Right coronary artery labelling with point annotations

*Master Thesis*

Laura Ferrero Montes

Supervisors:
Vlado Menkovski
Javier Oliván Bescós
Tanir Özcelebi

Eindhoven, August 2019
Abstract

Coronary Artery Disease (CAD) is a disorder that affects the coronary arteries and it is characterized by the narrowing or blockage of these vessels. Since they provide oxygen-rich blood to the cells of the cardiac muscle, CAD can produce an angina or heart attack. This disorder is diagnosed using angiography and this imaging technique is also used during the treatment of this disease so physicians can determine the exact position of the lesion in real-time. The pathological findings associated with CAD must be reported per arterial branch. In consequence, an automatic labelling of the different branches will be very useful for physicians and it will also be beneficial for the quantitative characterization of the coronary structure.

Research in this field has been focused on the extraction of the centerlines of the arteries and the classification of each of them in one of the coronary artery tree divisions. A common strategy used to label the centerlines is to map them to a model that encodes the clinical knowledge but the main challenge of the arterial structure is the heterogeneity among patients that is difficult to include in such a model and the local similarity among the segments. Such approaches, however, depend on the quality of the extracted centerlines so they cannot solve directly the problem and no previous research has tried to identify the branches directly from the images.

Deep learning has shown promising results in other tasks related with medical imaging but the main limitation of deep learning for medical data is the amount of information necessary to train the networks. In addition, the ground-truth data must be annotated by medical experts. To overcome this problem, this project proposes an algorithm to label the different segments of the right coronary tree by the regression of the start and end point of each branch. Therefore, the annotation process is more efficient. In addition, this proposal directly works with angiographies, thereby it does not need previous steps as segmentation or centerline extraction.

This algorithm has a fully convolutional architecture and it includes the hierarchical structure of the data. The information from points that are connected in the coronary tree is included for the final prediction of the model. It was trained with 111 angiographies and evaluated with 23 images. The performance of the model was measured with the average Euclidean distance between the predicted coordinate of the point and the ground-truth. The error obtained in the location of the different points varies from 4 to 8% of the image. However, more images that cover a higher amount of the data heterogeneity would be helpful to improve the generalization of the model.
Acknowledgements

Before starting with the description of my project, I would like to express my gratitude to everybody that helped me to reach this goal. I would like to thank Vlado Menkovski, my supervisor at the university, for his guidance and supervision during the process and also for his support. From Philips, I want to thank Javier Oliván Bescós who helped me to start with my own approach but also gave me freedom to experiment with different ideas. Without his suggestions when I was feeling stacked I could have never finished it. I also want to thank Roy van Pelt for his supervision of my writing and also for reminding me which was the clinical goal of the project.

I want to thank my family and my boyfriend who were really supportive when I was feeling really stressed and especially the last weeks.

I also want to express my appreciation to my friend Teresa who was also working at Philips for all the hours we spent giving support to each other, for the coffee breaks and for making my days way funnier.

Finally, I would like to thank my Heistraat family for sharing our pains and joys. I felt at home.
# Contents

Contents vii

List of Figures ix

List of Tables xiii

1 Introduction 1
1.1 Motivation ........................................ 1
1.2 Problem statement .................................... 2
1.3 Outline .......................................... 2

2 Preliminaries 3
2.1 Clinical background ................................... 3
2.1.1 Topology ..................................... 3
2.2 Deep learning background ................................ 5
2.2.1 Convolutional neural networks ......................... 6

3 Related literature 9
3.1 Coronary tree labelling .................................. 9
3.1.1 Centerline extraction ............................... 9
3.1.2 Annotation .................................... 11
3.1.3 Deep Learning .................................. 15
3.2 Human Pose Estimation ................................ 19

4 Dataset 25

5 Methodology 29
5.1 Convolutional Neural network architecture ....................... 29
5.1.1 VGG16 Fully Convolutional ........................... 29
5.1.2 U-net ....................................... 32
5.1.3 VGG16 Feature Pyramid Networks ....................... 32
5.2 Tree-structure information: Graph neural network .................. 34
5.3 Loss function ....................................... 36
5.4 Generalization techniques ................................ 38
5.4.1 Data augmentation ................................ 38
5.4.2 Semi-supervised .................................. 39
5.4.3 Propagation of annotations ........................... 39
5.4.4 Evaluation .................................... 39

Right coronary artery labelling with point annotations vii
CONTENTS

6 Results and discussion ................................. 41
6.1 Experiments ........................................ 41
6.1.1 Experiment 1: Deep learning registration .......... 41
6.1.2 Experiment 2: Elastic deformation .................. 43
6.1.3 Experiment 3: Loss function ........................ 44
6.1.4 Experiment 4: Neural network ...................... 44
6.1.5 Experiment 5: Graph neural network ................. 46
6.1.6 Experiment 6: Semi-supervised ..................... 48
6.1.7 Experiment 7: Comparison .......................... 48
6.2 Results and discussion .............................. 48

7 Conclusions and future lines .......................... 55

References ............................................... 57
List of Figures

2.1 Different parts of the heart: ventricles and atriums and the location of the coronary arteries (Human heart with coronary arteries, 2018) ................................................. 4

2.2 Classification of the coronary arteries. Ao is the aorta, RCA is the right coronary artery, LCA the left coronary artery that starts as the left main branch (LM) that is bifurcated into the LAD and LCX (Habets et al., 2011) ................................................. 4

2.3 Neural network. Example of a neural network with only one hidden layer that is connected with the input and output layer (Multi layer neural network, 2010) .......... 6

2.4 Convolutional layer. The layer on the left has three feature maps and the layer of the right is being created by applying the filter bank to the previous one. The filter bank is the cube in the middle and the weights are shared among all the neurons of the same map (on the right). Each layer in the filter bank that has a different color are the weights which will be applied to each feature map from the layer on the left. The filter map moves through the units of the previous layer and every neuron of the following layer processes a patch of previous units using the filter map weights (Convolutional network with feature layers, 2019) ................................................. 7

2.5 Transpose Convolution. The weights of the filter bank are applied to all the pixels of the image and then they are combined. The combination is defined by the stride and the padding. In this example, the stride is 2 ................................................. 8

3.1 Image of a right coronary artery tree with the centerlines extracted. It also shows the different points: root point in green, bifurcation points in blue, end points in red and the rest points of the centerlines are simple points ................................................. 10

3.2 Tracking procedure at different points of the vessels. (a) At the point $p_i$ the algorithm draws a circumference and identifies two border points, $b_1$ and $b_2$. The midpoint between $b_1$ and $b_2$ is the next identified point of the path, $p_i + 1$ and the distance between $b_1$ and $b_2$ defines the diameter, $d_i + 1$. (b) Tracking process at a bifurcation where the algorithm finds two pairs of border points (Haris et al., 1999) .............................. 10

3.3 The ray casting procedure at different points of the coronary tree: A and B are bifurcations, C is a simple point and D is outside the structure (Yang, Kitslaar et al., 2011a) ................................................. 12

3.4 The process of matching the (a) Artery Segment Graph to (b) the 2D Model Graph and it results in (c) the Association Graph (Haris et al., 1999) ................................................. 13

3.5 How the weights are assigned to each branch based on the number of end nodes (Yang, Broersen et al., 2011) ................................................. 14

3.6 Binary representation. The figure on the left is the coronary tree and the one on the right is the binary tree-shape representation (Gülşün, Funka-Lea, Zheng & Eckert, 2014) ................................................. 15

3.7 (a) Sphere with the different directions the tracker can follow. (b) The tracker follows the direction with maximum probability (Wolterink, van Hamersvelt, A. Viergever, Leiner & Igum, 2018) ................................................. 16

3.8 Tracking procedure. At the beginning, in the seed point, the model identifies two directions (Wolterink et al., 2018) ................................................. 16

Right coronary artery labelling with point annotations ix
3.9 Neural network with a tree shape. (a) Coronary tree with the different branches. (b) Neural network that has the same topology as the data. (c) The different nodes are connected using LSTM units (Wu et al., 2018) ........................................ 17

3.10 Overview of the neural network architecture. There are $N$ points with $d$ features. PFE are the modules of the encoder and PFP are the modules of the decoder. It has two outputs, the first one is a classification in which it assigns a score to each point of being padding, branch and end point and the second output is a clustering in as many groups as branches (Halupka, Garnavi & Moore, 2019) .................................................. 18

3.11 Points selection and grouping of neighbouring points. At the beginning, there are $N_i$ points with $d + C_i$ features, then the algorithm chooses $N_{i+1}$ points and takes $K$ neighbour points for each of them (Halupka et al., 2019) ........................................................ 18

3.12 Architecture of one encoding node. It starts with $N$ points that have $d + C$ features. It performs sampling and grouping, then tree convolutions followed by a pooling. This is done at multiple scales. Afterwards, the different results are concatenated and go through another convolution to get $N_{i+1}$ points with new features, $d + C_{i+1}$ (Halupka et al., 2019) ................................................................. 19

3.13 Initial stage of the architecture and the following stages. The initial stage uses the whole image and outputs the coordinates of the point. The stage s only uses as input data a bounding box centered in the point predicted by the previous stage and predicts a displacement of the coordinates (Toshev & Szegedy, 2013) ........................................ 20

3.14 The upper part of the architecture sees the image at full resolution and the lower part sees it at half of the resolution. Then, the features obtained at the low resolution are up-scaled and combined with the high-resolution ones in order to get the final output (Noothout, De Vos, Wolterink, Leiner & Igum, 2018) ......................................................... 20

3.15 The first stage of the model extracts the features from the image and obtains the first score maps. Then these maps are concatenated in the following stage with the image features and the maps are refined. This process is repeated in all the stages. Moreover, the result obtain in each stage is considered in the loss of the neural network (Wei, Ramakrishna, Kanade & Sheikh, 2016) ........................................................................ 21

3.16 Composition of two neural networks: part detection and regression network. The outputs of the part detection network are combined with the input image and used as input for the regression network that will output the final score maps (Bulat & Tzimiropoulos, 2016) ........................................................................ 22

3.17 (a) Input image, (b) heat-maps for the left elbow, (c) 2D vector fields of part affinities (Z. Cao, Simon, Wei & Sheikh, 2016) ................................................................. 22

3.18 Schema of the Cascade Prediction Fusion using the Hourglass network. The predictions from the different stages are combined to get the final prediction. Moreover, the output from each stage is used as a prior for the following one (Zhang et al., 2019) ........................................................................ 23

3.19 Feature Pyramid Network. It predicts at different dimensions so it is scale invariant (Lin et al., 2016) ........................................................................ 23

4.1 Acquisition angles. (b) C-arm system showing the two different acquisition angles: RAO (right anterior oblique) and LAO (left anterior oblique). (a) and (c) are two images of the right coronary tree projected at LAO and RAO respectively. .................. 25

4.2 Frames from an angiography sequence. Their acquisition order is from left to right, the image on the left does not have contrast in the vessels, the following one shows how the contrast starts to flow, the third one shows the complete structure of the coronary vessels and the fourth one happens when the contrast liquid leaves the arteries. ........................................................................ 26
5.1 Flowchart of the steps followed to predict the location of the start/end points of the right coronary tree branches. It starts with the images from the dataset that are modified for data augmentation purposes. The following step is a convolutional neural network that takes the images and outputs probability maps (heat-maps). These maps are refined using a graph neural network. The next step are different techniques to improve the generalization of the model: semi-supervised learning and propagation of the annotations to other images from the same sequence of acquisition by registration. Finally, the results are evaluated by computing the Euclidean distance between the located point and the ground-truth. 

5.2 Ground-truth heat-maps. Image on the left is a sample from the dataset that has the five points to identify highlighted. The five images on the right are the heat-maps corresponding to each point which are created with a Gaussian distribution with a standard deviation of 45 pixels. 

5.3 Architecture of the VGG16 Fully Convolutional (VGG16-FCN) which is composed by an encoder and a decoder. 

5.4 Architecture of the U-net. It is composed by an encoder and a decoder that are symmetric (Ronneberger, Fischer & Brox, 2015). 

5.5 Graph Neural Network. The image on the left is a sample from the dataset which has annotated the five points to identify. The schema on the right shows the graph that models the structure of the right coronary artery tree. Each point in the image is associated with a node in the graph and the branches of the coronary tree delimited by the annotated points are associated with the arcs of the graph. 

5.6 Gated Recurrent Unit. Schema (a) shows the graph that models the structure of the right coronary artery tree. Each nodes receives information from its parent and child node; the first node only gets information from the child node and the last node only gets information from the parent node. (c) Each node has as many Gated Recurrent Units (GRU) as number of times the hidden state is updated and they share the weights. This example shows how the hidden state of a node (Node 1) is updated twice. $h^0_1, h^1_1, h^2_1$ are the hidden states of the node at different time steps and $x^0_1, x^1_1$ are the combination of the hidden state of the parent and child nodes also at different steps. 

5.7 Limitations of Mean Squared Error (MSE) loss. Image (a) shows the ground-truth, and (b) and (c) are different predictions. The pixel-wise MSE between (a) and (b) and (a) and (c) will have the same value, although the prediction is closer in (b) than (c). Therefore, MSE is not the ideal function that the model should optimize. 

5.8 Calculation of the coordinates for the loss function. The normalized heat-map produced by the neural network is used to get the average $x$ and $y$ pixel coordinates. The coordinates are normalized between -1 and 1 so the network is forced to learn heat-maps which are symmetric with respect to the point. 

5.9 Labels. The image on top is a sample from the dataset with the end point of the catheter highlighted. The images on the bottom are the labels: the two images on the left are the coordinates of the vectors that centered in each 8x8 patch of the image point towards the landmark position and the image on the right shows the classification label in which only the patch that contains the point has a value of 1. 

6.1 Three examples of registration results. The images from left to right are the reference, moving and the transformed image using the deformation field produced by the neural network. The pair of images was compared before and after the transformation using the Mean Squared Error (MSE) and Structural Similarity Index (SSIM). 

Right coronary artery labelling with point annotations
6.2 Comparison of the similarity between the images to map before and after the transformation using deep learning. The $x$ axis represents the differences in the metrics after the registration and before and the $y$ axis is the number of images. The graph on top shows the results compared using Mean Squared Error (MSE) and the lower graph shows the results with Structural Similarity Index (SSI).

6.3 Elastic deformation. (a) Image from the dataset which is (b) fully deformed that causes the borders to lose their square shape and (c) only the difference between image (a) and the background image is deformed and the result is added to the background image. Although the image (c) shows a defined catheter, it shows another one less defined caused by the presence of the catheter in the background image.

6.4 Error with test data computed at the end of each training epoch. At the end of each epoch, the weights of the model were used with the test data and two different error measures were computed: Mean Squared Error (MSE) and Euclidean distance in pixels. The graph on top shows the MSE which is the loss function the model is trying to reduce; it is the average among all the five predictions. The bottom part of the figure shows the Euclidean distance error compute for each one of the points separately. Point 1 is the start point of the catheter; Point 2 is the end point of the catheter; Point 3 is the end point of the proximal division; Point 4 is the end point of the mid division and Point 5 is the end point of the distal division.

6.5 Graph Neural Network models. (a) Simple graph that only draws arcs among points that are connected in the right coronary tree; (b) Graph model that adds an extra connection between the last point (end point of the distal division) and the root (start point of the catheter).

6.6 Evolution of the loss value during training and validation using the VGG16-FCN when the Normal distribution around the point of interest that the model must learn have a variance of 45 or 25 pixels.

6.7 Box and whiskers plot of the error between the prediction and the ground-truth in pixels for the start and end point of the catheter. While the line inside the boxes represents the median of the values, the cross is the mean value. Each of the boxplot represents the error obtained with a different model: VGG16-FCN, the previous one combined with a simple graph neural network and the combined with a loopy graph model.

6.8 Box and whiskers plot of the error between the prediction and the ground-truth in pixels for three points located in the right coronary tree. While the line inside the boxes represents the median of the values, the cross is the mean value. Each of the boxplot represents the error obtained with a different model: VGG16-FCN, the previous one combined with a simple graph neural network and the combined with a loopy graph model.

6.9 Predictions in the test set. The points highlighted are the correct annotations in green and the predictions in red. Each column shows the predictions inferred with a different model.

6.10 Predictions in the test set. The points in green are the ground-truth annotations and the points in red are the predictions. Each column shows the predictions inferred with a different model. The image on top left shows that the fourth and five points are predicted together.


## List of Tables

6.1 Average Euclidean Distance error and standard deviation (in pixels) in the prediction for the test data. Results obtained when the model was trained using the Mean Square Error (MSE) loss and the Differentiable Spatial to Number Transform (DSNT) loss. ....................................... 44

6.2 Average Euclidean Distance error and standard deviation (in pixels) in the prediction for the test data. Results obtained with two models: VGG16-FCN and VGG16-FPN. ......................................................... 45

6.3 Minimum and maximum value of the patch that has the highest likelihood of containing the point of interest among the test samples. ......................................................... 48

6.4 Summary of the average Euclidean Distance error and standard deviation (in pixels) in the prediction for the test data. ......................................................... 49
Chapter 1

Introduction

1.1 Motivation

Cardiovascular diseases are a group of disorders of the heart and blood vessels and they are the main cause of death worldwide (Cardiovascular diseases (CVDs), n.d.). Coronary Artery Disease (CAD) is a disorder that affects the coronary arteries and it is characterized by the narrowing or blockage of these vessels and it is the most common type of cardiovascular disease. CAD is generally originated by atherosclerosis that occurs when a plaque is developed in the wall of the arteries and in the case of the coronary disease, it occurs in the inner wall of the coronary arteries (Thej, Kalyani & Kiran, 2012). The plaque is created by the accumulation of deposits made of cholesterol and other cellular products contained in the blood. The forming process of the plaque usually starts in a part of the vessel wall that was damaged. Eventually, this plaque can harden and it can narrow the lumen of the arteries. Since they provide oxygen-rich blood to the cells of the cardiac muscle, CAD can result in an angina or heart attack. Moreover, if the surface of the plaque breaks, it starts a process in which blood cells called platelets will cover the injury trying to repair the artery which can narrow it more (Boudoulas, Triposciadis, Geleris & Boudoulas, 2016).

Percutaneous Coronary Intervention (PCI) is the main treatment for the CAD. During the intervention, a catheter which has a deflated balloon is introduced through big vessels like the femoral artery to the region of the occlusion and the balloon is inflated. Once the artery is opened, the balloon is deflated and the catheter is removed (Percutaneous coronary intervention, n.d.). PCI is a type of a group of interventions called Image Guided Therapy (IGT). These therapies are characterized by the use of imaging systems to support physicians in the decision, guidance, treatment and confirmation of the right care during the interventions. These systems display pre-operative and real-time image information and they are integrated with different technologies for imaging, guidance and therapy. In contrast with traditional approaches, IGT provide a life view of the anatomical structure without the risks of the open surgery (Image-guided therapy, n.d.). Therefore, during PCI, the physicians can now the exact position of the catheter in real time.

Image-guided surgeries together with image-guided minimally invasive interventions have emerged as an alternative to traditional invasive approaches, allowing doctors to operate with less body damage. It entails less suffering for the patient, shorter stays in the hospital and lower complications likelihood. Therefore, they have proven to be more efficient and less expensive than traditional interventions (Nugteren, 2010).

The imaging system used to visualize the coronary arteries for diagnosis and intervention is the X-ray cardiac angiography. Physicians have to capture the patient anatomy of the coronary artery tree during the diagnosis and after the treatment. This information will be included in the medical record of the patient and it is used by other physicians that want to study the case or to monitor the patient evolution.

The medical record includes the description of the coronary artery morphology, the devices used during the intervention and the pathological findings that are reported per segment or per...
CHAPTER 1. INTRODUCTION

artery. The anatomy of the arteries is heterogeneous and patients can have some missing segments. In addition, the differences among the different branches are not always evident so it is a time-consuming task for the doctors. Therefore, an algorithm that automatically labels the different segments will be beneficial for the doctors since it will help to reduce their workload during reporting and it will also be useful for the quantitative characterization of the coronary structure.

1.2 Problem statement

This thesis attempts to address the problem of the automatic labelling of the coronary arteries. Research on this field has been focused on the extraction of the centerlines of the arteries and the classification of each of them in one of the coronary artery tree divisions (Chalopin, Magnin & Finet, 1998; Chalopin, Finet & Magnin, 2002; Haris et al., 1999; Yang, Broersen et al., 2011; Q. Cao et al., 2017; Gülşün et al., 2014; Liu, Hou, Qin & Hao, 2015; Akinyemi, Murphy, Poole & Roberts, 2009). Such approaches, however, depend on the quality of the extracted centerlines so they cannot solve directly the problem and to the best of our knowledge, no previous research has tried to identify the branches directly from the images. Since deep learning techniques have shown promising results in other tasks related with medical imaging like segmentation, lesion identification and classification between healthy and pathological images (Anas, Mousavi & Abolmaesumi, 2018; Akkus, Galimzianova, Hoogi, Rubin & Erickson, 2017; Samala et al., 2017; Shin et al., 2016), in this thesis existing deep learning techniques and architectures are studied with the aim of finding the best approach for this problem.

The main limitation of deep learning for medical data is the amount of information necessary to train the networks. In addition, the ground-truth data must be annotated manually by medical experts. The annotation of a labelling task consists of selecting all the pixels of the image that belong to each branch so it is time-consuming. To overcome this problem, we have used an alternative approach to annotate the data in a faster way in which only the start and end points of each division of the coronary tree are necessary. Therefore, the problem is transformed into a point location task and the divisions will be defined as the segments between two points. On the other hand, with the objective of keeping the problem simple due to the lack of data, this work will only focus in the right coronary artery tree.

The aim of this work is to find a deep learning architecture that identifies the start/end points of the different divisions of right the coronary artery from angiographies.

The contributions of this work as presented as follows:

- Labelling of the right coronary artery without the need of extensive data annotation and directly from angiographies so previous steps like segmentation and centerline extraction are not necessary.
- Inclusion of information related to the topology of the right coronary tree in the labelling process.
- Generalization to unseen cases with limited data for training.

1.3 Outline

The thesis is presented as follows: chapter 2 provides an explanation of related concepts to this work such as the anatomy and function of the coronary arteries and convolutional neural networks; chapter 3 discusses the related literature; chapter 4 introduces the dataset that has been employed; chapter 5 explains the experimental setup including the proposed architectures and the different techniques employed; chapter 6 consists of experiments and results and includes the discussion and finally chapter 7 contains the conclusions of the thesis and future possible lines of research.
Chapter 2

Preliminaries

Before the review of the different approaches used to address this problem and the introduction of our own approach, this chapter provides some clinical background and some key concepts of Deep Learning. It is not a comprehensive explanation but provides a general description that may be useful to follow this work.

2.1 Clinical background

The coronary arteries are vessels located around the heart (Figure 2.1) that supply oxygen-rich blood to the cells of the cardiac muscle. The lungs enrich the blood with oxygen that goes to the left atrium. The contraction of the atrium moves the blood to the left ventricle. Then, the aorta (Figure 2.1) collects this blood and bifurcates into different vessels that spread it around the body. The coronary arteries are branches connected to the aorta which distribute the blood enriched with oxygen to the heart (Coronary artery disease (atherosclerosis), 2019).

2.1.1 Topology

There are two different branches that emerge from the aorta: the left coronary artery tree and the right coronary artery tree. The left artery supplies the left part of the heart and it bifurcates into the left anterior descending artery (LAD) and the left circumflex artery (LCX). The right coronary artery (RCA) irrigates the cells of the right side of the heart. RCA, LAD and LCX also have different divisions. However, the arteries present different variations in their branching pattern and also in their origin, course and termination. Figure 2.1 shows the location of the different arteries in the surface of the heart (Rao, Pimpalwar, Yadu & Yadav, 2017; Das, Das, Dipak & Talukdar, 2010).

The minimal features that characterize the different arteries are the following (Angelini, 2002):

- LAD: It is located in the anterior interventricular sulcus.
- LCX: It is situated at the left atrioventricular sulcus.
- RCA: It is situated at the right atrioventricular sulcus.

The Posterior Interventricular Artery (PIA) is a subdivision of the coronary arteries that irrigates a small posterior part of the interventricular septum and the posterior aspect of the right ventricle. The origin of this branch determines a property called the dominance of the heart. The origin of the PIA from the RCA is called right dominance and it is called left dominance in those cases in which the branch is bifurcated from the LCX (Bridgeman & Adds, 2015).

The reference classification of the different coronary divisions used for this study is the one proposed by the American Heart Association (AHA) Figure 2.2. It classifies the arteries in 17 segments. Figure shows how the branches are divided.

The different segments shown in Figure are defined as follows (Austen et al., 1975):

Right coronary artery labelling with point annotations
CHAPTER 2. PRELIMINARIES

Figure 2.1: Different parts of the heart: ventricles and atriums and the location of the coronary arteries (Human heart with coronary arteries, 2018).

Figure 2.2: Classification of the coronary arteries. Ao is the aorta, RCA is the right coronary artery, LCA the left coronary artery that starts as the left main branch (LM) that is bifurcated into the LAD and LCX (Habets et al., 2011).

• RCA:
  - 1: Proximal. From the ostium of RCA to one-half the distance to the acute margin of heart.
  - 2: Mid. From 1 to acute margin of heart.
  - 3: Distal: From 2 to the beginning of the posterior descending branch (4).
  - 4: Posterior Descending. It starts from 3.
  - 17: Minor branches that some patients have and other do not.
CHAPTER 2. PRELIMINARIES

- LCA:
  - 5: Left Main (LM). From the ostium of LCA to the first bifurcation.

- LAD:
  - 6: Proximal. From the bifurcation until the origin of the first side branch including it.
  - 7: Mid. From 6 until the LAD forms an angle and if this angle cannot be identified, it ends at half of the distance between its origin and the apex of the heart.
  - 8: Apical. The last segment of the LAD.
  - 9: First diagonal. It has its origin in the proximal segment and it is the first diagonal branch.
  - 16: Side branch connected to 6 that can be present or not.

- LCX:
  - 11: Proximal. First segment of LCX until the obtuse marginal branch starts.
  - 12: Obtuse Marginal (OM): Largest branch of the LCX.
  - 13: Distal. From the origin of OM that is situated in the posterior left atrioventricular groove.
  - 14: Posterolateral (PL): Second side branch from LCX. This segment can be absent or a bifurcation from the OM.
  - 15: Posterior Descending (PD): Last segment of the main branch that can also be absent.

2.2 Deep learning background

Deep learning is a sub-field of machine learning that is inspired by the functionality and structure of the human brain. The target of any machine learning method is to create a model that, given sample data, could make a prediction for a specific task for which it was trained. A model in deep learning, also called neural network, is built using different processing layers, each of them composed by a group of neurons. All deep learning models have an initial layer that is connected with the input data and a final layer that produces the result. In addition, there can be a variable number of layers connecting the input with the output layer that are called hidden layers. Since the layers are connected, the neurons can share information during the processing of data. An example of a neural net with one hidden layer can be seen in Figure 2.3.

Neurons are processing units that transform the data received from the previous layer and send it to the following one so the neural net learns representations of the data at different levels of abstraction. Each processing unit applies a linear transformation to the input defined by two parameters, weight and bias, as shown in Equation 2.1. Then, another transformation that can is applied to the result of Equation 2.1 as it is shown in Equation 2.2 (LeCun, Bengio & Hinton, 2015). This function can be linear or non-linear like ReLU (Nair & E. Hinton, 2010).

\[
z = wx + b \tag{2.1}
\]

\[
y = f(z) = f(wx + b) \tag{2.2}
\]

\(w\) are the weights, \(b\) the bias, \(x\) is the input, \(y\) is the output and \(f\) is the activation function.

Regarding the training of the neural net for a specific task, the model needs an error function that measures the distance between the result produced and the desired one. There are a set of algorithms that can be used to train the model but the most employed one is back-propagation.
The process is iterative, every time the model receives new data, it computes the derivative of the error function with respect to the parameters of the model (i.e. gradients) and they are updated as it is shown in Equation 2.3. Therefore, the model tries to reduce the error by updating the parameters of every neuron transformation so the output gets closer to the ideal result.

\[ \Delta w = -\alpha \frac{\partial E}{\partial w} \] (2.3)

\(\alpha\) is the learning rate and it controls to what extend the weights are adjusted in each iteration and \(E\) is the error function.

The simplest model architecture is the Feed-forward Neural Network (FFNN) in which the information goes from the input layer to the output layer without forming any cycle or loop. The multilayer perceptron is a type of FFNN and it is one of the basic structures from which other more complex architectures were derived. The output of a neuron goes to all the neurons of the following layer so it has a fully connected structure (shown in Figure 2.3) (Mlakarand & Boznar, 2011).

2.2.1 Convolutional neural networks

Convolutional Neural Networks (CNN) are a type of neural networks designed to process data in form of multiple arrays and they have been mainly used for the analysis of visual imagery. However, there are many other applications that have data in form of multiple arrays like signals, sequences, audio spectrograms or video (LeCun et al., 2015).

Fully connected architectures have some limitations when dealing with multiple array data. Since the data is high dimensional, the neural network has many parameters to train which can cause overfitting if the data available for training is not enough. Overfitting is a modelling error that happens when the model achieves good results with the data used for training but when it tries to fit unseen data, the results are poor. In addition, the memory requirement to store all the parameters can lead to hardware level limitations. On the other hand, another drawback of fully connected neural networks is that they ignore the topology of the input because the data is treated as a flat vector of features. However, images and signals have a 2D and 1D strong local structure respectively, which in the case of image means that pixels that are close are highly correlated and these properties could be exploited to get more accurate results. CNN are designed to overcome these constraints using local receptive fields, shared parameters and the possibility of sub-sampling (Bengio & Lecun, 1997).

The architecture of a CNN is a combination of two types of layers, convolutional and pooling layers and they can also have fully connected layers in the last stages. The neurons in a convolutional layer are organized in feature maps and each feature map gets all the feature maps from the previous layer. Each unit processes a local patch from every previous feature map using a

---

Figure 2.3: Neural network. Example of a neural network with only one hidden layer that is connected with the input and output layer (Multi layer neural network, 2010).
A group of weights called filter bank which is shared among all the units in the same feature map.

The receptive field of a neuron is the hyperparameter that defines the size of the local region that is processed by it, i.e. the size of the filter. Therefore, the number of parameters to train in a layer is reduced to the number of feature maps multiplied by the size of the filter bank and the number of feature maps from the previous layer. Figure 2.4 shows an example of a convolutional layer, the filter bank is in the middle and the values are the weights used to process the data by the layer on the right. Besides reducing the number of parameters to tune, another advantage of the convolutional layer is that it can identify distinctive local patterns created by local groups of values that are often highly correlated. Furthermore, considering that the weights are shared among the units, it could detect a concrete pattern at any location (LeCun et al., 2015).

The filter bank which is also called kernel can have different sizes defined by the user and it is moved through the output from the previous layer. Every time it is moved, it samples the data covered by the filter and it is processed by a neuron. The displacement of the filter can be done with a space of one pixel so there will be a space of one pixel between each subsequent sample. This displacement space is called stride and it can also be defined by the user. The size of the convolutional layer output depends on the stride size; bigger strides will produce smaller outputs. There is another parameter in the convolution that can be defined by the user that is the padding. Padding is the addition of pixels of value 0 around the borders of the image so it changes its dimension and also the dimension of the output (LeCun et al., 2015).

![Figure 2.4: Convolutional layer. The layer on the left has three feature maps and the layer of the right is being created by applying the filter bank to the previous one. The filter bank is the cube in the middle and the weights are shared among all the neurons of the same map (on the right). Each layer in the filter bank that has a different color are the weights which will be applied to each feature map from the layer on the left. The filter map moves through the units of the previous layer and every neuron of the following layer processes a patch of previous units using the filter map weights (Convolutional network with feature layers, 2019).](image)

There is another operation called Transpose Convolution which reverts the spatial transformation of the convolution operation. While the convolution can reduce the dimension of the data based on the size of the filter, the stride size and the padding, the transpose convolution upscales the data. Every pixel in the image is multiplied with all the weights of the kernel so the result of a pixel has the same size as the filter bank. Then the results from all the pixels are combined by addition. The combination is defined by the stride. Figure 2.5 shows a schema of the transpose convolution operator with stride 2 and without zero padding.
Whereas the role of the convolutional layer is to detect local patterns from the previous layer, the pooling layer merges semantically similar features. The feature maps obtained after a convolutional layer capture the precise position of the local patterns they could detect and, in consequence, small changes in the position of the motif in the input images can result in a different feature map. This problem is addressed by adding pooling layers after the convolutions. A pooling layer reduces the dimension of the feature maps by summarizing the information contained in each of them. Each pooling unit aggregates the information contained in a local patch of units in a feature map and the aggregation can be done using the average or the maximum value. Neighbour pooling units take input from patches that are shifted by more than one unit in the feature map, hence reducing dimensionality. Thus, the position of each motif is coarse-grained so feature maps still contain important structural elements without fine details and it allows the model to create data representations invariant to small shifts (LeCun et al., 2015).
Chapter 3

Related literature

In this chapter, relevant literature with regards to coronary artery tree labelling is reviewed. Moreover, another approach called Human Pose is discussed due to its similarities to the current problem. In section 3.1, an overview of different approaches to solve the problem of coronary tree annotation is presented, including state of the art algorithms. Then, in section 3.2 different ways to approach the Human Pose challenge are explored and similarities between the two different problems are highlighted.

3.1 Coronary tree labelling

This section provides a summary of all the relevant methodologies used to annotate the different branches of the coronary tree. This problem has been divided in two sub-problems: artery centerline extraction and the annotation of the segments. The first process consists of obtaining the skeleton of the arteries, so it encodes the geometrical and topological information of the tree structure and the second one is the labelling of each segment of the skeleton. Regarding Deep Learning, subsection 3.1.3 describes the different works that made use of neural networks to solve any of the two sub-problems.

3.1.1 Centerline extraction

The centerline of the coronary tree is defined as a group of points that are located in the centre of the arteries. The points can be divided in four groups as shown in Figure 3.1: root points, simple points, bifurcation points and end points.

The different methodologies used to extract this group of points in 2D and 3D data can be summarized in tracking and morphological operators, region growing and thresholding methods and filtering.

Tracking and morphological operators

Ezquerra et al. (1998) and Haris et al. (1999) presented a tracking method that starts with a seed point defined by the user or automatically selected and it identifies the following centerline point. It uses a circumference with a radius bigger than the width of the arteries centered in the seed point and those points within the circle that have a gradient of intensity higher than a threshold are considered borders. The centerline point is defined as the midpoint in the line that connects both border points and the width of the artery at this point is defined by the length of the line. When there is a bifurcation, the algorithm finds more than one pair of border point so it continues the tracking in two different directions. The circumference is centered in all the points already identified in order to find the following ones. Figure 3.2 (a) shows how the tracking process works and Figure 3.2 (b) shows an example of bifurcation. In addition, Haris et al. (1999) used morphological operators to improve the results of the centerlines and borders detected. However,
CHAPTER 3. RELATED LITERATURE

Figure 3.1: Image of a right coronary artery tree with the centerlines extracted. It also shows the different points: root point in green, bifurcation points in blue, end points in red and the rest points of the centerlines are simple points.

this method fails to detect the borders in images that have noise or when there are many crossings between branches.

Figure 3.2: Tracking procedure at different points of the vessels. (a) At the point $p_i$ the algorithm draws a circumference and identifies two border points, $b_1$ and $b_2$. The midpoint between $b_1$ and $b_2$ is the next identified point of the path, $p_{i+1}$ and the distance between $b_1$ and $b_2$ defines the diameter, $d_i + 1$. (b) Tracking process at a bifurcation where the algorithm finds two pairs of border points (Haris et al., 1999).

Region growing and thresholding

A region growing and thresholding method was proposed by Metz et al. (2007) and Yang, Kitslaar et al. (2011b). The first is a semi-automatic algorithm which starts with a seed point in the arteries defined by the user and then, the algorithm adds neighbouring voxels that have a similar intensity to the voxels already included in the segmentation. The method also identifies bifurcations by computing connected components after every two iterations. If the voxels added during these two iterations are not connected, it assumes that there is a bifurcation. In addition,
CHAPTER 3. RELATED LITERATURE

it also avoids non-vascular structures or the aorta being included in the segmentation with a threshold in the number of voxels that can be added to the segmentation every two iterations. If the threshold is exceeded, these voxels are removed. Subsequently, the skeleton is obtained from the segmentation using an iterative thinning algorithm. Then, in order to get separated segment lines, the user has to select end points and for each of them, the algorithm tries to find the longest path between the first seed point and the end point that has the lowest Euclidean length to the thinned coronary path. Finally, the resulting centerlines are smoothed by applying a Gaussian kernel.

The approach of Yang, Kitslaar et al. (2011b) does not need user intervention. It is a combination of region-growing and the hit-or-miss transform. The hit-or-miss transform is a classic mathematical morphological operator. The structural operator used contains foreground and background pixels and it represents a pattern of pixels to identify in the images. The transformation is done by translating the origin of the operator to all pixels of the image and when there are pixels in the image that have the same pattern as the structural operator, they are highlighted. The algorithm does not rely on seed points selected by the user to start the segmentation. It looks for circular shapes in the image trying to find the aorta and once it is located, it also identifies the starting points of the left and coronary arteries. Afterwards, starting from these points, it performs region growing.

Filtering

The Frangi vesselness filter (F. Frangi, Niessen, Vincken & A Viergever, 2000) that extracts tubular structures from images using Hessian matrix was employed in the procedures of Yang, Kitslaar et al. (2011a) and Liu et al. (2015). Yang, Kitslaar et al. (2011a) proposed a method to improve this filter. Its main limitation is that it creates a step-edge response at the boundary of non-vessel structures like cardiac chambers. The methodology distinguishes between false step-edge responses and real vessel responses using geometric features. It performs ray casting in a local sphere at each voxel and the rays stop when they reach the border of the sphere or the border of a structure. Afterwards, the rays that have the same direction are clustered. The groups that have a large number of rays are penalized and the results of the filter are updated with these results. Therefore, the model can distinguish between voxels included in the vessels and the ones included in other structures. After the segmentation, connected component labelling is used to remove small components with few voxels followed by erosion to get the centerlines of the arteries. Finally, since small gaps can appear in the segmentation due to artifacts in the images, the algorithm checks for unconnected branches close to end points and it draws connections if there are. Figure 3.3 shows the ray casting at different points of the structure, D is not located in the coronary tree so when it performs clustering, the number of rays in the same group exceeds the threshold and it is classified as non-vessel structure. Nevertheless, the boundaries cannot be detected precisely when there are lesions or calcifications.

Liu et al. (2015) use the results of the tubular filter to classify pixels in three groups: foreground (arteries), background and undefined. The labels of each pixel are refined iteratively following the grow cut algorithm. In grow cut, each pixel is defined by its label, intensity and strength. The strength of the pixel depends on its neighbours and if this value weighted by its intensity distance to the neighbours is lower than one of the neighbour, it acquires its label. The process continues until convergence. Once the arteries are segmented, the skeleton is extracted by the location of the source points, which are bifurcation and end points, and tracing the shortest part between the start point and each of them. The result of this step is the coronary artery tree skeleton divided in line segments. This proposal has also some limitations when dealing with images that have artefacts caused by heart movements.

3.1.2 Annotation

Once the centerline segments are obtained, the following step is to assign a label to each of them. The different methodologies used to address this issue can be divided in two categories: mapping
to a clinical model and classification.

**Clinical model**

Chalopin et al. (1998, 2002), Ezquerra et al. (1998) and Haris et al. (1999), represented the extracted centerlines as a 2D graph which has attributes describing the geometry of the coronary tree and it was matched to a 2D reference graph that represents the clinical knowledge and it has the labels of each segments. Therefore, after the mapping, the labels from the reference are translated to the data graph.

In the graph proposed by Ezquerra et al. (1998) and Haris et al. (1999), each node in the graph represents a group of centerline points that pertain to the same artery segment, i.e. sequence of points from a root/bifurcation point to a bifurcation/terminating point. Moreover, each centerline point contains three attributes: position, direction and approximated width of the vessel. The edges of the graph are parent-child relationships. While the nodes in the model correspond to complete vessels, the nodes identified in the created graph may not because the tracking could have stopped at a bifurcation, overlap or lesion. Therefore, the nodes of the graph can correspond to the same vessel and the labels can be repeated. The prior anatomical and geometrical knowledge about the expected coronary structure is represented as a 3D reference graph. The nodes of this graph contain a polyline approximation of the 3D centerlines and the diameter. This model is projected to construct a 2D model graph.

In the work of Ezquerra et al. (1998), the labelling process starts from the root node mapping the labels of the model to the data. Once a segment is labelled, its child can be labelled as an extension or as a different branch. The mapping is done trying to find the labelling result that has the lowest cost function. This cost function is defined to ensure that attributes of the segment in the model and in the graph, are similar, the number of segments labelled is maximum and that segments assigned to the same vessel have also similar attributes. In those cases when an overlap between vessels occurs, it is also encoded in the graph. In order to deal with it, the approach uses other frames from the same angiogram sequence to find the same segment but without crossings and it is used to choose the most suitable label.

The graph matching proposed by Haris et al. (1999) is slightly different. In order to get the best mapping, an association graph is created that constructs all the possible matches at the
different levels. Figure 3.4 shows an example of an association graph. Finally, the sub-graph in the association graph that has the lowest cost associated is considered the best result. The cost function is similar to the function used in the approach of Ezquerra et al. (1998). The graph built from the data is directed so it does not allow for a node to have more than one parent. Therefore, overlaps are not modeled.

Figure 3.4: The process of matching the (a) Artery Segment Graph to (b) the 2D Model Graph and it results in (c) the Association Graph (Haris et al., 1999).

Chalopin et al. (1998, 2002) also used a graph representation of the data and a graph model for labelling. However, the graph representation is different: the nodes correspond to end-point segments and bifurcations and the edges are the vessel segments. The nodes and segments have different attributes that differ between the two studies: Chalopin et al. (1998) used as node attributes the point coordinates in the 3D space and the attributes of the edges were the proximal and distal vessel orientation, the number of the position of the arc and the artery name; Chalopin et al. (2002) also included the length of the segment and nodes were classified as root, bifurcation or leaf nodes. Moreover, instead of using a single projection of the 3D clinical model, it was projected using different angles and the only the projection most similar to the data was used for the labelling process.

The labelling strategy in the study of Chalopin et al. (1998) consists in identifying which segment in the model corresponds to each arc in the data graph. The comparison is done by checking the number of arcs going into and out of the proximal and distal node of the segment the edge position number. When there is more than one possible label, the segments selected are the ones more similar in orientation. Finally, the labels are propagated from the model to the data.

Chalopin et al. (2002) presented a more complex labelling process. The algorithm identifies the root node of the 2D data graph as the LM artery and its distal node is assumed to be the bifurcation that leads to LAD and LCX arteries. Subsequently, the algorithm chooses which segment is the initial arc of the LAD artery and which one is the initial arc of the LCX artery by comparing the proximal orientations between the model and the real data. Once the first arc of LAD and LCX is identified, the algorithm extracts the whole segment as a group of connected arcs fulfilling two assumptions: LAD and LCX cannot cross more than a fixed number of times and the main artery has some geometrical properties. Afterwards the sub-branches are labelled, the approach tracks the main arteries in the 2D data graph and the 2D reference graph and compares the geometrical parameters of the sub-branches. The branches that satisfy the geometrical conditions and that
are issued from a bifurcation node can be labelled. The nodes are classified between bifurcation
nodes and crossing nodes considering the angle that the branches form.

In the study of Yang, Broersen et al. (2011), a 3D model is used that represents the clinical
knowledge. The model only represents the right-dominant type of coronary arteries and it is based
on the 17 segments defined by the American Heart Association (AHA) (Austen et al., 1975). The
side branches belonging to the left or co-dominant types of the coronary tree are not included.

The main branches are first identified, then the other segments are labelled and finally, the
results are improved using clinical criteria. The main branches are identified by a point-set reg-
istration between the already extracted centerlines of the coronary tree and the model. The
side branches except RPLB are removed and points along the centerlines are assigned to different
weights in the registration that are equal to the number of end points as shown in Figure 3.5.
Thus, the registration is guided by the main branches. Subsequently, the distance of each pathline
from the ostium to an end point is computed in the real data and the aligned model. The pathlines
that are closer to the main branches in the model receive their labels.

Figure 3.5: How the weights are assigned to each branch based on the number of end nodes (Yang,
Broersen et al., 2011).

The remaining side branches are divided in three sub-trees which have a main branch (identified
in the previous step) and several side branches. The model with all the side branches is deformed
using the rigid transformation obtained in the previous step. For each sub-tree, a path of the
same sub-tree in the model is chosen as it was the main branch identified and since the pathline
is composed by segments with a concrete length, the one identified is divided the same way.
Considering the branch selected, all the side branches are labelled considering all the possible
combinations but following two rules: hierarchical relationship and prior knowledge. The cost of
the division and labelling is computed and the process is repeated iteratively until all the pathlines
of the subtree in the model have been considered. Finally, the labelling result with minimal cost
is selected and it is refined using some rules based on clinical knowledge.

The work of Gülsün et al. (2014) consists on labelling a coronary tree by finding a minimum
cost deformation of a reference model. The cost of the transformation is measured as the Quotient
Euclidean Distance (QED) between the target tree and the deformed model and the likelihood of
the assigned labels to the target coronary tree. The coronary tree and the model are represented in
a high dimensional space that captures the topology and geometry. The tree-shape is represented
as a pair \((T, f)\) where \(T\) is an ordered binary tree with branches and a root point and \(f\) are the
branch attributes that map each branch to landmark points sampled along it. The trees that do
not have a binary shape, are considered as if they have zero-length branches. Figure 3.6 x shows an
example of binary representation. The geodesic metric in this space is the QED and it is the path
with minimum distance cost over all possible paths. The distance cost between two tree-shape
representations is the \(L_2\) norm of the deformation cost between corresponding branches. The
branch deformation is computed as the sum of the Euclidean distance between landmark points
of the two branches.

The Dijkstras algorithm is used to compute the QED geodesic between the target coronary tree
and the atlas. In addition, the algorithm has a second cost term that depends on the likelihood
of where a particular branch tends to be located over the heart territory. Once the path with
the minimum cost is selected, the labels are propagated from the model to the target artery tree.
The main limitation of their approach is that the atlas model only considers right-dominant and
CHAPTER 3. RELATED LITERATURE

Figure 3.6: Binary representation. The figure on the left is the coronary tree and the one on the right is the binary tree-shape representation (Gülsün et al., 2014).

co-dominant circulation.

Q. Cao et al. (2017) used a similar approach to Yang, Broersen et al. (2011) but they extended their methodology to left-dominance coronary artery tree. They used two 3D models with a priori knowledge of the coronary arteries. The right dominant model was obtained from Yang, Broersen et al. (2011) and the left-dominant model was created following a one-in-cross approach with 11 training examples. Each time, one sample was used to create a model and the lengths of the branches were normalized using the average lengths of the 11 training samples. Then this model was validated and the one that got the highest accuracy was selected as the final LD model. The labelling process is the same used in Yang, Broersen et al. (2011) in which firstly the main branches are identified and then the coronary tree is divided in 3 sub-trees that are subsequently labeled.

Classification

Akinyemi et al. (2009) proposed a Gaussian classifier that extracts different metric features from the vessel segments and assigns labels to them. The algorithm uses complete vessels as segments instead of topological segments which are the ones that go from bifurcation point to another bifurcation point (i.e., a vessel segment can be a combination of 3 topological segments). The features measured are: length, length proportion to the tree, angle to its parent, tortuosity, diameter, average diameter, volume, volume proportion to the three and direction. These features are used as input to a Gaussian classifier that computes the mean and covariance for each class. Therefore, it creates a Gaussian function for each class that gives the probability of pertaining to it.

The labelling process is a recursive function that assigns a label for each topological segment. It checks if the labelling is topologically and geometrically legal and computes the merit of the classification. The process is repeated with all possible combinations and at last, the result with the highest merit is recorded. In order to know if the topology and geometry are legal, the model uses some rules based on expert knowledge like how many bifurcations a segment can have.

Although the algorithms presented are different, they all start with the already extracted centerlines of the coronary artery tree. Therefore, their performance is limited to the accuracy of the extraction.

3.1.3 Deep Learning

In this section, different approaches that employed Deep Learning techniques for the labelling of the coronary artery tree are reviewed. Furthermore, other proposals that work with images from the coronary arteries and use Deep Learning but for other purposes are discussed.

In the study of Huang et al. (2018), a CNN was used to work with data of the coronary arteries. However, the goal of this proposal was to segment the arteries and not to label them. The architecture used was a 3D U-net with two parts: encoder and decoder. The encoder increases the number of feature maps while decreasing the size of the data and the decoder restores the high resolution features. In addition, the high dimensional features are concatenated with the low dimensional ones so it can decode the fine details of the objects.
Convolutional Neural Networks have also been used to extract the centerlines of the coronary arteries as proposed in the work of Wolterink et al. (2018). Their approach consists on a tracking method that starts with a seed point located in the coronary tree and predicts the following point until it cannot make more predictions with high certainty. The seed point can be manually or automatically selected and then the model takes an image patch centred in this point as input. The model is a CNN that stacks different convolutional layers with increasing stride value that enlarges the receptive field and it outputs two results: the direction of the following point and the radius of the coronary arteries. Regarding the direction, the model predicts the probability over a discrete set of directions showed in Figure 3.7 (a). In the case of the radius, it can be considered as a regression problem in which the model only outputs one value.

![Figure 3.7](image)

Figure 3.7: (a) Sphere with the different directions the tracker can follow. (b) The tracker follows the direction with maximum probability (Wolterink et al., 2018).

In the first iteration of the model, after the seed point, the model identifies two directions with maximum probability that form an angle equal or higher than 90 degrees between them. Figure 3.8 shows an example of the two directions identified by the model. It starts following the centerline in one direction and once the tracker cannot find more points, it continues in the other direction. The tracker moves from one point to the next one in the direction with maximum probability and with a step defined by the radius. The model stops when it cannot find a direction without uncertainty. This algorithm showed some errors when the arteries had severe calcification, stenosis or sharp angles.

![Figure 3.8](image)

Figure 3.8: Tracking procedure. At the beginning, in the seed point, the model identifies two directions (Wolterink et al., 2018).

Wu et al. (2018) and Halupka et al. (2019) developed two different deep learning methodologies that use the centerlines of the distinct branches of the arteries and assign a label to each of them.

The methodology of Wu et al. (2018) transforms the centerlines to a spherical coordinate system. Then, some features are obtained from the original 3D space of the angiographies and the spherical coordinates. The method inputs the features to a neural network that has long short-term memory and it assigns one class to each vessel segment. In order to normalize the data among patients, the points of the centerlines are converted to spherical coordinates where
the origin is the center of the bounding box and the north is the direction from the origin to the aorta. As a result, the points of the centerlines are represented in a 2D image with the azimuth and elevation angle as the new axes.

Regarding the feature selection, for each vessel segment, defined as the segment between two bifurcations, some features are extracted: parent-child segment angles in 3D and spherical space; spherical coordinates of the first, center and end point and the directions in the spherical space between start and end segment point. The neural network constructs an architecture with the same tree-shape as the artery tree and each segment is a node. Figure 3.9 shows the neural network generated for a concrete sample of data. Each node contains a Multilayer Perceptron (MLP) module, two Long Short-Term Memory (LSTM) units and a Softmax layer. In the model, one of the two LSTM units per node incorporates information from the child nodes and the other one from the parents. Each unit has as many cell states as nodes connected and since the LSTM units have a forget gate, the model can preserve the information from each child/parent selectively. Figure 3.9 (c) shows the connections between three segments of the artery tree. The MLP acts as an encoder for each segment that transforms the features into a compact feature vector. These high-level representations are used by the Tree-LSTM that creates long-term dependencies among segments to predict their labels. The main advantage of this methodology is that it builds a different model for each data sample so it can be adapted to different topologies.

In the labelling of coronary arteries, the complexity of these structures must be represented in the data with high detail which implies high resolution images. Therefore, the analysis of this data is computationally expensive. Halupka et al. (2019) developed a deep learning algorithm that uses the representation of the arteries as centerline points in the Euclidean Space. The model extracts meaningful topological information like the location of the bifurcations, the number of branches and the branches to which point belongs. The coronary tree is represented as an unordered set of 3D points where each point has tree features that are the coordinates. The neural network is composed of an encoder and a decoder and it has two outputs: one output is the classification of points between padding, branch and end point and the second output is a clustering of the points so the points that pertain to the same branch are grouped together. Figure 3.10 shows the schema of the model.

Figure 3.9: Neural network with a tree shape. (a) Coronary tree with the different branches. (b) Neural network that has the same topology as the data. (c) The different nodes are connected using LSTM units (Wu et al., 2018).

In the labelling of coronary arteries, the complexity of these structures must be represented in the data with high detail which implies high resolution images. Therefore, the analysis of this data is computationally expensive. Halupka et al. (2019) developed a deep learning algorithm that uses the representation of the arteries as centerline points in the Euclidean Space. The model extracts meaningful topological information like the location of the bifurcations, the number of branches and the branches to which point belongs. The coronary tree is represented as an unordered set of 3D points where each point has tree features that are the coordinates. The neural network is composed of an encoder and a decoder and it has two outputs: one output is the classification of points between padding, branch and end point and the second output is a clustering of the points so the points that pertain to the same branch are grouped together. Figure 3.10 shows the schema of the model.
CHAPTER 3. RELATED LITERATURE

Figure 3.10: Overview of the neural network architecture. There are $N$ points with $d$ features. PFE are the modules of the encoder and PFP are the modules of the decoder. It has two outputs, the first one is a classification in which it assigns a score to each point of being padding, branch and end point and the second output is a clustering in as many groups as branches (Halupka et al., 2019).

The encoder creates a representation of the points in a fixed feature vector by aggregating information at multiple scales. The input to an encoder node are a group of $N$ points with $C$ features for each of them. The first stage in the encoder is Sampling and Grouping, it chooses $N'$ points and for each of them it takes its neighbours that are defined as the $K$ pixels in a radius $r$ around each point. This process is done in multiples scales, i.e. different radius. Figure 3.11 shows an example of grouping. Then, the feature vector of each point is updated with three convolutional layers and finally there is a Max Pooling layer that works over the $K$ points so the result are a group of $N''$ points with new features, $C''$. Figure 3.12 shows the architecture of one encoding node. There are different encoding nodes that work reducing the number of sampled points but increasing the information contained in each.

Figure 3.11: Points selection and grouping of neighbouring points. At the beginning, there are $N_i$ points with $d + C_i$ features, then the algorithm chooses $N_{i+1}$ points and takes $K$ neighbour points for each of them (Halupka et al., 2019).

Afterwards, the decoder performs up sampling to get the same number of original points. It is done using interpolation and across level links. The points are then classified between end, padding and branch points and they are also clustered in as many groups as branches. Therefore, the model learns with a combined loss. Since the number of clusters depends on the number of branches that exist in the data, it can be generalizable to topologies in which there are some branches missing.

Deep learning approaches for the labelling of coronary arteries, similarly with the algorithms presented in the previous sub-section, use the centerlines of the coronary artery tree. Thereby, the results depend on the quality of the extracted centerlines.
CHAPTER 3. RELATED LITERATURE

Figure 3.12: Architecture of one encoding node. It starts with $N$ points that have $d+C$ features. It performs sampling and grouping, then tree convolutions followed by a pooling. This is done at multiple scales. Afterwards, the different results are concatenated and go through another convolution to get $N_{i+1}$ points with new features, $d+C_{i+1}$ (Halupka et al., 2019).

3.2 Human Pose Estimation

The Human Pose problem consists in the localization of the human joints in order to estimate the body pose in the space of all articulated poses. It can be considered analogous to the coronary tree labelling problem because of several reasons:

- The human body and the coronary tree have a tree shape topology in which the relationships among joints are crucial. For example, the head has to be in a higher level than the shoulders and similarly the LM segment in the left coronary tree will always be followed by the LAD and LCX.

- In both cases, the structure is projected into a 2D image which can cause some parts to be occluded or overlapped.

- Finally, both structures are non static. The musculoskeletal system allows the body to move and the cardiac cycle causes the movement of the arteries.

The last two reasons are challenges that the methodologies must consider and learn to deal with. In consequence, this section will review different approaches to solve the Human Pose problem including state of the art methodologies.

Initially, this problem was addressed by the extraction of body features and the creation of models that used them to locate the different points. However, since the models relied on hand crafted low level features like edges, contours or color histograms, the learning was limited (Toshev & Szegedy, 2013).

Toshev and Szegedy (2013) formulated the Human Pose problem as a joint regression task in which a CNN has to learn the coordinates of the different body joints. Full images that contain different body positions are used as input for the neural network that learns a representation of the data in order to get the coordinates of the joints. The advantage of using neural networks is that the model itself learns the best representation of the data for this task so there is no need to design feature descriptions of the data like model topology or interactions between joints. The CNN has seven layers, the first two convolutions are followed by a normalization and a pooling layer, then after other two convolutions there is another pooling layer and it ends with two fully connected layers. In pursuance of finding precisely the body joints, Toshev and Szegedy (2013) approach concatenates CNN at different stages creating a cascade of pose regressors. In the first stage the model uses the full image and estimates the coordinates of the joints. Afterwards, the model creates a bounding box of a fixed size centered in this point and it estimates a displacement of the coordinates. The process continues at different stages refining the results of the prediction. Figure 3.13 shows the architecture of the cascade pose regressor.

The mapping from an image to a pose vector of coordinate is a non-linear and difficult transformation to learn and it could be the reason why the approach of Toshev and Szegedy (2013) did not show accurate results in high-precision regions. Tompson et al. (2014) presented a different
CHAPTER 3. RELATED LITERATURE

Figure 3.13: Initial stage of the architecture and the following stages. The initial stage uses the whole image and outputs the coordinates of the point. The stage s only uses as input data a bounding box centered in the point predicted by the previous stage and predicts a displacement of the coordinates (Toshev & Szegedy, 2013).

architecture in which the network predicts in two dimensions so it directly shows in which pixels of the image the body joint is likely to be. The output of their model is a group of heat-maps, one for each point skeleton joint, that produce a per pixel likelihood. The model analyses the data at multiple resolutions so it can find the points using different amounts of context information which facilitates the generalization to different situations. Figure 3.14 shows the model which analyses the data at two resolutions: the full image and half of its resolution. The first stage is a convolution layer followed by a ReLU and pooling layer, then both results go through another convolution layer that shares the weights. Afterwards the low-resolution features are up-scaled and added to the high-resolution ones and finally the combined features go through another convolution layer that outputs the group of heat-maps.

Figure 3.14: The upper part of the architecture sees the image at full resolution and the lower part sees it at half of the resolution. Then, the features obtained at the low resolution are up-scaled and combined with the high-resolution ones in order to get the final output (Noothout et al., 2018).

However, the heat-maps can contain more than one possible joint location that are anatomically incorrect. In order to reduce the number of false positives, the authors developed another model called Spatial-Model that learns the conditional distribution of the location of one body joint given the location of another. For example, if in all the images the right shoulder is located lower right of the face, it would not accept points located in different parts.

Although the Spatial-Model reduces the number of outliers, if a dataset covers many possible poses, the conditional distributions would be less constraint and the results may be less accurate. The work of Wei et al. (2016) overcomes this limitation by concatenating different CNN that refine the heat-map prediction from the previous stages. Therefore, another model to refine the results is not necessary since it is all combined together.

They based their approach in the idea of increasing the effective receptive field that each unit
processes so if the field is big enough, it can capture long-range interaction between the different body parts. The belief maps obtained in the first stage, as shown in Figure 3.15, are generated from a network that examined the image locally with a small receptive field. The following stages receive two inputs, the output from the previous stage and image features. The image features are shared among the different stages and they are generated by making the image go through three convolutional layers followed by pooling layers and one extra convolution. The receptive field is increased through the stages and the computed heat-maps are increasingly refined. The model computes the Mean Square Error (MSE) between the prediction and the ground-truth at each stage and all losses are added for the training.

Figure 3.15: The first stage of the model extracts the features from the image and obtains the first score maps. Then these maps are concatenated in the following stage with the image features and the maps are refined. This process is repeated in all the stages. Moreover, the result obtained in each stage is considered in the loss of the neural network (Wei et al., 2016).

Tompson et al. (2014); Wei et al. (2016) architectures predict the belief maps at a lower dimension than the input data caused by the pooling layers. Therefore, the precision of the results is limited to the dimension of the output. The proposal of Bulat and Tzimiropoulos (2016) uses two different CNN for the location of the body joints and it does it at the input scale. The first CNN is a part detection network that has the objective to highlight the areas of the image where the points can be. For this first stage, they did trials with two Fully Convolutional (FCN) models: VGG16-FCN and Hourglass Network. The second model performs the regression of the points focusing in the areas previously detected and it is another CNN that does not have pooling layers so it applies convolutions with padding to keep the dimension of the data and at the end it performs a transpose convolution to recover high dimensional information lost. The output of the first module are belief maps in which each pixel has a likelihood from 0 to 1 of having the joint. Then, these heat-maps are concatenated with the original image and go through the second model that outputs heat-maps with the refined results. The part detection network learns using a cross entropy loss function and the regression network does it with the Mean Square Error (MSE). Figure 3.16 shows a schema of the architecture of the model.

Their main contribution is the robustness of the model against joints that are occluded in the image because when it happens, the first output will produce low confidence scores for occluded parts. Therefore, the regression network instead of focusing on the pixels where the joint should be, it will use contextual information in order to predict the location of these parts.

Introduced by (Long et al., 2015), the FCN networks produce dense predictions at the same resolution as the input data. FCN substitute the fully connected layers for fully connected convolutional layers with a kernel of size one and once the feature maps are at the lowest dimension, it performs upsampling until they get the same dimension as the input data. The upsampling
can be done using interpolation like the bilinear interpolation that for each output pixel uses four neighbour input pixels and combines their values using a linear map that depends only in their relative position. Another upsampling operation is the reversed convolution, also called deconvolution. However, when performing an upsampling from really low dimension to high one, the output is coarsed and the prediction has a limited scale of detail. As a solution, the propose to add links that combine the final layer with lower layers which have finer details so the model can make predictions that respect the global structure of the data.

Z. Cao et al. (2016) developed a model that could detect the 2D pose of multiple people in an image. The neural network had two outputs: heat-maps for the different body parts and a set of 2D vector fields of part affinities that represent the association between different parts of the same person. Figure 3.17 shows the confidence maps and the part affinity fields. The features from the image are obtained using the first ten layers of the VGG16 neural network. Then, the features are used to predict in different stages the confidence maps and affinity fields. The part affinity fields are created for each limb and all the pixels that belong to it have a unit vector in the direction from one part of the limb to the other.

Once the different body parts are obtained, the objective is to match the ones that belong to the same person. First, all the parts are connected in all the possible combinations but keeping the tree shape of the skeleton, i.e. the shoulder has to be connected first to the elbow and then the elbow to the hand. Then, the matching problem is decomposed in a set of bipartite matching sub-problems. For each pair of parts, all the possible connections are weighted by measuring the
alignment of the predicted affine field with the candidate limb that would be formed joining those two parts. Therefore, the model gives the connected pairs that form a limb for every person. Finally, the connections that share the same part are assembled together to get the full body pose.

Though all the works reviewed showed good performance in the Human Pose problem, Zhang et al. (2019) achieved state of the art results. The proposal is based in two ideas: Cascade Prediction Fusion that accumulates the predictions made in previous stages to get the final one and a Graph Neural Network that represents the relationships among connected joints.

Regarding the Cascade Prediction Fusion (CPF), the predictions obtained at different stages are used as a prior for the following stage. They go through a 1x1 convolution to increase the number of features maps and they are merged with the output of the following stage to create the new prediction. The neural network is trained considering the loss of all the predictions. Figure 3.18 shows the architecture of the Cascade Prediction Fusion. The final prediction is obtained by the accumulation of auxiliary predictions. Therefore, the information captured by low level layers like signal localization and the semantic information captured by high level layers are combined to get the final result.

![Figure 3.18: Schema of the Cascade Prediction Fusion using the Hourglass network. The predictions from the different stages are combined to get the final prediction. Moreover, the output from each stage is used as a prior for the following one (Zhang et al., 2019).](image)

Two different neural networks were compared in the work Zhang et al. (2019): ResNet-50 and Stacked Hourglass. The Stacked Hourglass network produces intermediate outputs at the same resolution of the input data but the ResNet-50 fully convolutional has only one output with the same dimension as the input data. Lin et al. (2016) introduced the Feature Pyramid Networks (FPN) as a multi-scale model that constructs features pyramids so the intermediate outputs at different dimensions are considered in the loss function of the model. Figure 3.19 shows an example of FPN. For the Cascade Prediction Fusion, the FPN was adopted in the ResNet-50 so the model could learn at different scales.

![Figure 3.19: Feature Pyramid Network. It predicts at different dimensions so it is scale invariant (Lin et al., 2016).](image)

The other novelty network introduced by Zhang et al. (2019) is the Graph Neural Network. The model builds a graph that has as many nodes as joints and the edges are the connections among joints, i.e. the limbs. Each node has a hidden state that is updated with the information
from the neighbours and also the current state from the node. All the nodes are initialized, i.e. current state, with the heat-maps obtained from last prediction of the Cascade Prediction Fusion. Then, every node gathers the information from its neighbours by applying a convolution on their current state and merging the results. This new information is added to the current state of the node also by means of convolutional operators. This process is repeated $T$ times and the final prediction is the addition between the first state of the node and the final hidden state. This way, the results obtained in the first model are refined taking into account the information from the connected parts.
Chapter 4

Dataset

The dataset used in this project is a group of angiographies taken by the Philips mobile C-arm device. Angiography is an imaging technique that uses radiation to visualize the blood vessels and some organs. This study used vessel images. A contrast agent is introduced into the desired vessels with a catheter and it acts as a radiopaque substance so the vessels appear darker than the background and they can be easily identified.

The images were acquired in two Dutch hospitals that have the Philips system. The Philips mobile C-arm is an X-ray system that can be brought into the operating room to provide live image guidance during interventions. The images are taken while the patient is lying on the table and the acquisition can be done in different angles. In order to get a good view of the coronary arteries, the most used acquisition angles for cardiac angiographies are the left anterior oblique and the right anterior oblique. Figure 4.1 shows the two different acquisition angles and two frames of the right coronary artery tree obtained at these angles. The data used for this project contains images acquired from the two different angles. Therefore, besides the inherent variability of the data caused by the anatomical differences, there are variations in the orientation of the structure.

Figure 4.1: Acquisition angles. (b) C-arm system showing the two different acquisition angles: RAO (right anterior oblique) and LAO (left anterior oblique). (a) and (c) are two images of the right coronary tree projected at LAO and RAO respectively.
CHAPTER 4. DATASET

The system records a sequence of image that starts with the injection of the contrast until it is vanished. Figure 4.2 shows some frames from the same sequence.

![Frames from an angiography sequence.](image)

Figure 4.2: Frames from an angiography sequence. Their acquisition order is from left to right, the image on the left does not have contrast in the vessels, the following one shows how the contrast starts to flow, the third one shows the complete structure of the coronary vessels and the fourth one happens when the contrast liquid leaves the arteries.

The images have a resolution of 512x512 pixels and have three channels that correspond to the RGB components. They were acquired in gray-scale but converted to RGB in order to use pre-trained neural network architectures. The frames have a dark squared around the vessel structure created by the shutter. This system is used to homogenize the trajectory of the radiation so all the rays go parallel and in the direction perpendicular to the object to view. Apart from the differences in anatomy and structure of the right coronary arteries, the heterogeneity of the dataset is also caused by acquisition factors:

- The distance of the body to the scanner changes from acquisition to acquisition so the images show different zoom level.
- The size of the shutter changes so the width of the square dark differs
- Some images have padding pixels in gray around the dark square in order to keep a constant resolution.
- Image contrast depends on the kilo-voltage employed during the examination but it is also affected by the scatter radiation ([Radiographic contrast, 2019](#)). Consequently, it is not a constant feature among the images.

The dataset contains images from 300 different patients. In some cases, more than one sequence were recorded from the same patient so there are in total 3000 sequences. 1396 were taken from the left coronary artery tree, 596 from the right tree and the remaining 1008 did not show any vascular structure. The number of frames per sequence varies from 20 to 100 frames.

The arteries were labelled manually to have a target that the model has to reach. The annotations of the images consisted of the location of the start and end points of each division of the coronary tree. The annotation was based on the 17-segment model defined by the American Heart Association ([Austen et al., 1975](#)) which is described in chapter 2,subsection 2.1.1. The idea behind this type of annotation is to address the labelling of the coronary arteries as the Human Pose challenge but instead of regressing body joints, the aim of this project is to regress the start and end point of the branches. Once these critical points are located, the branches can be defined as the segment between two points. The points to be identified are the following:

1. Start point of the catheter
2. End point of the catheter /start of the right coronary artery that is the proximal segment.
3. End of the proximal division/start of the mid division

Right coronary artery labelling with point annotations
4. End of the mid branch/start of the distal division

5. End of the distal division/start of the posterior descending segment

Whereas the geometry of the right coronary artery tree has a shape as a C that is slightly constant among patients, the structure of the left coronary tree has more variation. On account of this heterogeneity, the aim of this project was focused only in the labelling of the right artery.

The images from 99 patients were annotated and in some cases more than one trial per patient. The patients were first divided into two groups: 80% for training and 20% for testing. The training group was also divided between training data (90%) and validation data (10%). Therefore, in total 111 images were used for training, 2 for validation and 23 for testing.
Chapter 5

Methodology

The goal of this thesis is to build a model that learns how to identify the different branches of the right coronary artery tree, taking into consideration its tree-structure and that generalizes with high precision to different variations in anatomy. In this chapter, different deep learning techniques will be discussed as well as methodologies which aim to improve the generalization of the model. Figure 5.1 shows the different steps of the proposal.

5.1 Convolutional Neural network architecture

The model proposed to identify the start/end points of the right coronary tree is a convolutional neural network. The main benefit of neural networks for this problem is that they learn the best features from the data in order to identify the points so it is not necessary for the user to extract and find the best data descriptors. Moreover, the convolutional neural networks work with images as input data by exploiting their local spatial coherence.

The network uses the images of the right coronary tree and outputs images that have the same size as the input. Therefore, the resolution of the results is not decreased. For each input image, the model will generate five different score maps, i.e. heat-maps, which contain a probability distribution among the pixels of the image of containing each of the five desired points. Figure 5.2 shows the ground-truth heat-maps of the five points to identify. Therefore, the input data has a shape of 512x512x3 and the shape of the output data is 512x512x5.

The neural network architecture is fully convolutional so it maintains the resolution of the images. The constraints of keeping high resolution images through the neural net is that the need for memory allocation is higher and the memory available may not be enough. A possible solution is to reduce the number of feature maps or the number of convolutional layers but shallow neural networks have the risk of underfitting which means that the network cannot model the training data and neither generalize to unseen data.

Another possible solution to reduce the memory storage needed by the algorithm is to reduce the dimension of the data and then rescale it to the original size. These type of architectures usually have an encoding part in which the dimension of the image is decreased and the number of feature maps is enlarged and a decoding part that performs the opposite operation. Moreover, the model will take advantage of the max pooling layers which reduce the dimension and are invariant to shifts or translations of the patterns in the image. For this thesis, three neural network architectures that are fully convolutional and have encoder-decoder structure are considered: VGG16-FCN, U-net, VGG16-FPN.

5.1.1 VGG16 Fully Convolutional

VGG16 Fully Convolutional (FCN) is a modification of the original VGG16. VGG16 is a convolutional neural network that has shown good performance in different classification and localization
CHAPTER 5. METHODOLOGY

Figure 5.1: Flowchart of the steps followed to predict the location of the start/end points of the right coronary tree branches. It starts with the images from the dataset that are modified for data augmentation purposes. The following step is a convolutional neural network that takes the images and outputs probability maps (heat-maps). These maps are refined using a graph neural network. The next step are different techniques to improve the generalization of the model: semi-supervised learning and propagation of the annotations to other images from the same sequence of acquisition by registration. Finally, the results are evaluated by computing the Euclidean distance between the located point and the ground-truth.
tasks with images due to its introduction of 3x3 convolutions filters and increased depth (Simonyan & Zisserman, 2014).

The output of the original VGG16 is a flat vector that has class probabilities for the classification task or bounding box coordinates for the location problem (Simonyan & Zisserman, 2014). The architecture is redesigned to have an output with the same shape as the input data. However, if the intermediate output at the lowest scale (16 pixels) is used for up-scaling, the results will be coarse. Long et al. (2015) proposed as a solution in which they used high resolution features that provide fine information with more details in combination with low resolution features that have semantic information to produce the final output.

Figure 5.3 shows the neural network architecture which was designed by bulat2016 to locate the body joints for the human pose estimation. The first five blocks are composed by two or three convolutions followed by the activation layers and a max pooling layer at the end of the block that reduces the dimension of the data to half. After the fifth block, the data goes through a 7x7 convolutional layer with 4096 features maps followed by another convolutional layer with kernel size of 1x1 and the same number of feature maps as the previous one. This neural network was adapted to the coronary point location problem and in contrast to Figure 5.3 that used ReLU activation function, leaky ReLU function was employed in the intermediate layers. This choice was made to avoid the “dying ReLU” problem that happens when a large gradient provokes a change in the weights in a way that the neuron will never activate again and the gradient that flows through this unit will always be zero. Then, the model has 1x1 convolutional layers that output feature maps that will correspond to the different points to locate. Another difference with respect to the design of Figure 5.3 is the number of output channels, whereas it had 16 in the original proposal, it has 5 for the coronary point location which correspond to the five points to identify. The predictions are done at three different scales, 64x64, 32x32 and 16x16 pixels, and combined.

Right coronary artery labelling with point annotations
CHAPTER 5. METHODOLOGY

Regarding the combination, the output at 16x16 is up-scaled using bilinear interpolation and it is combined with the output at 32x32 resolution using pixel-wise addition. Afterwards, the result is also up-scaled using bilinear interpolation and combined with the output at 64x64 scale and the outcome is up-scaled to reach the dimension of the input data.

Figure 5.3: Architecture of the VGG16 Fully Convolutional (VGG16-FCN) which is composed by an encoder and a decoder.

5.1.2 U-net

The second architecture is the U-net. Ronneberger et al. (2015) modified and extended the fully connected architecture proposed by Long et al. (2015). Their main contribution is the design of an expanding path, i.e. decoder, which is symmetric to the encoder so it has more feature maps than the decoder proposed by Long et al. (2015). Therefore, more context information will be propagated to higher resolution layers from the low resolution ones. In addition, U-net has skip connections that combine the intermediate output of all the scales in the encoder to the up-scaled outcomes of the decoder. In contrast to the VGG16-FCN, U-net uses as up-scaling operator the transposed convolution and the features from low layers are concatenated to the features from high layers instead of computing the addition. Figure 5.4 shows the schema of the U-net.

5.1.3 VGG16 Feature Pyramid Networks

The third neural network to compare is the VGG16- Feature Pyramid Networks (FPN) which is also based in the VGG16 (Simonyan & Zisserman, 2014) but it uses the outputs obtained at different scales of the decoder to learn. The model uses the feature maps obtained at five different resolutions, 32x32, 64x64, 128x128, 256x256 and 512x512 pixels, for the prediction. These intermediate outputs go through two convolution operators with filter bank 1x1 and the weights are shared among all the scales. Since the loss functions includes the error of all the outputs, the model is forced to learn at different dimensions which is useful when the objects to detect in the input data have different sizes. Feature Pyramid Networks were introduced by Lin et al. (2016) and based on the results of their experiments, they claimed that although the neural networks
Figure 5.4: Architecture of the U-net. It is composed by an encoder and a decoder that are symmetric (Ronneberger et al., 2015).

are invariant to scale changes, multi-scale problems should be addressed with pyramid network in order to get more accurate results.
5.2 Tree-structure information: Graph neural network

The main limitation of convolutional neural networks for the localization of the start/end points of the right coronary artery tree is that they give independent predictions for each point. However, these points are connected in the vessel structure and a change in one point should affect the rest. This subsection introduces a type of neural networks that can model tree-structured data.

The coronary arteries have a tree-shape structure which means that there is a hierarchy among the segments. Graph Neural Networks (GNN) (Scarselli, Gori, Tsoi, Hagenbuchner & Monfardini, 2009) have been employed to model the relationship among the different points of the right coronary tree.

The annotated structure of the right coronary tree can be seen as a graph in which each start/end point corresponds to a node and the arcs of the graph are the artery branches (shown in Figure 5.5). Except the first and last node, every node in the graph is connected to a parent and child node. In the case of the start point of the catheter, it is only connected to a child node which is the end point of the catheter and regarding the end point of the distal division, it only has a parent node which is the start point of the distal division.

Figure 5.5: Graph Neural Network. The image on the left is a sample from the dataset which has annotated the five points to identify. The schema on the right shows the graph that models the structure of the right coronary artery tree. Each point in the image is associated with a node in the graph and the branches of the coronary tree delimited by the annotated points are associated with the arcs of the graph.

Each node in the graph is associated with a hidden state. The hidden state is the probability distribution among the pixels of the image of containing the point associated to the node, i.e. heat-map. The hidden state is initialized using the output of the CNN aforementioned and is recurrently updated using the information from the neighbouring points (parent and child nodes). The GNN is intended to improve the results obtained with the CNN by taking into account the information from the connected points. Therefore, the heat-maps are refined.

The information from a node and its neighbouring nodes is combined using the Gated Recurrent Unit (GRU) (Y. Li, Tarlow, Brockschmidt & Zemel, 2015) to update its hidden state. Each GRU receives two inputs: the hidden state of the node, $h_k^t$, and the combination of the hidden state of
the neighbouring nodes, $x_k^t$, shown in equation Equation 5.1.

$$x_k^t = \sum_{k,k' \in \Omega} W_{p,k} h_{k'}^{t-1} + b_{p,k} \quad (5.1)$$

And the following are the operations that occur inside the GRU:

$$z_k^t = \sigma(W_{z,k} x_k^t + U_{z,k} h_k^{t-1} + b_{z,k}), \quad (5.2)$$

$$r_k^t = \sigma(W_{r,k} x_k^t + U_{r,k} h_k^{t-1} + b_{r,k}), \quad (5.3)$$

$$\tilde{h}_k^t = \tanh(W_{h,k} x_k^t + U_{h,k} (r_k^t \odot h_k^{t-1}) + b_{h,k}), \quad (5.4)$$

$$h_k^t = (1 - z_k^t) \odot h_k^{t-1} + z_k^t \odot \tilde{h}_k^t \quad (5.5)$$

$\Omega$ is the set of neighbouring nodes; $k$ represents a node and $t$ a concrete time step; $W_{p,k}, b_{p,k}, W_{z,k}, U_{z,k}, b_{z,k}, W_{r,k}, U_{r,k}, b_{r,k}, W_{h,k}, U_{h,k}, b_{h,k}$, are the parameters of the convolutions; $W, U$ are the weights of the kernel and $b$ are the bias.

Each GRU updates the hidden state of a node once and its parameters are shared among all the GRU of the same node. Finally, equation Equation 5.5 gives the new hidden state of the node. The number of recurrent updates is set to two based on the results of the experiments of (Zhang et al., 2019) and in order to avoid that nodes which are not directly connected have an influence in the prediction.

Figure 5.4 shows (a) the graph that has as many nodes as points and the hidden state of each node is updated using the hidden states from the connected nodes and (c) the two GRUs associated to a node that are used to gather the information from proximal nodes and to update the hidden state twice.

Since the points in the extremes are only affected by one neighbour, the predicted location may have more freedom. Thus, another GNN which draws a connection between the last and first point as a loop was also employed for this project. This connection may guide the model to learn which should be the relative distance between these points.

In order to initialize the hidden state of the nodes using the CNN, both networks were concatenated and trained at the same time. Consequently, although the loss function that both models tried to minimize was the same, the loss function of the whole model was a combination of both.
CHAPTER 5. METHODOLOGY

Figure 5.6: Gated Recurrent Unit. Schema (a) shows the graph that models the structure of the right coronary artery tree. Each node receives information from its parent and child node; the first node only gets information from the child node and the last node only gets information from the parent node. (c) Each node has as many Gated Recurrent Units (GRU) as number of times the hidden state is updated and they share the weights. This example shows how the hidden state of a node (Node 1) is updated twice. $h_0^1, h_1^1, h_2^1$ are the hidden states of the node at different time steps and $x_0^1, x_1^1$ are the combination of the hidden state of the parent and child nodes also at different steps.

5.3 Loss function

The proposed approach to solve the branching location of the right coronary tree is based on the Human Pose Estimation principles. The majority of proposals that address this task and use fully convolutional neural networks, minimize the Mean Squared Error (MSE) (Equation 5.6) pixel-wise between the output of the network and the ground-truth which is a heat-map with a Gaussian distribution centered in the point (Zhang et al., 2019; Lin et al., 2016; Z. Cao et al., 2016; Long et al., 2015; Tompson et al., 2014; Wei et al., 2016). The main limitation of this approach is that the model is minimizing a function that is not the target metric. The model is trying to give an output that is as close as possible to the ground-truth image and then it computes the argmax function to get the coordinates of the pixel which has the maximum likelihood of having the point. Since the argmax function is not differentiable, it cannot be integrated in the learning process. Therefore, the learning stops at image level and the gradient flow starts at the heat-map rather than at the coordinates of the point. The target metric for Human Pose and the current problem is to minimize the distance between the predicted point and the real one and a decrease in the MSE may not imply that the prediction has improved. Figure 5.7 shows this effect.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y_i})^2$$  \hspace{1cm} (5.6)

where $y_i$ is the output of the model, $\tilde{y_i}$ is the ground-truth and $n$ is the number of pixels in the image.

36 Right coronary artery labelling with point annotations
CHAPTER 5. METHODOLOGY

Figure 5.7: Limitations of Mean Squared Error (MSE) loss. Image (a) shows the ground-truth, and (b) and (c) are different predictions. The pixel-wise MSE between (a) and (b) and (a) and (c) will have the same value, although the prediction is closer in (b) than (c). Therefore, MSE is not the ideal function that the model should optimize.

In order to overcome the limitations of the MSE, our approach tries to minimize the Differentiable Spatial to Numerical Transform (DSNT) loss (Nibali, He, Morgan & Prendergast, 2018). It is a combination of two terms (see Equation 5.7). This loss function allows the model to directly learn the objective metric using a fully differentiable approach that can be included in the back-propagation process.

\[ L = L_{euc}(\mu, p) + \alpha L_D(\hat{Z}, p) \] (5.7)

Each of the heat-maps produced by the model describes a probability distribution of the point of interest among the pixels of the image. All the outputs are positive because the last layer of the network has a ReLU activation function and they are normalized in the loss function so their sum is 1.

The first term of the loss function is the Euclidean distance between the predicted coordinates of the point and the ground-truth (see Equation 5.8).

\[ L_{euc}(\mu, p) = \|\mu - p\|_2 \] (5.8)

\( \mu \) are the ground-truth coordinates and \( p \) are the coordinates predicted by the model.

The second term in Equation 5.7 is the Jensen-Shannon divergence between the heat-map and a Normal distribution centered in the predicted location and with a variance of 45 pixels (Equation 5.9). It is multiplied with \( \alpha \) which is a parameter that controls its importance for the final metric. This term ensures that the final heat-map has the shape of a 2D normal distribution with a peak in the point of interest. This way, the model is enforced to identify only one pixel that is the most likely to have the point and its neighbouring pixels. Consequently, if the model cannot find the exact point because, for example, if it is occluded, it will find the neighbors and it will approximate the position of the point. The loss for the five predicted points is computed and the final loss is a summary of the individual errors.

\[ L_D(\hat{Z}, p) = D_{JS}(p)\|N(p, \sigma^2)) \] (5.9)
CHAPTER 5. METHODOLOGY

Figure 5.8: Calculation of the coordinates for the loss function. The normalized heat-map produced by the neural network is used to get the average \(x\) and \(y\) pixel coordinates. The coordinates are normalized between -1 and 1 so the network is forced to learn heat-maps which are symmetric with respect to the point.

5.4 Generalization techniques

The generalization of the model is limited by the data available for training. Therefore, this subsection presents different techniques to increment the amount of information the model can receive so it can generalize to unseen cases with more precision.

5.4.1 Data augmentation

Data augmentation techniques were specifically designed to include the different possible variations the data can have which can help the model to become invariant to these variations.

In this study, image contrast variation was considered so the intensity of the images was randomly modified in each iteration as shown in Equation 6.1 with the parameter \(\gamma\) that ranges from 0.3 to 1:

\[
G(I, j) = \gamma F(I, j)
\]  

Additionally, as aforementioned, the geometry of the coronary arteries can differ significantly among different data samples. With the aim of increasing the range of geometries included in the training dataset, the coronary arteries were deformed elastically.

The elastic deformation employed is based on the approach of (Simard et al., 2003) with some modifications. Firstly, an affine deformation is applied to the image. It consists in taking a 2D triangle in the image and moving each of the vertices with a translation field created from a uniform distribution between \(-\beta_{\text{affine}}\) and \(\beta_{\text{affine}}\). Then, the affine transformation between the original triangle and the modified one is estimated and applied to the whole image.

The second step of the methodology is the non-linear transformation of the image. For the deformation, a displacement field is created by sampling random numbers from a uniform distribution lying between -1 and 1. A Gaussian filter with standard deviation \(\sigma\) is applied to the deformation field and then, it is multiplied by \(\beta\) that controls the strength of the deformation. Finally, the image is transformed using the deformation field. For the experiments \(\sigma\) is set to 40, \(\beta_{\text{affine}}\) and \(\beta\) change randomly from iteration to iteration ranging from 0 to 104 and from 0 to 46 respectively. The same transformation is applied to the annotations so the ground-truth labels keep correspondence with the images.
The images were deformed in two different ways. One procedure consists of the transformation of the full image which implies that the borders created by the shutter are also non-linearly deformed and they may not keep forming straight lines. In an attempt to control the noise that the deformation of borders may cause, the other approach applies the transformation only to the vessel structure. Since the angiography sequence also contains the images when the vascular structure is not highlighted, the subtraction between the image with the vessels and the background image gives only the coronary arteries. Therefore, the transformation is applied to the difference image and the result is added to the background image.

5.4.2 Semi-supervised

Semi-supervised learning (SSL) is one of the most common procedures for addressing the lack of annotated data. SSL algorithms can improve the performance of the model by also using the data that does not have labels. The unlabeled data provides extra information about the structure of the data that may improve the results of the model for the task it was designed to (Chapelle, Schölkopf & Zien, 2010).

SSL is incorporated to the VGG16-FCN to improve the way the encoder represents the data. The decoder of the neural network was duplicated in order to perform another task which was the reconstruction of the images and the accuracy of the results was measured with the MSE. First, the encoder of the neural network was connected to the reconstruction decoder and was trained. Afterwards, the layers of the encoder were frozen so they were not trained and it was connected to the decoder that locates the points of the right coronary artery. Therefore, the encoder learned a common representation of the data for the two tasks.

5.4.3 Propagation of annotations

As an alternative approach to solve the problem of the shortage of data, more images from the same angiography sequence can be included in the training. There is only one image annotated in each sequence so the other images can be labelled by propagating the annotations. Under the assumption that images form the same sequence show the same vascular structure but deformed and moved, it is possible to find a transformation that applied to one image will result in the other one. This alignment process between two images is called registration and consists of mapping equivalent pixels from the image that has the annotations and the one without. Then, the mapping transformation is applied to the ground-truth data and the other images from the sequence are also annotated.

The registration was done using fully convolutional neural networks that output two deformation fields, one for the X axis and another for the Y axis of the image. The input of the neural network are two images, the reference that does not have the annotations and the moving that has the annotations. The model tries to find a deformation field that applied to the moving image will result in the reference one (H. Li & Fan, 2017). The model starts with two blocks that have 3x3 convolution operation, followed by batch normalization, ReLU activation function and max pooling. These two blocks are followed by 4 blocks of 3x3 convolution, batch normalization and ReLU. The output at 128x128, 256x256 and 512x512 pixels go through a 1x1 convolution layer.

The outputs of the model are deformation fields at these three scales. The moving image is deformed using this field and the values of intensity are obtained using bilinear interpolation. Afterwards, the loss is computed using the MSE between the image deformed and the reference. Besides the MSE, there is another term in the loss function that controls the smoothness of the transformation. This term is computed using the gradient of the deformation field.

5.4.4 Evaluation

The evaluation of the performance is measured using the Euclidean Distance (Equation 5.8) between each of the predicted points and the ground-truth when the models attempt to localize the points in the images for testing.

Right coronary artery labelling with point annotations
CHAPTER 5. METHODOLOGY

In order to assess the performance of the approach proposed in this project, although no other study has tried to identify the different start/end points of the divisions of the right coronary tree, the results have been compared with the work of Noothout et al. (2018). Noothout et al. (2018) developed an algorithm for the location of a set of cardiac landmarks and one of the points is in the left coronary tree.

Noothout et al. (2018) developed their work under the assumption that pixels which are close to the one that has the point of interest will have more relevant information for the final prediction than the rest. Their proposal consists of a fully convolutional neural network composed by three blocks that have a 3x3 convolution, followed by batch normalization, ELU activation function and max pooling. These blocks are followed by two blocks that have a convolution layer with a 3x3 filter bank followed by batch normalization and ELU activation. Finally, the model has two outputs that go through a 1x1 convolutional layer followed by batch normalization, ELU activation function and another 1x1 convolution. One output consists of two filter maps obtained after applying a sigmoid function to the output of the previous layer and the other one is a single map obtained directly from the previous convolution.

For the labels, the images are divided in patches of 8x8 pixels so they get the same scale as the output from the neural network. There are two types of labels in correspondence with the outputs from the model (64x64 pixels). The output from the neural network with two feature maps corresponds to the \(x\) and \(y\) coordinates of vectors centered in the patches and pointing towards the interest landmark as can be seen in Figure 5.9. The other type of annotations are images that have 0 or 1 value, if a 8x8 patch contains the landmark, its value will be 1 and the rest will have value 0 (shown in Figure 5.9). This proposal is based on the assumption that patches with a high probability of having the point of interest will have crucial information for the final prediction so only the vectors from these patches are averaged to get the result.

Figure 5.9: Labels. The image on top is a sample from the dataset with the end point of the catheter highlighted. The images on the bottom are the labels: the two images on the left are the coordinates of the vectors that centered in each 8x8 patch of the image point towards the landmark position and the image on the right shows the classification label in which only the patch that contains the point has a value of 1.
Chapter 6

Results and discussion

This chapter reports results from the different convolutional neural networks, the graph neural network and goes on to show how the different generalization techniques, loss functions and variations in hyper-parameters can affect the performance.

The section 6.1 section describes the different experiments performed and it also shows the results of the experiments that were not satisfactory so they were not included in the main proposal. The section 6.2 evaluates the results of the experiments that constitute the proposed approach and provides a discussion about the different findings.

6.1 Experiments

The first experiment consists of the registration between images that have labels and images from the same sequences that do not have labels in order to have more data for training. The second experiment compares the performance of the models with the two different elastic deformation techniques for data augmentation purposes: deformation of the whole image or only the vessel structure. The third experiment compares the DSNT loss (chapter 5) with the MSE to verify the assumption that the performance of the model is more precise when it is directly learning the metric to minimize. After this trial, all the experiments are run with the DSNT loss. The fourth experiment addresses the comparison among different neural network architectures described chapter 5. In the fifth experiment, the two Graph Neural Networks detailed in chapter 5 are combined with the fully convolutional neural networks to see if the results are refined by using neighbouring information. The purpose of the sixth experiment was to analyse if the inclusion of images without labels may help the model to represent more accurately the data, hence giving better predictions. Finally, the last experiment compares the performance of our approach to another solution designed for cardiac landmark location.

In all the trials, the optimization technique used was Adam with a learning rate of 0.0001. In addition, in all the models but the Graph Neural Network and the seventh experiment, the batch size was set to one image. All the experiments but the semi-supervised and alternative approach were trained during 1000 epochs.

6.1.1 Experiment 1: Deep learning registration

The images to register are frames from the same sequence of acquisition. Each image with annotations was paired with one image captured five frames in advance and with another one captured five frames after. This choice of time windows was done to get images that differ from the original one but they still have the vessels highlighted. Not all angiography sequences have at least ten frames so only 210 pairs of images were used by the model. Figure 6.1 shows some results.

The accuracy of the registration was measured using the MSE and the Structural Similarity Index (SSIM). Results show that there are samples that differ significantly, though, they are from Right coronary artery labelling with point annotations
CHAPTER 6. RESULTS AND DISCUSSION

Figure 6.1: Three examples of registration results. The images from left to right are the reference, moving and the transformed image using the deformation field produced by the neural network. The pair of images was compared before and after the transformation using the Mean Squared Error (MSE) and Structural Similarity Index (SSIM).

The neural network’s objective is to learn a deformation field that defines a transformation from the same patient and acquisition sequence and the model could not learn a deformation that maps them. Figure 6.2 shows the difference in MSE and SSIM between the pair of images before and after the transformation of the moving image (Equation 6.1, Equation 6.2). Whereas the SSIM increases in the majority of samples after the transformation which determines that the structure of the moving image after the transformation is more similar to the reference, the MSE increases and it implies that the pair of images differ more after the transformation. Overall, these results indicate that the transformation slightly increases the similarity between the images to register or it worsens it. In consequence, the propagation of annotations using the same deformation field could not be done correctly and they were not included in the training dataset.

\[
\Delta \text{MSE} = \text{MSE}_{\text{after transformation}} - \text{MSE}_{\text{before transformation}} \quad (6.1)
\]

\[
\Delta \text{SSIM} = \text{SSIM}_{\text{after transformation}} - \text{SSIM}_{\text{before transformation}} \quad (6.2)
\]

The neural network’s objective is to learn a deformation field that defines a transformation from...
CHAPTER 6. RESULTS AND DISCUSSION

Figure 6.2: Comparison of the similarity between the images to map before and after the transformation using deep learning. The $x$ axis represents the differences in the metrics after the registration and before and the $y$ axis is the number of images. The graph on top shows the results compared using Mean Squared Error (MSE) and the lower graph shows the results with Structural Similarity Index (SSI).

6.1.2 Experiment 2: Elastic deformation

The two different ways of deforming elastically the images presented in chapter 5 for data augmentation purposes were compared using the VGG16-FCN model. The main limitation of the whole deformation of the image, as aforementioned, is that the borders in the image produced by the collimator are deformed in a non-affine way which can be seen in Figure 6.3 (b). On the other hand, the deformation of the subtracted image can be also a source of errors because in some background images the catheter appears highlighted and it also appears in the image with the vessels but in a different position due to the movements of the heart. Consequently, the addition operation may result in images with two catheters which can be seen in Figure 6.3 (c). The results of this experiments will be shown in the next section.
CHAPTER 6. RESULTS AND DISCUSSION

Figure 6.3: Elastic deformation. (a) Image from the dataset which is (b) fully deformed that causes the borders to lose their square shape and (c) only the difference between image (a) and the background image is deformed and the result is added to the background image. Although the image (c) shows a defined catheter, it shows another one less defined caused by the presence of the catheter in the background image.

6.1.3 Experiment 3: Loss function

To verify the advantages of the DSNT loss function in comparison to the traditional MSE introduced in chapter 5, the VGG16-FCN model was trained trying to minimize these two error functions. The Euclidean Distance between the prediction and the ground-truth was measured at the end of each iteration with the data for testing for the five points.

The alpha value of the DSNT loss equation (Equation 5.7) was set to 100. This value was experimentally chosen to make the divergence term (Equation 5.9) of the loss significant for the whole loss value and the variance of the Normal distribution was set to 45 pixels.

Figure 6.4 shows that when the model is minimizing the MSE, it does not always mean that the prediction is getting closer to the point of interest, for example in Point 3 and 4 (end point of the proximal and mid division), the prediction is not getting closer to the real point location. The reason of this behaviour is that the model is optimizing its parameters based on a function that does not measure the final goal which is the accurate location of a point.

Table 6.1 shows the results obtained with the same model (VGG16-FCN) but different loss functions. Although the start and end point of the catheter can be located by the model trained with MSE with lower average error, the model trained with DSNT can identify with more precision the cardiac points. Therefore, the results support the notion that the performance of the network is better when it is directly learning the objective metric to minimize.

Table 6.1: Average Euclidean Distance error and standard deviation (in pixels) in the prediction for the test data. Results obtained when the model was trained using the Mean Square Error (MSE) loss and the Differentiable Spatial to Number Transform (DSNT) loss.

<table>
<thead>
<tr>
<th></th>
<th>Start catheter</th>
<th>End catheter</th>
<th>End prox. div.</th>
<th>End mid div.</th>
<th>End dist. div.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNT</td>
<td>19.70±20.13</td>
<td>20.31±15.33</td>
<td>40.15±21.33</td>
<td>49.18±30.22</td>
<td>53.36±37.51</td>
</tr>
<tr>
<td>MSE</td>
<td>12.33±9.19</td>
<td>17.58±7.56</td>
<td>51.35±36.46</td>
<td>60.70±42.47</td>
<td>62.63±56.22</td>
</tr>
</tbody>
</table>

6.1.4 Experiment 4: Neural network

This subsection compares the results obtained with different neural networks. Firstly, an experiment with only one bifurcation point was run to investigate which fully convolutional architecture
CHAPTER 6. RESULTS AND DISCUSSION

Figure 6.4: Error with test data computed at the end of each training epoch. At the end of each epoch, the weights of the model were used with the test data and two different error measures were computed: Mean Squared Error (MSE) and Euclidean distance in pixels. The graph on top shows the MSE which is the loss function the model is trying to reduce; it is the average among all the five predictions. The bottom part of the figure shows the Euclidean distance error computed for each one of the points separately. Point 1 is the start point of the catheter; Point 2 is the end point of the catheter; Point 3 is the end point of the proximal division; Point 4 is the end point of the mid division and Point 5 is the end point of the distal division.

The two architectures to compare are VGG16-FCN and VGG16-FPN (see chapter 5). Table 6.2 presents the average error and the standard deviation in pixels for the location of the end of the catheter. From this data, it can be seen that VGG16-FCN which does not use predictions at different scales achieves better results.

Table 6.2: Average Euclidean Distance error and standard deviation (in pixels) in the prediction for the test data. Results obtained with two models: VGG16-FCN and VGG16-FPN.

<table>
<thead>
<tr>
<th></th>
<th>End catheter</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16FCN</td>
<td>17.92±12.52</td>
</tr>
<tr>
<td>VGG16FPN</td>
<td>28.60±22.62</td>
</tr>
</tbody>
</table>

of the VGG16 is more suitable for this problem. The point to localize was the end point of the catheter because it is the one that has less variation among the different images compared to the other start/end points of divisions so it was assumed to be the easiest one to identify.

The two architectures to compare are VGG16-FCN and VGG16-FPN (see chapter 5). Table 6.2 presents the average error and the standard deviation in pixels for the location of the end of the catheter. From this data, it can be seen that VGG16-FCN which does not use predictions at different scales achieves better results.

Table 6.2: Average Euclidean Distance error and standard deviation (in pixels) in the prediction for the test data. Results obtained with two models: VGG16-FCN and VGG16-FPN.
One interesting finding is that the VGG16 based architecture that uses predictions at different scales does not outperform the precision of the VGG16 which only produces one output at the highest scale. The Feature Pyramid Networks have been used to detect objects by the prediction of the coordinates of bounding boxes and to predict image masks (Lin et al., 2016). Contrary to the approach of (Lin et al., 2016) which attempts to detect objects of different sizes so each level of the model is responsible of the location of a concrete size objects, the VGG16-FCN used in this project detects the same object at all the scales. The images from the dataset have different zoom level so the main benefit of the FPN was to locate points at arteries that have different sizes but the objects to locate are points that have a constant size, thereby their detection cannot be divided into different levels so all the scales of the model must find the points.

When comparing our results to Zhang et al. (2019) which used FPN to locate the human joints, although all the levels of the model must find the same point, they do it at the scale of the input data. It is possible because Zhang et al. (2019) used the Stack Hourglass architecture (Newell, Yang & Deng, 2016) that has intermediate outputs the same size as the input data. Since the VGG16-FPN proposed in this project could not find accurately the point in lower dimensions, this may be the reason why the precision in higher dimensions does not improve.

The second comparison was done between VGG16-FCN and U-net and the results are shown in the next section.

6.1.5 Experiment 5: Graph neural network

In order to investigate the effectiveness of the GNN, this architecture was concatenated with the VGG16-FCN. This experiment analyses if the results of the CNN are refined when the predictions go through the GNN. The outputs of the VGG16-FCN are the input for the GNN, the loss function is a combination from the loss of the VGG16-FCN output and the one from the GNN and both models are trained at the same time with a batch size of two. The GNN needs a higher batch size otherwise the behavior is unstable and the model cannot converge to a local minimum error. It can be explained by the fact that connected points contribute to the final prediction of each other so if in one image the way they are combined is unusual, this combination will be applied to the following image which can lead to big errors. By a batch size of two, the average of both gradients is used to update the parameters of the neural networks so it does not allow drastic changes. Two different GNN were employed as shown in Figure 6.5. While Figure 6.5 (a) shows a GNN in which each node represents a point and receives information from the points to which it is connected, Figure 6.5 (b) draws an extra link between the first and last node creating a loop although they are not connected in the coronary artery tree. This extra connection forces their predictions to be related and the model may learn relative properties. Therefore, both types of graph representations were compared and results are shown in the following section.

Figure 6.5: Graph Neural Network models. (a) Simple graph that only draws arcs among points that are connected in the right coronary tree; (b) Graph model that adds an extra connection between the last point (end point of the distal division) and the root (start point of the catheter).
Additionally, the effect of the variance $\sigma$ of the Normal distribution centered in the point of interest was evaluated with the VGG16-FCN in combination with the simple graph. This probability distribution among the pixels, i.e. heat-map, is the ground-truth for the divergence term of the loss function (Equation 5.9). Figure 6.6 shows the evolution of the loss function of the model when the variance of the Normal distribution has a value of 45 pixels and also 25. From the results, it can be seen that the model learns faster with the variance of 45. In addition, the model can generalize better to unseen cases and the training process shows less oscillations so it can convergence faster to the minimum error. It can be explained by the fact that a higher $\sigma$ implies that the distribution is more spread out and the number of samples in which it is likely to identify the point of interest is higher. Therefore, during training there are more cases in which the gradient is not zero and the network can learn faster.

Figure 6.6: Evolution of the loss value during training and validation using the VGG16-FCN when the Normal distribution around the point of interest that the model must learn have a variance of 45 or 25 pixels.
6.1.6 Experiment 6: Semi-supervised

The purpose of this experiment was to include data that is not labeled during the training so the model may locate the points more accurately. Firstly, the VGG16-FCN was trained in an unsupervised way during 50 epochs. The goal of the model was to reconstruct the images so the loss function was the MSE between the reconstruction and the original data. Secondly, the first two convolutional blocks of the VGG16-FCN were frozen, thereby their parameters did not change and the model was trained with the annotated data to identify the points for 2000 iterations. 4096 images were employed for the unsupervised execution and 111 were used during the supervised learning. The difference between the performance of the model when it was trained previously for the unsupervised task and when it was directly trained for the supervised task are shown in the next section.

6.1.7 Experiment 7: Comparison

The aim of this experiment is to compare our proposal with an alternative approach which was also intended for cardiac points although they did not work with the right coronary tree (Noothout et al., 2018). The algorithm designed by Noothout et al. (2018) and described in subsection 5.4.4 has been applied to the right coronary artery dataset in order to identify each of the points separately. There are two differences with respect to the experiments of Noothout et al. (2018). On one hand, the number of iterations used during training in this project was limited to 10000 instead of 60000, hence the computational time is similar to the other approaches to compare. On the other hand, the threshold of patches to include for the final prediction calculation is set to 0. Table 6.3 shows that for the testing data, the value of the patch that has the maximum likelihood of containing the landmark differs considerably among the test samples. Thus, it was not possible to establish a threshold suitable for all the data. The results of the comparison are shown in the next section.

Table 6.3: Minimum and maximum value of the patch that has the highest likelihood of containing the point of interest among the test samples.

<table>
<thead>
<tr>
<th></th>
<th>Start catheter</th>
<th>End catheter</th>
<th>End prox. div</th>
<th>End mid. div</th>
<th>End dist. div</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Min</strong></td>
<td>8.940697e-08</td>
<td>5.9604645e-08</td>
<td>0.0</td>
<td>1.4901161e-07</td>
<td>7.778406e-06</td>
</tr>
<tr>
<td><strong>Max</strong></td>
<td>0.9999976</td>
<td>0.99631697</td>
<td>0.999831</td>
<td>0.9994475</td>
<td>0.9994594</td>
</tr>
</tbody>
</table>

6.2 Results and discussion

In this section, the goal is to discuss and evaluate the results obtained in the experiments and make a comparison with other approaches.

Table 6.4 shows a summary of all the experiments performed an also two extra experiments that consist of the combination of the techniques that showed the best performance: VGG16-FCN with the simple graph and the data augmented using the elastic deformation of the subtraction and the same one but combined with unsupervised learning.

When comparing the results obtained between the different ways of performing elastic deformation that are the deformation of the whole image or the deformation of the subtraction between the image with the vessels highlighted and the background image; the performance of the two techniques is similar. The results can be seen in Table 6.4: VGG16-FCN with the whole image deformation and the deformation of the subtracted image which only contains the vessel structure and the same with U-net. Interestingly, while the precision in the identification of the points located in the coronary artery structure is similar or slightly better in the transformation of the subtraction, the prediction of the start point of the catheter is less accurate. As aforementioned, this alternative way of deforming the image can cause the duplication of the catheter in some images. Thus, it could be the reason of the poorer performance of the model in this point.
Table 6.4: Summary of the average Euclidean Distance error and standard deviation (in pixels) in the prediction for the test data.

<table>
<thead>
<tr>
<th></th>
<th>Start catheter</th>
<th>End catheter</th>
<th>End prox. div</th>
<th>End mid. div.</th>
<th>End dist. div.</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16-FCN (Whole deformation)</td>
<td>19.70±20.13</td>
<td>20.31±15.33</td>
<td>40.15±21.33</td>
<td>49.18±30.22</td>
<td>53.36±37.51</td>
</tr>
<tr>
<td>+ Simple graph</td>
<td>17.21±21.01</td>
<td>20.40±16.48</td>
<td>36.00±20.62</td>
<td>51.53±30.82</td>
<td>61.33±48.71</td>
</tr>
<tr>
<td>+ Loopy graph</td>
<td>16.81±17.77</td>
<td>19.07±15.00</td>
<td>38.76±19.93</td>
<td>42.04±18.08</td>
<td>46.73±32.49</td>
</tr>
<tr>
<td>+ Semi-supervised</td>
<td>25.31±20.69</td>
<td>21.06±19.15</td>
<td>36.21±20.34</td>
<td>45.49±30.38</td>
<td>52.49±48.41</td>
</tr>
<tr>
<td>Vessel deformation</td>
<td>23.35±26.84</td>
<td>16.99±12.89</td>
<td>35.19±19.47</td>
<td>49.90±31.26</td>
<td>49.36±40.47</td>
</tr>
<tr>
<td>U-net (Whole deformation)</td>
<td>14.42±13.70</td>
<td>25.63±20.38</td>
<td>34.16±19.66</td>
<td>51.25±27.05</td>
<td>48.48±39.41</td>
</tr>
<tr>
<td>+ Simple graph</td>
<td>14.75±15.46</td>
<td>23.35±22.10</td>
<td>41.00±25.71</td>
<td>58.34±32.56</td>
<td>51.82±46.19</td>
</tr>
<tr>
<td>+ Loopy Graph</td>
<td>16.87±16.73</td>
<td>25.78±18.54</td>
<td>43.58±21.95</td>
<td>53.44±30.03</td>
<td>52.26±41.87</td>
</tr>
<tr>
<td>Vessel deformation</td>
<td>18.88±24.65</td>
<td>19.99±18.39</td>
<td>35.19±19.47</td>
<td>49.90±31.26</td>
<td>49.36±40.47</td>
</tr>
<tr>
<td>Alternative approach</td>
<td>71.16±63.36</td>
<td>89.85±67.50</td>
<td>126.52±91.95</td>
<td>110.88±71.67</td>
<td>75.82±38.73</td>
</tr>
<tr>
<td>(Noothout et al., 2018)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Combined (VGG16-FCN)</td>
<td>18.61±20.02</td>
<td>19.10±12.10</td>
<td>37.36±22.11</td>
<td>43.74±25.76</td>
<td>44.95±41.95</td>
</tr>
<tr>
<td>(+ Simple graph)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Vessel deformation)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ Semi-supervised</td>
<td>23.67±30.75</td>
<td>20.14±15.88</td>
<td>35.79±19.60</td>
<td>39.58±21.21</td>
<td>40.58±35.22</td>
</tr>
</tbody>
</table>

Regarding the neural network architecture experiments, it remains unclear which architecture is better to perform regression of the start/end points of the right coronary tree divisions. It can be seen in Table 6.4: VGG16-FCN and U-net, that there are three points that U-net can identify with lower average error and the other two are identified more precisely by the VGG16-FCN. Overall, the average error is lower with the U-net but both architectures perform equally well.

VGG16-FCN and U-net have an encoder and decoder structure but the layers and parameters show differences. While VGG16-FCN downscales the size of the input data from 512x512 pixels to feature maps of 16x16, U-net only reduces the scale to 32x32 because it has one convolutional block less in the encoder. The encoder of VGG16 is more expressive which can mean that it can learn better representations of the data but the authors of Huang et al. (2017) claim that by the concatenation of feature maps from different levels of the model, it can learn the same representation as less compact models. The results obtained in the comparison between the two models confirmed the findings of Huang et al. (2017) because both models reach a similar performance which is slightly superior in the case of U-net.

Besides the differences in the number of layers between U-net and VGG16, the main distinction between the models is the way the features from low layers are combined with the ones from high layers in order to get the final prediction. Whereas VGG16 performs an addition of the feature maps, U-net concatenates them. The main difference between the two merge operations is that the concatenation can learn which features have more importance and it also means more parameters to train. However, Kayalibay et al. (2017) demonstrated that the difference in the combination affects the way the information flows. In the concatenation of features, there are two separate streams of information which are the local and global information from the images that are combined in the last layer of the model and in the summation, there is a single stream of information that is refined.

The results Graph Neural Network are shown in Table 6.4: VGG16-FCN and U-net combined with the simple graph and the loopy graph. GNN performs a refinement of the results by means of the addition of information from the neighbouring points to the prediction.

The results of the experiments found clear support for the assumption that the inclusion
of hierarchical information to the model, improves the results of VGG16-FCN. Figure 6.7 and Figure 6.8 show that the accuracy of the base model improves when it is concatenated with the simple graph model. Concretely, this difference is higher in the third and fourth point, the end points of the proximal and mid division respectively. In some samples from the test set, VGG16-FCN mismatched the prediction of these two points. However, when the results were processed using the simple graph model, the predictions of the points were not mixed and the model always kept a separation among the points. Figure 6.9 presents some images from the test set with the points highlighted, the points in green are the ground-truth annotations and the points in red are the predictions. Whereas the results from VGG16-FCN indicate that the fourth point is predicted as the third, when the predictions are refined with the simple graph, the mismatching does not occur. In addition, Figure 6.10 presents examples in which the last points from the coronary structure, the end points of the mid and distal division, are predicted together or really close but the graph model forces the predictions to be separated.

![Box and whiskers plot of the error between the prediction and the ground-truth in pixels for the start and end point of the catheter. While the line inside the boxes represents the median of the values, the cross is the mean value. Each of the boxplot represents the error obtained with a different model: VGG16-FCN, the previous one combined with a simple graph neural network and the combined with a loopy graph model.](image)

However, the results of the U-net show a different situation in which the prediction does not improve. It is difficult to explain the reasons of this behaviour but it may be related with the stream of information. In the GNN, the information of the connected points is added as a correction to
the whole representation so when the model is doing back-propagation, the error of the prediction of a point affects the prediction of this point obtained with the VGG16-FCN or U-net but also the prediction of its neighbouring points. The VGG16-FCN with the addition of feature maps is performing a refinement in the prediction at every layer, thereby the back-propagation of the error got at the GNN will update the parameters of every layer of the VGG16-FCN in order to improve the refinements. On the other hand, in the case of the U-net, there is one gradient flow from the output of the GNN and this gradient will follow the local features stream in order to update these maps for a better prediction but also the global features one for the same purpose. Both kind of features need to be updated in order to locate more precisely a point but also to improve the location of points that are connected to it. This combination of features is more complex for the model to learn and it could be the reason why GNN only improves the results of the VGG16-FCN.

The comparison between the two types of graph presented in Figure 6.5 proved that the simple tree-structured graph reaches better performance than the loopy graph. These results do not match with Zhang et al. (2019) because the authors got better results with the loopy graph. It can be attributable to the fact that their problem was to locate the joints of the human body and one point could have more than one child point but the points of the right to identify from coronary tree have a less complex structure which can be represented by a simpler graph.

Turning now to the experiment of the model that combined unsupervised and supervised learning, the results demonstrate that the information added by the unlabeled data improves the prediction of those points the model found more difficult to locate. The results show that the location of the end point of the mid division and the end point of the distal division, which are the most challenging to identify, are the ones whose prediction improves the most after adding information from unsupervised data.

Contrary to previous studies that have trained the network for the unsupervised and supervised task at the same time for efficiency reasons (Rasmus, Valpola, Honkala, Berglund & Raiko, 2015), in this work the model was pre-trained for the reconstruction of unlabeled data and it was followed by the regression task. Rasmus et al. (2015) developed an approach that is computationally efficient but at expenses of poorer hidden representation of the data because it was trained for a classification task which did not need rich encodings. However, the regression of points in the coronary arteries is a task that needs higher precision.

Finally, when the approach presented in this project was compared to another approach Noothout et al. (2018) for cardiac landmark location, the proposed methodology showed better performance. The are several possible explanations for the difference in accuracy between both approaches. Firstly, Noothout et al. (2018) employed a simple model with only 9 convolutional layers so it may not be able to learn complex representations of the data. It could also influence that the number of iterations during training was set to 10000 instead of 60000 as in the original work. Additionally, the model of Noothout et al. (2018) learns to identify each point independently so it does not take into consideration the information from other points and it was not intended for tree-structured data so it does not model the relationships among points. The results of Noothout et al. (2018) applied to the coronary artery dataset show that the main limitation was the second output of the model that tries to identify the patch of the image that contains the point of interest. In addition, their work is based on the assumption that pixels close to the region of interest have crucial information for the final result. However, in the case of coronary arteries, the points to identify can have local similarity. This similarity is more evident when the points of interest are not the start of a bifurcation which can happen in some patient anatomies so they could be similar to any other point located in the vessel structure.

To conclude, considering that the first point to identify is not included in the vessel structure, the approach that identified the points in the right coronary tree with lower average error is the semi-supervised learning with VGG16-FCN combined with the simple graph and the data augmented using the elastic deformation of the subtraction.
Figure 6.8: Box and whiskers plot of the error between the prediction and the ground-truth in pixels for three points located in the right coronary tree. While the line inside the boxes represents the median of the values, the cross is the mean value. Each of the boxplot represents the error obtained with a different model: VGG16-FCN, the previous one combined with a simple graph neural network and the combined with a loopy graph model.
Figure 6.9: Predictions in the test set. The points highlighted are the correct annotations in green and the predictions in red. Each column shows the predictions inferred with a different model.
Figure 6.10: Predictions in the test set. The points in green are the ground-truth annotations and the points in red are the predictions. Each column shows the predictions inferred with a different model. The image on top left shows that the fourth and five points are predicted together.
Chapter 7

Conclusions and future lines

The purpose of the current study was to develop a model that could identify the start and end point of the different divisions of the right coronary artery tree. The branches can be defined as the segments between the start and end point and they can be labelled automatically. This study has compared the performance of different methodologies that have been used with medical data in order to apply their methodology for the current problem. In addition, approaches employed for human pose estimation have also been considered due to its similarity to this problem because in both cases the data has a tree structure and there is high variability among samples.

The findings from this study make several contributions to the current literature. First, the coronary artery tree labelling problem has been addressed using point annotations which are faster to make than annotations based on the segmentation of the different branches. The approach that has shown the best performance can find the points of interest with an average error that ranges from 20 to 40 pixels in images with a resolution of 512x512 which means 4-8% of image size error. Second, the evidence from this study suggests that the incorporation of hierarchical representation for problems related with tree-structured data, improves the generalization of the models. This information was included using the Graph Neural Network that was designed to work with the human skeleton in order to find the body joints (Zhang et al., 2019) and it was adapted to the right coronary artery structure. This additional information reduces the average error of the predicted points and the improvement ranges from 0.2% to 1.64% of image size. Moreover, the accuracy of the results is improved the most in the end points of the mid and distal division which are the most challenging to locate because they have more variation among the samples. Third, the experiments with semi-supervised learning confirmed that learning hidden representations for different tasks can improve the performance of the model for unseen cases. In accordance with the present results, previous studies have demonstrated that sharing feature representations of the data among different tasks might improve the generalization of the model (Rasmus et al., 2015). By the addition of semi-supervised learning to the model combined with the GNN, the error is decreased in a higher extend that ranges from 0.2% to 0.8% of image size. Finally, although the current study is based on a small sample of images from the right coronary tree, the findings suggest that by means of data augmentation techniques that modify the images in different ways, the samples can represent more variability of the data.

The start and end point of the catheter can be identified by all the models with more precision. The position of these points is slightly constant in all the images and it does not depend on the anatomy. The point that has the highest error is the last one and it is also the one that has more variability. In order to get better results, it would be interesting to include in the experiment more images that represent better the anatomical variance among the different patients.

To conclude, the proposed method performed accurate prediction of the start and end point of the catheter in the right coronary tree in angiographies. However, since medical products need to be really precise, the average errors obtained in the prediction of the end points of the proximal, mid and distal division are not suitable for a real-world product. It would be necessary to work with more data and from different places in order to get more accurate results. Future research
should be devoted to the replication of this approach with an extended dataset and once the results are good enough, it would be interesting to apply the same approach for the left coronary tree.
References


	doi: 10.1007/s10278-017-9983-4

	doi: 10.1016/j.media.2018.05.010

	doi: 10.1016/S0003-2778(10)80023-1


	doi: 10.1016/j.pcad.2016.04.003

	doi: 10.5603/FM.2015.0057


	doi: 10.1016/S1361-8415(01)00047-0


References


coronary artery centerline extraction in computed tomography angiography data. In (p. 856 - 859). doi: 10.1109/ISBI.2007.356987


Percutaneous coronary intervention. (n.d.). Retrieved July 05, 2019, from https://atgprod.heart.org/HEARTORG/Encyclopedia/Heart-Encyclopedia_UCM_445084_Encyclopedia.jsp?title=percutaneous%20coronary%20intervention

Radiographic contrast. (2019).


