel Craciun for all the unconditional encouragement and support that I received.

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Valentin Craciun

Signature: 

Date: May 1, 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

1.1.1 Cardiovascular Diseases Scale (CVDs)

Cardiovascular diseases represent the problems related with heart or blood vessels [1]. Through analyzing of statistical data, I found that in United kingdom the mortality rate caused by cardiovascular disease has decreased from 2000 (495 deaths in 100,000 people) to 2019 (255 deaths in 100,000 people). Because the numbers are still high, this illness is still on the top of mortal disease [2]. Furthermore, following the reports published by World Health Organization, about 17.9 million people died from CVDs in 2019 representing 32% of all global deaths [3].

1.1.2 CVDs Manifestation

By doing a thorough search over the medical literature, I have found that over 3 quarters of the deaths suffered from CVDs have been declared in the low and medium income countries. Most cardiovascular diseases are the outcome of risk factors such as smoking, unhealthy diet, physical inactivity and harmful use of alcohol [4]. Because most of the time, there are no prior symptoms of the disease, the first signs only appearing only when is already late (discomfort of the chest, heart attack or strokes), most cardiovascular disease could be prevented by addressing the risks factors mentioned above [4].

1.1.3 Current methods of analyzing the heart

Considering that it is easier to prevent a cardiovascular disease and lower the risk of it aggravating during an early stage, the methods of analyzing and diagnosing someone's analyses are very important [1]. Currently, the most used procedures for scanning the heart include echo-cardiogram, magnetic resonance imaging (MRI), cardiac computed tomography (Cardiac CT) and thallium scans[5].

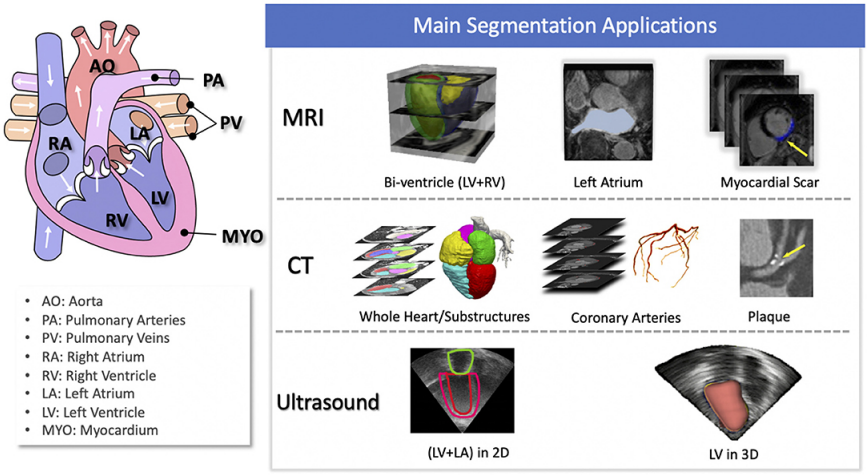

Figure 1.1: Segmentation Applications [17]

Echo-Cardiogram: represents non-invasive investigation which is very useful in finding the the size and thickness of the chambers, how the valves of the heart function, the direction of blood flow, any blood clots and any damaged areas or problems affecting pericardium [6]. The scans are made using ultrasounds after a special jelly was applied on the patient's chest and then a wand called "transducer" is placed on the outside of the chest, near the heart. The device sends sound waves through the chest and into the heart which bounce off and create images of the heart's structures on a screen [6] [7].

Thallium Scan is used to show how well the blood is reaching the heart muscles. This process is obtained by injecting a small amount of thallium and using a special camera, it can check the traces of the substance (the substance not being able to reach parts were there is a poor supply of blood) [5].

Cardiac CT and Magnetic Resonance(MRI) represents a non-invasive investigation which uses a special x-ray machine that takes detailed pictures of the heart and by analyzing them, the experts can find if it has any problems or not [5]. The process involves passing the patient through a large metal cylinder, which moves and is using magnetic fields in order to produce detailed images of the heart and blood vessels [5]. The particularities of a Cardiac CT scan are that they can provide faster pictures of tissues, organs and skeletal structure whereas the MRI is mostly used to determine if there are abnormal tissues within the body and offers more detailed images [7].

1.1.4 Interpret the data

By analyzing specialized data, I found out that the current common ways to interpret the data obtained from different types of scans are [8]:

1. Evaluating the anatomy and function of the heart chambers, heart valves, size of different components of the heart and blood flow through major vessels and the surrounding

structures such as the pericardium (the sac that surrounds the heart) [8];

2. Diagnosing a variety of cardiovascular (heart and/or blood vessel) disorders such as tumors, infections, and inflammatory conditions [8].

1.1.5 Current Problems

Researching some specialized articles, I found that one of the objectives from "High-level Meeting of the General Assembly on the Prevention and Control of Non-communicable Diseases" - the sixty-six world health assembly, is to reduce with 25% the overall mortality of cardiovascular and other diseases by 2025 [9] and in present the procedure to scan the heart is taking between 15 to 90 minutes [10] and the interpretation of the scans usually are taking up to 24 hours [11]. Because manual image segmentation requires precious time from medical experts, faster ways are needed in order to decrease the interpretation time and to keep the accuracy that a doctor could produce.

1.2 Overview of the Report

The thesis is divided into 6 chapters: "introduction", "background", "objectives analysis and legal requirements", "design", "experiments and results and conclusion".
Chapter 2, "Background" covers the literature review regarding the technologies that are used to do image segmentation, current programs that are able to perform image segmentation and the typical data format for this type of task.
Chapter 3, "Objective analysis and Legal Requirements" are presented the requirements for this project, an analysis of the data on which I've been training the model and the legal requirements.
textbfChapter 4, "Design", present the tools that have been used to accomplish this task, program architecture and particularities and the novel contributions that I have added implemented.
Chapter 5, "Experiments and results" show a couple of steps which were required in order for the project to reach the current stage and the results that were obtained after running the system using multiple configurations.
Chapter 6, "Conclusion" presents a summary of the thesis.

Chapter 2

Background

In order to find witch is the best way to approach this project and to obtain good results, I have conducted a literature survey. In the begging of this chapter, I presented what is meant by image segmentation and a few methods that could have been used to approach the task. In the following part I have included a short introduction to neural networks and the most important architectures that are currently used. In the end, I have included a few details of the existent system and an introduction to the data formats that are usually used for image segmentation tasks.

2.1 Segmentation - definition

Image segmentation is the process of dividing a digital image into multiple sub-areas (sets of pixels) in order to simplify and change the original image into something easier to analyze while keeping relevant information from the input image. In general, the image segmentation algorithms are used to locate boundaries and assign labels for every pixel in the image such that the pixels with the same label share the same characteristics [12].
The segmentation process can be approached using a variety of methods [13]:

1. Threshold segmentation is a simple and fast technique where the label of a pixel is determined by comparing one or more values;
2. Region segmentation methods, in which areas of an image are classified by grouping similar parts of the image in terms of colors or pixel values;
3. Edge (contour segmentation) in which the contours in an image are found using various methods and after, all the pixels bound by the contours are labeled;
4. Using a variety of neural networks architectures in order to perform image segmentation, the general idea being to train a multilayered perceptron and then to use a decision classifier in order to group the pixels. In a network, there are usually a high number of connections which can solve the problems even in a noisy environment and which

can learn complex behaviors from unstructured data. The main challenges while using this approach are: the amount of certainty for the quality of the labeled data, the understanding and prepossessing correctly of the data, selecting the appropriate network architecture(s) and having enough computational power and time to run the algorithms [13].

While all the methods from above are reliable and used for different computer vision tasks, in this project I have used a combination between first and last ones in order to have the best properties from both.
I chose to use different image processing techniques as they can be very quick and offer very good results in combinations with neural networks which were able to surpass traditional approaches during image segmentation competitions (ex: The Medical Image Computing and Computer Assisted Intervention Society (MICCAI)) [14] or WMH Segmentation Challenge [15].

2.2 Current methods of segmentation for CVDs

2.2.1 Neural Networks (NN)

The preliminary theoretical base for contemporary neural networks was independently proposed by Alexander Bain(1873) and William James (1890). In their work, both thoughts and body activity resulted from interactions among neurons within the brain[16].
Neural networks are a series of interconnected artificial neurons which uses a mathematical computational model (an input, a weighted sum with a bias, an activation function and an output) to simulate biological neurons[16].

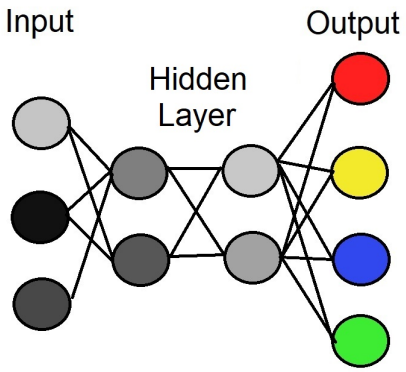

Figure 2.1: Neural Network Structure

The artificial networks are used for predictive modeling, the self-learning process resulting from allowing the network to train itself over a number of epochs, deriving conclusions from a complex set of information [16].
Neural networks are classified by multiple architectures which represent different ways to

connect and use the neurons inside the "Hidden Layer" in order to create a network with a specific set of qualities. Depending on the architecture, it might have a set of advantages, for a particular task in contrast with other neural networks (for example there are neural networks that are able to classify images better or others that work better with data processing) and at the moment there doesn't exist a neural network which excels in every task.

Through the analysis of some current studies, I observed some image segmentation papers which outlines state of the art algorithms, that achieved good results. Below I will enumerate the some of the highlighted models and their particularities [17].

2.2.2 Recurrent Neural Networks (RNN)

Recurrent neural network (RNN) is a type of neural networks, which uses sequential data or time series data. An RNN can remember the past and use the knowledge learned to make decision, the two most popular architectures being Long Short Term Memory (LTSM) and Gated Recurrent Unit (GRU). A use case for cardiac segmentation is to combine an RNN with a 2D FCN so that the combined network is capable of capturing information from adjacent slices to improve the inter-slice coherence segmentation results [17].

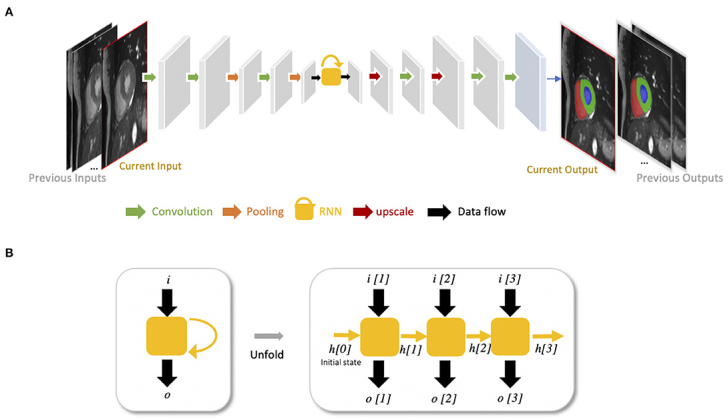

Figure 2.2: Recurrent Neural Networks architecture [17]

The advantages of using an RNN model are: can remember information through time (from previous inputs, each node of the model acting as memory cells). [20].
The downsides of using an RNN model are: vanishing and exploding of the gradients (model might not be able to learn depending on how deep it is) and it cannot process very long sequences [20].

2.2.3 Generative Adversarial Networks (GANS)

Generative Adversarial Networks (GANs) are a type of generative models that learn to model the data distribution of real data and are able to create new image examples. GANs consists of two networks: a generator network and a discriminator network. During training, the two

networks are trained to compete against each other: the generator is producing fake images aimed at fooling the discriminator, whereas the discriminator tries to identify the real images from the fake ones. As shown in the Figure 2.5, the generative model tries to create image segmentations as close as possible to the real ones [17].

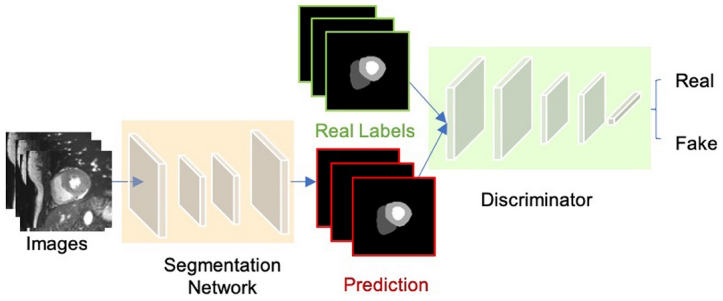

Figure 2.3: Generative Adversarial Networks [17]

The advantages of using GANs models for image segmentation are: can interpret diverse data, generates impressive result related to content, can be modeled for particular applications [25] [26].
The disadvantages of using GANs models for image segmentation are: hard to train (computationally expensive), can look "realistic" but are not real, do not work well for small scale structure. [25] [26].

2.2.4 Convolutional Netural Network (CNN)

Inspired by the connectivity patterns of the neurons in animal visual cortex [18], convolutional neural networks are the most common type of deep neural networks for image analysis. CNNs have multiple variants but all of them have four similar components:

1. convolutional layers(with the role to learn features' representation). This section is using a sliding kernel (usually the size being 3x3) and a step (stride) to compute the dor product between the kernel and the patch that is placed onto,
2. pooling layers (reduce the dimensionality of the network in order to be able to abstract the information). This operation helps by reducing the complexity of the calculations that the system has to do and also is helping with generalization.
3. fully-connected layers (combining all the resulted convolutional layers and try to estimate an output),
4. activation functions (which is used to weight the inputs between neurons' layers) [19].

CNN have been successfully applied to advance the state of the art on many image classification, object detection and segmentation tasks by levering their ability to reduce the dimensionality using a sliding windows and pooling layer [17].
Some notable CNNs models that proved to obtain high accuracy in computer vision challenges

are: LeNet, AlexNet, VGGNet and ResNet. Over the time, networks have been observed to become deeper in order to get better feature representations and therefore more precise results [19].

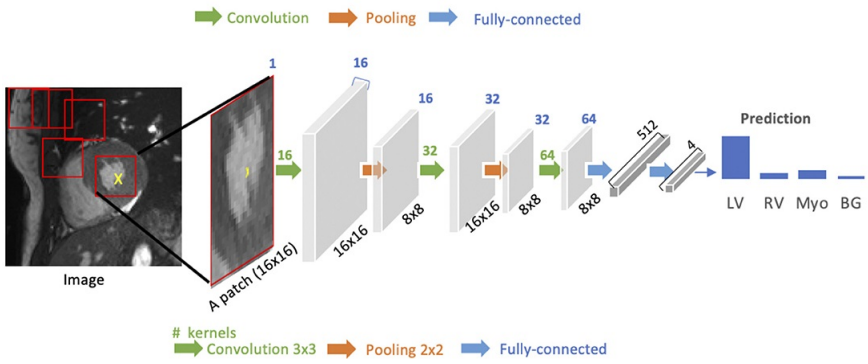

Figure 2.4: Convolutional neural netowrk - heart example [17]

The advantages of using a CNN are: very high accuracy for image recognition problems, automatic feature detection and weight sharing [20].
The disadvantages of using a CNN are: does not encode position or orientation and a lot of training data is required to train the networks[20].

2.2.5 Fully Convolutional Neural Networks (FCN)

Fully convolutional neural networks are a special type of CNNs that do not have any "Dense" layers (as CNNs) [21], instead it has convolutions that perform the task of fully connected layers (Dense Layers). They are designed to have an encoder (reducing the dimensional of the image) and after to reconstruct the image using a decoder. The most well-known variant of FCNs for bio-medical image segmentation is the U-net architecture variant which skips connections between the encoder and decoder to recover spatial context loss in the down-sampling path, yielding more precise segmentation [17]. One important advantage that U-Net has is the large number of up-sampling layers, which increases the resolution, allowing the network to propagate contextual information to the later layers. [22] The U-Net architecture is formed by 3 main parts:

1. Encoder (or contracting path) which is represented on the left most part of the figure and it follows a traditional CNN architecture. It is formed from applying 4 times a double convolution layer, after each of them having a max polling layer in order to reduce the number of dimensions of the image;

2. The middle part from the bottom of the network (usually called bottle neck) has 2 convolutions and no max polling layers;

3. Decoder (or the expansion part)which is displayed on the right most part of the figure and it integrates two components: the skipped connection from the encoding part and

the transposed convolutions (which expands the size of the image);

4. In the end, the last part of the network is to reshape the matrix and to have it the same shape as in the begging. Afterwards, the result is fed into a fully connected neural network which will segment the image into the number of classes that are necessary (in the case of left atrium segmentation, there will be 2 classes: background and left atrium).

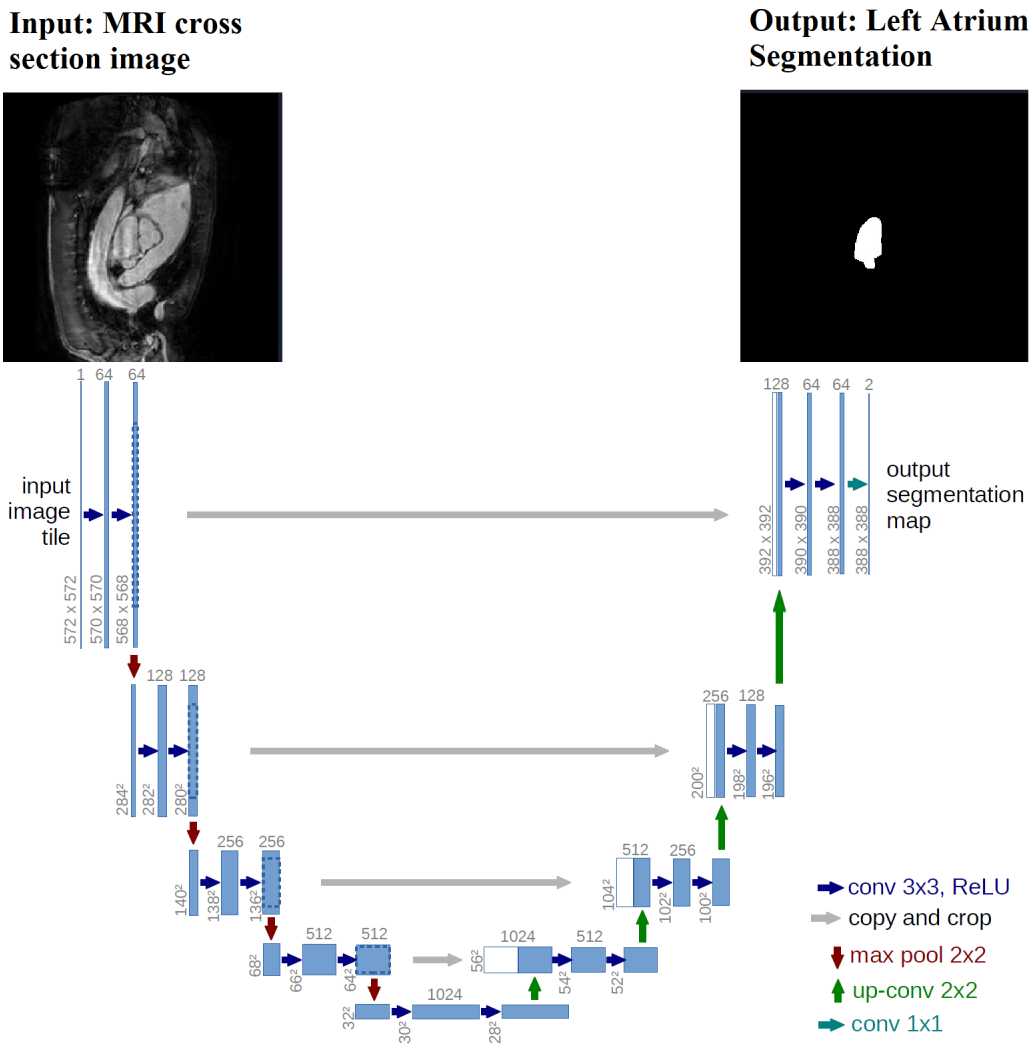

Figure 2.5: FCN UNet Architecture [24]

There are multiple variations of U-net model: 3D U-Net (which is used to learn to perform dense volumetric segmentation), pixel-wise regression (in order to combine high resolution gray scale images with small resolution, colored images to output high resolution color images)[22] [23].
The advantages of using U-Net architecture are: the architecture can accommodate large

images, works well with small training sets because of the robustness offered by data augmentation [24].
The disadvantages of U-Net architecture are: it needs contextual data in order to work properly, a lot of GPU memory is required for larger images (ex: between 4 and 8 640x959 images per batch can require 6GB GPU; for 1 or 2 1280*1918 images per batch can require 12 GB of GPU) [24].

2.3 Existent Code

2.3.1 nnUNet

As a result of image segmentation being a popular field in computer vision and medical areas, there are multiple open source codes available online. One algorithms that I looked into and tried to understand was nnUNet [27]. It is a bio-medical image segmentation algorithm, which condenses and automates the keys decisions for designing a successful segmentation pipeline for any given dataset. After running the nnUNet on one of the examples, I found out that it ran for a total of 9 hours, the training accuracy for the particular task being around 90%.
The advantages of using this model are:

1. based on multiple image segmentation tasks on which the model was used it proved to be robust and to performed well,
2. it usually sets the threshold for developing new image segmentation methods,
3. it is "ready out of box", if the user has the data partitioned as git repository's user guide mentions.

The downsides of this library are:

1. even after it reached a desirable accuracy of 90% after 30 epochs (around 15 minutes of run-time) the program continued to run for 8 more hours trying to do small adjustments and just bouncing around,
2. another downside is that the checking of any additional segmentation task will take a long time as well,
3. it requires a powerful graphics card (at least Nvidia Graphics card 2080),
4. it needs an SSD hard drive with a few hundred of GBs of available space.

2.3.2 Other Image Segmentation techniques

A few other methods and their results for image segmentation can be extracted from "Deep Learning for Robust Segmentation" paper [50].The first method presents 2 interconnected neural network layers, initialization and spatial propagation which were trained for 23 and 44 hours using GPU acceleration. Using this method, the author was able to achieve a dice

score accuracy of 0.78 [50].
The second method proposed in this paper, presents a combination of interest region method (either by using the center of the image or by training a neural network to detect it) with a convolution neural network approach using the U-Net architecture. Furthermore, the predictions are enhanced by using a series of post processing techniques (detecting the locations of the segmentation, extreme pixel value thresholding using 5th and 95th percentiles and normalization). By using this method, the average the dice score obtained was 0.91 [50].

2.4 Data Formats

In medical domain, image analysis is different from other domains because of its characteristics. The images usually store more information, ranging from 2D to 5D. In general, the heart analysis (MRI, CT) is storing volumetric data and thus the images are taken longitudinally (over time). Therefore, in order to accommodate these characteristic, specialized data formats have been implemented, for example NIfTI and DICOM [32].
Neuroimaging Informatics Technology Initiative (NIfTI) is an open file format commonly used to store brain imaging data obtained using Magnetic Resonance Imaging methods (MRI) [33]. Digital Imaging and Communications in Medicine (DICOM) file format is the standard for the communication and management of medical imaging information and related data and therefore some of the data-sets that are available might need to be converted to NIfTI format for ease of use [34].

Chapter 3

Objectives analysis and Legal requirements

3.1 Project Requirements

For this project, the goal is to design and implement a left atrium segmentation system for which the tasks are to:

1. use deep learning to predict accurately, given a set of MRI images, the location of the left atrium similar with what an expert or a state of the art system would produce;
2. implement a robust program for which if new data is presented, to be able to make good predictions;
3. train and do predictions faster than the current state of art systems;
4. output data to include: heart volume, 3D point cloud and visual representation of the predictions;
5. modular (a new user should be able to extend the code or modify some parts of it without being too difficult);

In order to evaluate the performance of the project I have used the following methods:

1. Dice score formula, which measures how similar two images are, and outputs a number between 0 and 1 (0 meaning low similarity and 1 meaning that the images are the same),

$$\text{Dice score} = \frac{2 \cdot \text{number of true positives}}{2 \cdot \text{number of true positives} + \text{number of false positives} + \text{number of false negatives}}$$

Figure 3.1: Gumeo - Dice Score formula [35]

2. Use of different graphical representations such as overlaying the predicted segmentation and the actual segmentation contours over the heart images and displaying a 3D point cloud of the actual segmentation

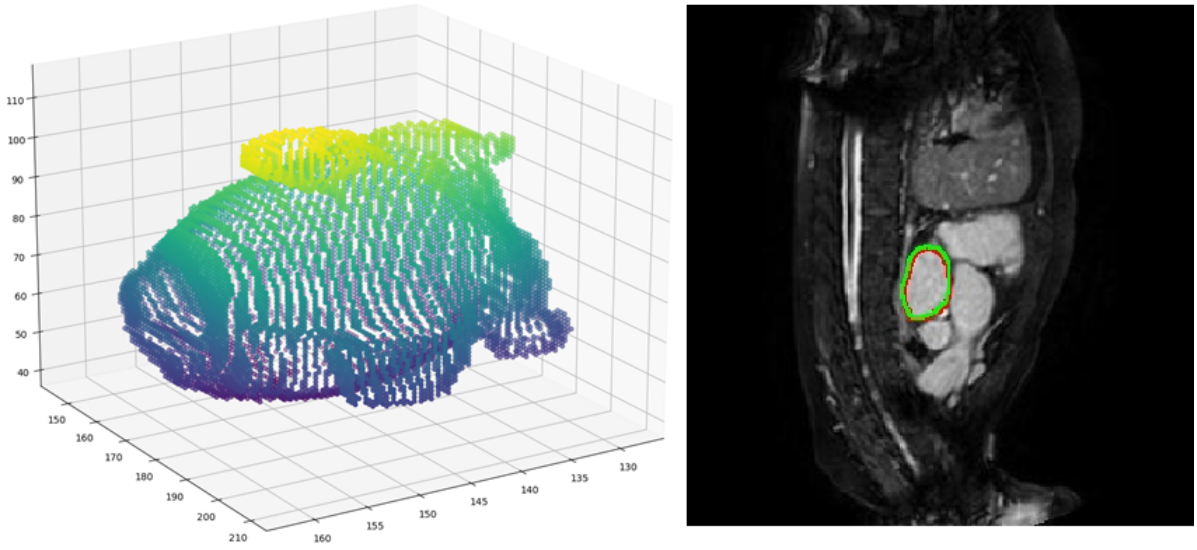

Figure 3.2: Interactive 3D Point Cloud representation and Contour overlay of the predicted output (in green) vs true output (in red) of the Left Atrium

3. Time analysis of the system's running time performance for different hyper-parameters to complete training, testing and process all the outputs in rapport with the hardware that I used.

3.2 Analysis

3.2.1 Data set used

The data set that I have used for this project, is one I found on Kaggle website from the Left Atrial Segmentation Challenge MICCAI 2013 [41]. This data-set consist of 20 patients who had a hearth MRI, with an average of 110 picture slices. The data is encoded as .nii format but it can be visualized and interpreted as normal .jpg images. Furthermore, the dataset comes with a set of binary images encoded as .nii (black - background and white - left atrium section) that represent that segmentation of the left atrium.
In order to calculate the volume of the left atrium based on the segmentation, I needed the distance between the voxels for the .nii images for witch I assumed it is 1mm x 1mm x 1mm as stated in the resources that I found [42].

To develop the system I have used just this data-set for multiple reasons: the difficulty of finding more recent data-sets alongside with the fact that I wanted to prioritize the performance and the quality due to time constraints.

Figure 3.3: Cross section image of the heart and the segmentation of the left atrium in the particular frame

3.2.2 Limiting factors

1. **Time constraints**: The completion of this project had to be done alongside other university projects witch have the deadline in the same period. Furthermore, this project requires a large amount of knowledge to be acquired (PyTorch, image processing, medical imaging data formats, literature analysis and understanding what has been already been done). Furthermore, training and testing the models, solving any problems that arise require additional time to be spent.

2. **Hardware constraints**: due to the computational complexity of the task, without relying on very powerful computers, depending how large the training set and for how many epochs the models are trained it can take a long time. Furthermore, depending on other parameters, the training might not even be possible: for example using mini-batches during the training, the system might crash due to it running out of virtual gpu memory (when trying to use more than 10 samples per batch).

3. **Accuracy of the system**: the limitations mentioned above, and the the fact that the amount of data that I have been working was very small for this type of task (20 patients with an average of 110 samples each) play an important factor in the overall result of the system.

3.3 Ethical, Professional and Legal Issues

For this project, the data-set that I have used for developing the system are publicly available, does not disclose personal details of the patients who had the MRI taken and therefore they do not require legal approval. The data-set that I used is from Left Atrial Segmentation Challenge MICCAI 2013 [37].

Chapter 4

Design

4.1 Design Introduction

The main tools that I have used to create an industry standard software that uses deep learning to do image segmentation are: Python, PyTorch (a popular deep learning library developed by Facebook), OpenCv (which stands for Open Computer Vision and it is usually used in tasks that require image processing) and SimpleITK (in order to work with the medical image formats .nii). The reason that I chose to use PyTorch for deep-learning is that it has a pythonic syntax and easier learning curve, in my opinion, compared with its competitor: TensorFlow.
Furthermore, in this project I have decided to use a fully convolutional neural network with the U-Net architecture because it seemed to be more suited for this task and for the amount of data that I had available. In addition, the other architectures ,which could could be used to do image segmentation, had more sever downside: for example RNNs have vanishing or exploding gradients, CNNs which usually require a lot of data. Therefore, from the literature review that I conducted and software de-risking phase I found out that the only limitation for this model would be the number of images I can include in a mini-batch.

4.2 System Overview

4.2.1 Architecture - Program Flow

The main modules of the project are:

1. Generator of the folder structure (where the user can place the training and testing data using .nii format);
2. An image format converter from .nii to .jpg, generating a set of 2D images that represent the cross-sections of the 3D image. The main reason of adopting this method is that after conducting a few experiments and trying to train deep learning models using

mini-batches on the .nii images, there were errors coming up,the reason being that the number of dimensions of .nii files were different from a patient to another.

3. The main part of the project, which combines the remaining sub-modules:
   (a) loads the training and testing data and apply different transforms over it;
   (b) trains a deep learning model using the U-Net architecture (or load a pre-trained one)
   (c) does a set of preliminary prediction using the test data;
   (d) upgrades the image predictions using a variety of image processing techniques;
   (e) calculates the Dice Score of the prediction and display it in the terminal
   (f) computes the left atrium volume using the predicted and true segmentation for each patient and displays it in the terminal;
   (g) saves for each tested image an overlay of the prediction's contour and the true segmentation over the input image;
   (h) for each tested patient saves a video which represents the overlay images one after another;
   (i) for each tested patient displays an interactive window which represent a point cloud structure of the 3D segmentation of the left atrium.

4.2.2 Architecture - Model Details

Program's parameters and hyper-parameters

For each component of the program, I have made it such that all the variables that need to be set, are in the begging of each file and therefore it facilitates the ease of use of the system (for example enhancing the data after testing, display or save the results of the testing).
Furthermore, the epoch hyper-parameter of the model is set to be a low number(in rapport with the complexity of this task) in order for the model to learn just the main feature of the segmentation, helping e model outputting images that represent the confidence level for each pixel for the segmentation instead of a true segmentation which highlights only the interest area.

Data Transforms

After loading the data, before training and testing, I am applying transform for the images in order to have the mean 0, standard deviation 1 and the maximum pixel value is set to 255.

U-Net model particularities

The model is formed by a series of 9 doubled convolutions (4 representing the down connections, 1 for the bottleneck and 4 for the upwards connections). For each convolution I used a kernel with the size 3 by 3, a stride of 1 and padding of 1.

$$[\{(n + 2p - f + 1) / s\} + 1] * [\{(n + 2p - f + 1) / s\} + 1]$$

Figure 4.1: Output image shape after applying a convolution. Parameters: Input Image Size (n x n); kernel size (f x f), padding (p), stride (s) [51]

After each convolution, I have applied a rectified linear activation function (ReLU) that will output the input directly if it is positive, otherwise, it will output zero. This function has became one of the most used activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance [52].

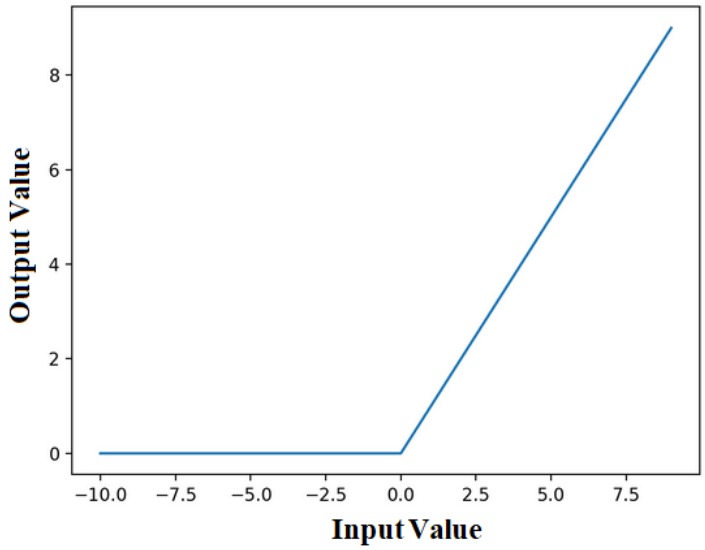

Figure 4.2: ReLU example plot [52]

After each set of double convolutions, in the descending path of the U-net architecture, I have added a max pooling layer with the size 2x2.

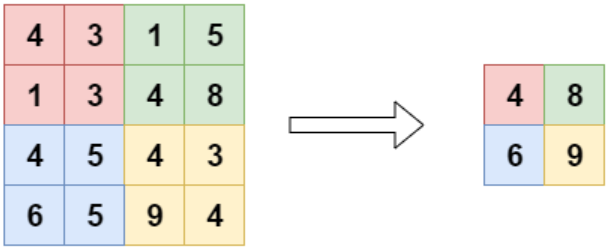

Figure 4.3: Max pooling layer illustration [51]

Furthermore, when initializing the model I set the number of "in" and "out" channels is set to 1 because both the input with the cross section of the heart and output image that contains the segmentation are grey-scaled images (1 channel images).

Loss Function

Binary classification is a problem where we have to segregate our observations in any of the two labels on the basis of the features [46]. In order to train a model to predict a binary classification we can use binary cross entropy (BCE) loss which compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value. BCE is the negative average of the log of corrected predicted probabilities [46].

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = -w_n \left[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n) \right]$$

Figure 4.4: Binary cross entropy loss formula[53]

The loss function that I have used for training the model is "BCEWithLogitsLoss" which is a combination between binary cross entropy (BCE) loss and sigmoid. This version is more numerically stable than using a plain sigmoid followed by a BCELoss. By combining the operations into one layer, this BCE with logits takes advantage of the log-sum-exp trick for numerical stability. In this way, the model can measure the reconstruction error for example for an auto-encoder[45].

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = -w_n \left[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n)) \right]$$

Figure 4.5: Binary cross entropy with logits loss formula[54]

Optimizer

The optimizer that I have used is Adam which is an algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights. It is an extension to stochastic gradient descent which instead of adapting the parameter learning rates based on the average first moment (the mean) as in RMSProp, Adam also makes use of the average of the second moments of the gradients (the uncentered variance), this helping to overcome biased estimations.[47].

Prediction Enhancements

In order for the program's output to be better, and to increase in some cases the accuracy of the prediction by over 50% I used a series of post-processing techniques.
The series of enhancements that I have implemented are:

1. An edge case is that when the output should be blank (no segmentation is discovered) the neural network outputs a whitish image of the input data. After multiple experiments, I found out that the average value in these type of images is much higher than the average when segmentation is actually happening therefore all the images that have a higher average than the normal segmentation are set to be 0 (no segmentation).

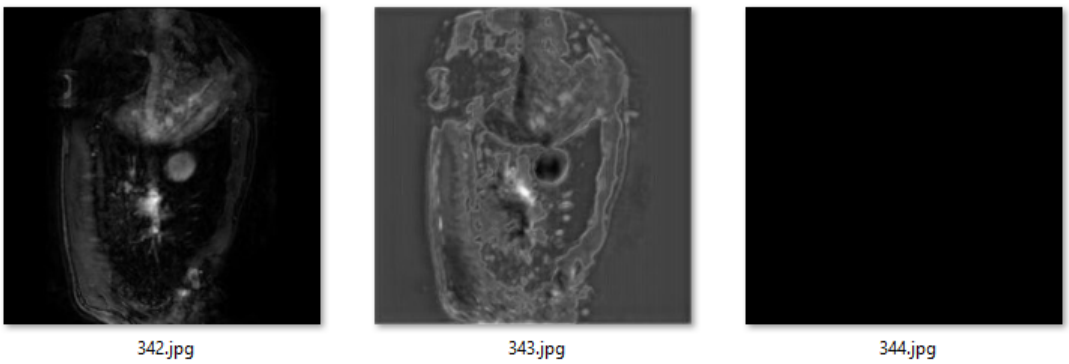

Figure 4.6: Edge case representation: first image is the input image, second the prediction and third the true segmentation (it is blank because the left atrium cross section didn't started yet)

2. Applying a threshold over the output images based on the confidence values of the segmentation setting it to foreground (255 - part of the left atrium) or background (0 - everything else).

3. Applying a morphological closing transform from the OpenCV library in order to cover any patches within the segmentation that might have had a low confidence estimate and have been considered to be background within the segmentation part. This process is using a kernel and two basic operations: dilatation and erosion [43].

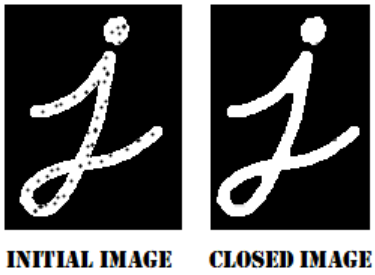

Figure 4.7: Example of closing operation [43]

During the erosion process the kernel slides through the image (as 2D convolution). A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is eroded (made to zero).So what happens is that, all the pixels near boundary will be discarded depending upon the size of kernel. So the thickness or size of the foreground object decreases or simply white region decreases in the image[43].

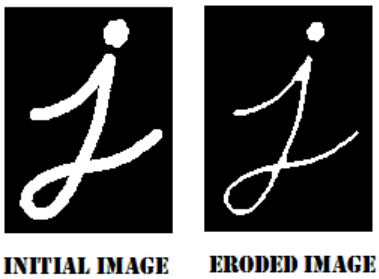

Figure 4.8: Example of dilatation operation [43]

The dilation process is just opposite of erosion. Here, a pixel element is '1' if atleast one pixel under the kernel is '1'. So it increases the white region in the image or size of foreground object increases [43].

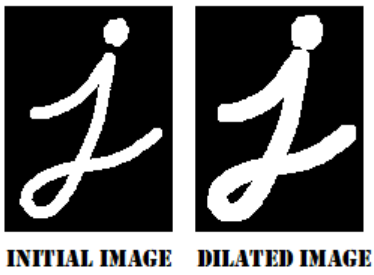

Figure 4.9: Example of closing operation [43]

4.2.3 Novel Contributions

1. **Speed and Accuracy**: by allowing the model to train only for a short period of time (a few epochs), it is able to capture enough information so that the predictions can be used afterwords in a post processing step which improve the quality of the prediction significantly by using image processing techniques. In this way, the accuracy is able to compete with state of art models while the training time is much faster (10 minutes for the training with a resukt of 87.97% accuracy for the system that I have implemented compared with nnUNet that achieves 89% accuracy and takes 8 hours for the training).
2. **A combination of output visualization**: using the segmentation images outputted by the system for the left atrium, the program can output multiple information about a patient: overlays comparing the prediction with the ground truth, a video for each patient which compares the output and ground truth, information about the volume of the left atrium (in centimeters cube and milliliters) and 3D point cloud visualization .

Chapter 5

Experiments and results

5.0.1 Preliminary Experiments

Using nnUNet

After setting up and start running nnUNet [27] for a segmentation task it obtained a dice score of approximately 89% after 8 hours (the main reason it took so long is that the model trained for 1000 epochs). Therefore, my goal was to try to obtain a similar accuracy but in a shorter period of time. Furthermore, nnUNet code was complex and not very easy to understand and one of my goals was to make a system which is much easier to use for a new user.

Inspecting the data

After analyzing the data that I was working with, I discovered that the heart's shape and its features can vary quite a lot from patient to patient.

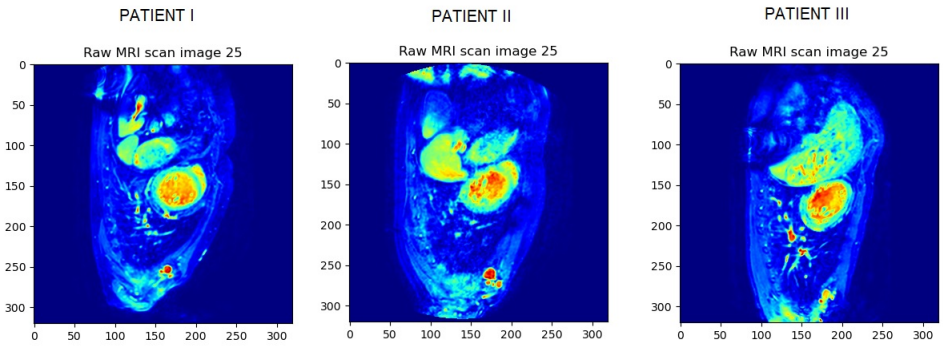

Figure 5.1: Different patients - different hearts

Only Applying Thresholds

In order to make a few experiments and to analyze the data-set, I created a GUI which offered the possibility to view a comparison between the true segmentation and a segmentation made by just applying a threshold. After this experiment (as it can be seen in Figure 5.2, in the last picture from the right), by only applying a threshold it is not enough to create a good segmentation of the left atrium as it doesn't look that similar with the third one which represent the segmentation done by an expert.

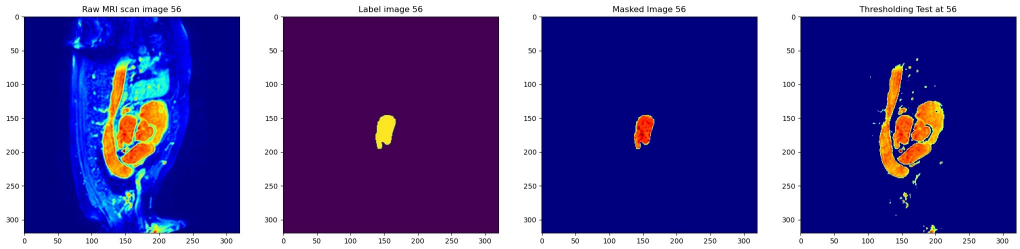

Figure 5.2: GUI created in order to analyse the heart data and threshold method. The first image represent the input image, the second one the true segmentation, the third one represent the segmentation mask applied over the input image and the forth one how the input image would look if only a threshold would have been applied

Only Applying Deep Learning

Therefore, I have implemented a model using the U-Net architecture which I trained using 85% of the data. The testing of the model was done on the remaining 15% (3 patients). The segmentation looked promising as it can be seen in Figure 5.3, but not good enough as there are intermediary values (between 0 and 255).

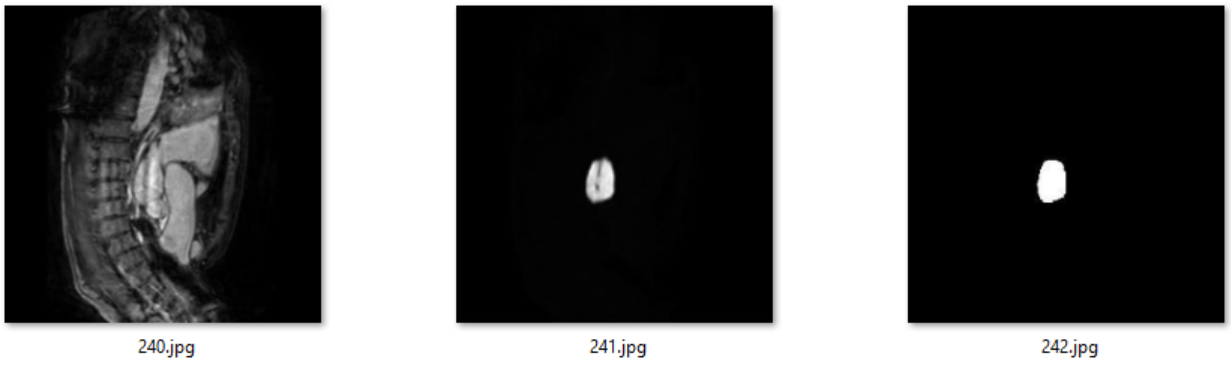

Figure 5.3: Applying only deep learning: First Image - input; Second Image - prediction; Third image - ground truth [Dice Score: 0.2413]

Deep Learning and Threshold

Therefore, in order to fix this issue, my first improvement that I made was to apply a threshold over the output in order to keep only the confident pixels (boosting the confident pixels to 255, and lowering the non-confident ones to 0). This improvement boosted the accuracy of the system (dice score) by approximately 30%.

Figure 5.4: First image is the input, second one the prediction and third is the ground truth [Dice Score: 0.4821]

Deep Learning, Threshold, Morphological transforms and edge case

After the previous improvement, I have decided to add the morphological transforms in order to close possible small gaps in the segmentation (if existent) because after analyzing the left atrium I found out that most of the times it should be a full shape.
Furthermore, I have considered the edge cases where the output should be ignored (as it is a whitish image) meaning that the cross-section of the left atrium segmentation hasn't started. After applying these 2 improvements and running the system multiple times, the dice score average was 0.85.

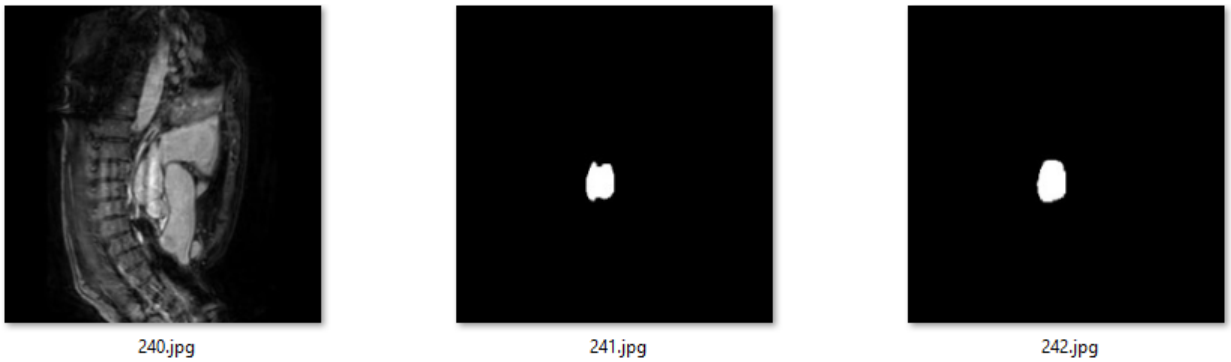

Figure 5.5: First Image is the input, second the prediction and third is the ground truth [Dice Score: 0.8503]

Best model trained so far

Because deep learning is a stochastic process and the accuracy can vary between runs I have added bellow the segmentation result using the model for which I obtained the highest accuracy.

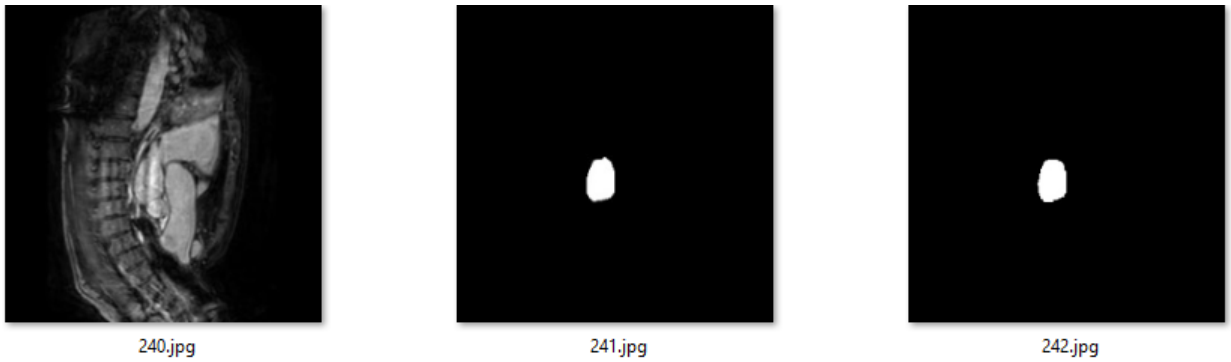

Figure 5.6: First Image is the input, second the prediction and third is the ground truth [Dice Score: 0.8797]

5.0.2 Results

Accuracy and time

In the begging, I have tried to add during the data augmentation process a small rotation to the heart images in order to make the model more invariant to change. Furthermore, these results are before applying morphological transforms. As it can be viewed from the next table, the model was not always able to perform at its full potential.

Learning Rate	Batch Size	Number of Epochs	Running Time (seconds)	Dice Score
1e-4	5	3	345	0.836
1e-4	5	3	347	0.794
1e-4	5	3	350	0.582
1e-4	5	3	366	0.72
1e-4	5	5	563	0.813
1e-4	5	5	564	0.608
1e-4	5	5	563	0.534
1e-4	5	10	3113	0.709
1e-4	5	10	2029	0.534

Table 5.1: Table presenting the results with a small rotation applied to the data and without morphological transforms

The average dice score in this setup was 0.68 with the highest dice score being 0.836 after a

training time of 345 seconds.
Therefore, after finding out that the model is not able to learn always the correct segmentation for the left atrium (using a small number of epochs), I decided to remove the rotation in the augmentation as I thought that the MRI data should be taken in a specific way and therefore the heart will always be positioned in the same position. Furthermore, I have implemented the image morphological transforms in order to fill small gaps in the segmentation. After I made these small changes, I found out that the model, regardless of the number of epochs was trained, will perform well, obtaining a dice score over 0.83 (this method includes deep learning and applying the post processing techniques that were discussed in the previous chapter). The results following this were:

Learning Rate	Batch Size	Number of Epochs	Running Time (seconds)	Dice Score
1e-4	3	3	255	0.87
1e-4	5	3	634	0.851
1e-4	5	3	533	0.853
1e-4	5	3	547	0.846
1e-4	5	3	547	0.861
1e-4	5	3	551	0.84
1e-4	5	3	550	0.865
1e-4	5	3	550	0.8797
1e-4	5	5	865	0.851
1e-4	5	5	860	0.834
1e-4	5	10	1768	0.846
1e-4	5	20	3379	0.79

Table 5.2: Table presenting improved results, after removing the rotation and adding a making the image post-processing more robust

The average dice score after the update was was 0.849 with the highest dice score being 0.8797 after 550 seconds. Therefore, after making the previous changes, the average dice score has increased the accuracy by 17% resulting in more consistent accurate predictions.

Program Outputs

Overall, the highest accuracy that I was able to achieve is 87.97% which is a big improvement compared with the accuracy obtained in the beginning (16%). In addition, this boost in accuracy helps with making more realistic predictions of the left atrium's volume, being only up to 3 milliliters off of the actual volume.
Normal volume of the left atrium is less than 28 mL/m2 or inscreased between 29 and 33 mL/m2, moderate is between 34 and 39 mL/m2 and severe is when it is more than 40 mL/m2 [48] [49].

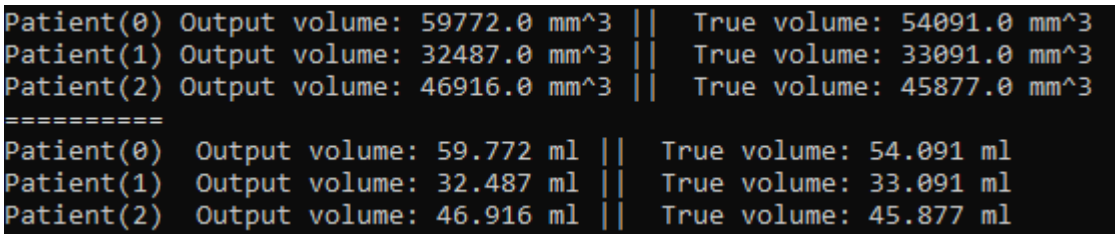

Figure 5.7: Output volume after after applying the heart segmentation

Chapter 6

Conclusions

In conclusion, the study regarding image segmentation using the U-net architecture was successful. Furthermore, the novel contributions of enhancing the prediction using different image processing techniques (in order to save time for training) and to have different way of displaying the outputs (3d representation, overlay comparison, videos that represent the comparison between the model's prediction and ground truth over time) worked well.
In the end, despite the limitations regarding the short time for this project to be delivered, not having enough data (only 20 patients; problem which can be addressed in the future by using multiple patients) I was able to achieve all of the project requirements discussed in chapter 3.1 such that a new user is able to train a model and test it on left atrium MRI data fast, to obtain accurate results and to obtain a high range of interpretations for the segmentation output.

Appendices

Bibliography

[1] Cardiovascular disease - Wikipedia (https://en.wikipedia.org/wiki/Cardiovascular_disease) [Accessed on: 15 November 2021]

[2] statista - mortality rates in UK (https://www.statista.com/statistics/940678/cardiovascular-disease-mortality-rate-in-the-united-kingdom-uk/) [Accessed on: 6 October 2021]

[3] World Health Organisation - Cardiovascular Disease (https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) [Accessed on: 1 November 2021]

[4] World Health Organisation (https://www.who.int/health-topics/cardiovascular-diseases) [Accessed on: 1 November 2021]

[5] Heart Scans (https://www.chss.org.uk/heart-information-and-support/about-your-heart-condition/tests-and-investigations/heart-scans/) [Accessed on: 1 November 2021]

[6] MedicalNewsToday - Echo Cardiogram (https://www.medicalnewstoday.com/articles/326727#about) [Accessed on: 14 October 2021]

[7] healthline: CT Scans vs MRI (https://www.healthline.com/health/ct-scan-vs-mri) [Accessed on: 14 October 2021]

[8] Magnetic Resonance Imaging - Cardiac (https://www.radiologyinfo.org/en/info/cardiacmr) [Accessed on: 14 October 2021]

[9] High-level Meeting of the General Assembly on the Prevention and Control of Non-communicable Diseases (https://apps.who.int/gb/ebwha/pdf_files/WHA66/A66_R10-en.pdf?ua=1) [Accessed on: 5 November 2021]

[10] NHS (https://www.nhs.uk/conditions/mri-scan/what-happens/) [Accessed on: 5 November 2021]

[11] MRI RESULT Time (https://www.independentimaging.com/when-can-i-expect-my-results-after-an-imaging-study/) [Accessed on: 27 November 2021]

[12] Image Segmentation Wikipedia (https://en.wikipedia.org/wiki/Image_segmentation) [Accessed on: 9 October 2021]

[13] Baowang Li - Segmentation of Images - University of Sheffield (https://www.dcs.shef.ac.uk/intranet/archive/campus/2018_2019/projects/msc/index.html) [Accessed on: 9 October 2021]

[14] MICCAI 2020 Challenge (https://abcs.mgh.harvard.edu/index.php/leader-board) [Accessed on: 20 November 2021]

[15] WHM Segmentation Challenge (https://wmh.isi.uu.nl/results/) [Accessed on: 20 November 2021]

[16] Neural Networks(https://en.wikipedia.org/wiki/Neural_network) [Accessed on: 28 November 2021]

[17] Chen Chen, Chen Qin, Huaqi Qiu, et al. - Deep Learning for Cardiac Image Segmentation (https://www.frontiersin.org/articles/10.3389/fcvm.2020.00025/full) [Accessed on: 28 November 2021]

[18] Convolutional Neural Network - wikipedia (https://en.wikipedia.org/wiki/Convolutional_neural_network) [Accessed on: 29 November 2021]

[19] Jiuxiang Gua, Zhenhua Wangb, Jason Kuenb, Lianyang Mab, Amir Shahroudyb, Bing Shuaib, TingLiub, Xingxing Wangb, Li Wangb, Gang Wangb, Jianfei Caic, Tsuhan Chenc - Recent Advances in Convolutional Neural Networks (https://arxiv.org/pdf/1512.07108.pdf%C3%A3%E2%82%AC%E2%80%9A) [Accessed on: 29 November 2021]

[20] GeeksforGeeks - Difference between ANN, CNN and RNN (https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/) [Accessed on: 30 November 2021]

[21] himanshu Rawlani - Understanding and implementing a fully convolutional network (FCN) (https://towardsdatascience.com/implementing-a-fully-convolutional-network-fcn-in-tensorflow-2-3c46fb61de3b) [Accessed on: 1 December 2021]

[22] U-NET - wikipedia (https://en.wikipedia.org/wiki/U-Net) [Accessed on: 1 December 2021]

[23] Pansharpened images - wikipedia (https://en.wikipedia.org/wiki/Pansharpened_image) [Accessed on: 1 December 2021]

[24] BrightlyDark - Image Segmentation using U-NET (https://ashm8206.github.io/2018/03/07/Ultrasound-Image-Segementation-Using-Unet.html) [Accessed on: 1 December 2021]

[25] Junaid Rehman - Advantages and disadvantages of generative adversarial networks (https://www.itrelease.com/2020/06/advantages-and-disadvantages-of-generative-adversarial-networks-gan/) [Accessed on: 4 December 2021]

[26] Thuong Nguyen Canh - Research Gate (https://www.researchgate.net/post/Can-anyone-inform-me-please-about-the-advantages-and-limitations-of-Generative-Adversarial-Networks-GANs) [Accessed on: 4 December 2021]

[27] MIC-DKFZ - nnU-Net (https://github.com/MIC-DKFZ/nnUNet) [Accessed on: 18 October 2021]

[28] kits19-cnn github (https://github.com/jchen42703/kits19-cnn) [Accessed on: 8 November 2021]

[29] kits19-challange - github (https://github.com/nitsaick/kits19-challenge) [Accessed on: 8 November 2021]

[30] kits19 - github (https://github.com/xperience-ai/kits19) [Accessed on: 8 November 2021]

[31] ai campus kits19 (https://github.com/jcausey-astate/ai_campus_kits19) [Accessed on: 3 December 2021]

[32] Eli Gibson, Wenqi Li, Carole Sudreb, Lucas Fidona, Dzhoshkun Shakira, Guotai Wanga, Zach Eaton-Rosen, et. al. - NiftyNet: a deep-learning platform for medical imaging (https://www.sciencedirect.com/science/article/pii/S0169260717311823) [8 November 2021]

[33] Neuroimaging Informatics Technology Initiative (https://en.wikipedia.org/wiki/Neuroimaging_Informatics_Technology_Initiative) [Accessed on: 3 December 2021]

[34] DICOM - wikipedia (https://en.wikipedia.org/wiki/DICOM)[Accessed on: 5 December 2021]

[35] Gumeo - (StackExchange) -Dice Score function (https://stats.stackexchange.com/questions/195006/is-the-dice-coefficient-the-same-as-accuracy)[3 December 2021]

[36] JunMan11 - Loss functions for image segmentation (github) (https://github.com/JunMa11/SegLoss) [3 December 2021]

[37] KA-KA-shi - Left Atrial Segmentation Challenge (https://www.kaggle.com/adarshsng/heart-mri-image-dataset-left-atrial-segmentation/version/1) [Accessed on: 3 December 2021]

[38] MICCAI challange 2017 -Automated Cardiac Diagnosis Challenge (ACDC) (https://www.creatis.insa-lyon.fr/Challenge/acdc/databases.html) [Accessed on: 3 December 2021]

[39] PyTorch Image Sementation Tutorial with U-NET(https://www.youtube.com/watch?v=IHq1t7NxS8k&t=118s) [Accessed on: 3 December 2021]

[40] Olaf Ronneberger, Philipp Fischer, Thomas Brox - U-Net: Convolutional Networks for Biomedical Image Segmentation (https://arxiv.org/abs/1505.04597) [Accessed on: 3 December 2021]

[41] Left Atrial Challenge 2013 Dataset (https://www.kaggle.com/datasets/adarshsng/heart-mri-image-dataset-left-atrial-segmentation) [Accessed on: 12 April 2022]

[42] Voxel spaceing in MRI images (http://jpeelle.net/mri/general/preliminaries.html) [Accessed on: 12 April 2022]

[43] Morpholocial Transformations - OpenCV (https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html) [Accessed on: 13 April 2022]

[44] Binary Cross Entropy Loss - PyTorch (https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html) [Accessed on: 13 April 2022]

[45] BCEWithLogitsLoss - PyTorch (https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html) [Accessed on: 13 April 2022]

[46] Binary Cross Entropy/ Log Loss for Binary Classification (https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/) [Accessed on: 14 April 2022]

[47] Gentle Introduction to the Adam Optimization Algorithm for Deep Learning by Jason Brownlee (https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/) [Accessed on: 14 April 2022]

[48] Left Atrial Volume Index Predictive of Mortality Independent of Left Ventricular Geometry in a Large Clinical Cohort With Preserved Ejection Fraction by Dharmendrakumar A. Patel, MD, MPH, Carl J. Lavie, MD, Richard V. Milani, MD, and Hector O. Ventura (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3146373/) [Accessed on: 15 April 2022]

[49] Normal reference ranges for left and right atrial volume indexes and ejection fractions obtained with real-time three-dimensional echocardiography by Erlend Aune, Morten Baekkevar, Jo Roislien, Olaf Rodevand, Jan Erik Otterstad (https://academic.oup.com/ehjcimaging/article/10/6/738/2366849) [Accessed on: 15 April 2022]

[50] Qiao Zheng. Deep Learning for Robust Segmentation and Explainable Analysis of 3D and Dynamic Cardiac Images. Artificial Intelligence [cs.AI]. Inria - Sophia Antipolis, 2019. English. tel-02083415v1 (https://hal.inria.fr/tel-02083415v1/document) [Accessed on: 23 April 2022]

[51] Convolution, Padding, Stride, and Pooling in CNN by Abhishek Kumar Pandey (https://medium.com/analytics-vidhya/convolution-padding-stride-and-pooling-in-cnn-13dc1f3ada26) [Accessed on: 1 May 2022]

[52] A Gentle Introduction to the Rectified Linear Unit (ReLU)by Jason Brownlee in Deep Learning Performance (https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/) [Accessed on: 1 May 2022]

[53] BCELOSS - PyTorch (https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html) [Accessed on: 5 May]

[54] BCEWithLogitsLoss - PyTorch (https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html) [Accessed on: 5 May]