Analysis and Segmentation of Coronary Arteries using Novel Deep Learning Techniques

Ilektra-Despoina Papamatthaiaki

School of Electrical and Computer Engineering

Thesis Committee:
Professor Michail Zervakis (Supervisor)
Professor Thrasyvoulos Spyropoulos
Professor Stavroulakis Georgios
(School of Production Engineering and Management)

Chania, July 2025

Acknowledgments

Having spent all these years at the Technical University of Crete, I would like to express my appreciation for all the opportunities and resources it has given me, which helped me complete this thesis.

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Michail Zervakis, for his invaluable guidance and continuous support throughout this journey. I would also like to thank Dr. Marios Antonakakis for his thoughtful insights and helpful suggestions, which played an important role in the progress of this work.

Moreover, I am especially thankful to my friends, whose constant assistance, encouragement, and belief in me helped me persevere, especially during challenging times. I would also like to extend my thanks to the thesis committee, Prof. Thrasyvoulos Spyropoulos, Prof. Georgios Stavroulakis, and the new friends and colleagues I met at the DISPLAY Lab, who offered their help and contributed to the development of this thesis.

Above all, I owe my tremendous gratitude to my family. Their emotional and practical support helped me pursue and complete my studies. I sincerely thank my brother for sharing his knowledge of heart anatomy, which greatly enhanced my comprehension of the medical component of this project. Finally, my parents assisted me through the years, and for that, I will always be thankful.

Abstract

Coronary Artery Disease (CAD) is among the principal causes of death globally. Since current diagnostic methods are primarily invasive, there is an increasing interest in accurate, non-invasive alternatives. In this regard, deep learning developments have proven pivotal in estimating diagnostic indices. 3D modeling of coronary arteries in an accurate manner is crucial for enabling dependable, non-invasive diagnostic processes. This thesis explores the use of deep learning for coronary arteries segmentation in computed tomography angiography (CTA) images, with a view towards facilitating early diagnosis and improving treatments. Two models are compared and evaluated: Basic U-Net, a convolutional neural network (CNN), and a transformer-based model, UNETR. Both are implemented within the same framework so that a direct and fair comparison is ensured.

Considering the increasing interest in transformer architectures within the medical field, this comparison intends to assess whether they hold tangible potential for improving the research in medical image segmentation. Although experimental results indicated that both models had high accuracy, Basic U-Net performed consistently better than UNETR, especially when there were constraints of limited data and computational resources. A quantitative evaluation using the Dice Similarity Coefficient (DSC) revealed an average score for Basic U-Net of 90.14%, in comparison to 89.56% for UNETR. Although theoretically, UNETR has an advantage in capturing dependencies over greater distances, its performance was likely constrained by its higher data requirements and sensitivity to computational limitations.

These findings indicate that convolutional architectures remain more reliable under low-resource conditions and highlight the importance of model selection and dataset size when it comes to medical applications in this field. Overall, the research affirms that deep learning is feasible for coronary arteries segmentation and further supports its potential application for non-invasive diagnostic processes for CAD.

Περίληψη

Η στεφανιαία νόσος (CAD) συγκαταλέγεται μεταξύ των κύριων αιτιών θανάτου παγκοσμίως. Δεδομένου ότι οι υφιστάμενες διαγνωστικές μέθοδοι είναι κυρίως επεμβατικές, παρατηρείται αυξανόμενο ενδιαφέρον για την ανάπτυξη ακριβών, μη επεμβατικών εναλλακτικών λύσεων. Στο πλαίσιο αυτό, οι εξελίξεις στη βαθιά μάθηση έχουν αποδειχθεί καθοριστικές για την εκτίμηση διαγνωστικών δεικτών. Η ακριβής τρισδιάστατη μοντελοποίηση των στεφανιαίων αρτηριών είναι ζωτικής σημασίας για την υποστήριξη αξιόπιστων, μη επεμβατικών διαγνωστικών διαδικασιών. Η παρούσα διπλωματική εργασία διερευνά τη χρήση τεχνικών βαθιάς μάθησης για την τμηματοποίηση των στεφανιαίων αρτηριών σε εικόνες αξονικής στεφανιογραφίας (CTA), με στόχο τη διευκόλυνση της έγκαιρης διάγνωσης και τη βελτίωση των θεραπευτικών παρεμβάσεων. Συγκρίνονται και αξιολογούνται δύο μοντέλα: το Basic U-Net, ένα συνελικτικό νευρωνικό δίκτυο (CNN), και το UNETR, ένα μοντέλο βασισμένο σε αρχιτεκτονική τύπου transformer. Και τα δύο μοντέλα υλοποιούνται στο ίδιο υπολογιστικό πλαίσιο, ώστε να διασφαλιστεί άμεση και δίκαιη σύγκριση.

Λαμβάνοντας υπόψη το αυξανόμενο ενδιαφέρον για τις αρχιτεκτονικές transformer στον ιατρικό τομέα, η παρούσα σύγκριση αποσκοπεί στην αξιολόγηση του κατά πόσο αυτές οι τεχνικές παρουσιάζουν ουσιαστική προοπτική για την πρόοδο της έρευνας στην τμηματοποίηση ιατρικών εικόνων. Αν και τα πειραματικά αποτελέσματα έδειξαν υψηλή ακρίβεια και για τα δύο μοντέλα, το Basic U-Net παρουσίασε σταθερά καλύτερη απόδοση από το UNETR, ιδίως σε συνθήκες περιορισμένων δεδομένων και υπολογιστικών πόρων. Η ποσοτική αξιολόγηση με τον Συντελεστή Ομοιότητας Dice (DSC) κατέδειξε μέση τιμή 90,14% για το Basic U-Net, έναντι 89,56% για το UNETR. Παρόλο που θεωρητικά το UNETR υπερέχει στην αποτύπωση εξαρτήσεων μεγάλου εύρους, η απόδοσή του φαίνεται να περιορίστηκε από την υψηλή απαίτηση σε δεδομένα και την ευαισθησία του σε υπολογιστικούς περιορισμούς.

Τα ευρήματα αυτά καταδεικνύουν ότι οι συνελικτικές αρχιτεκτονικές παραμένουν πιο αξιόπιστες σε περιβάλλοντα με περιορισμένους πόρους και υπογραμμίζουν τη σημασία της επιλογής κατάλληλου μοντέλου και του μεγέθους του συνόλου δεδομένων στις ιατρικές εφαρμογές αυτού του πεδίου. Συνολικά, η παρούσα έρευνα επιβεβαιώνει ότι η βαθιά μάθηση είναι εφαρμόσιμη για την τμηματοποίηση των στεφανιαίων αρτηριών και ενισχύει περαιτέρω την προοπτική αξιοποίησής της σε μη επεμβατικές διαγνωστικές διαδικασίες για τη στεφανιαία νόσο.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Subject and Motivation of the Thesis

Coronary Artery Disease (CAD) is a very important health problem that can lead to death of many people. It is responsible for roughly 610,000 people losing their lives each year in the United States, estimated for one in four deaths, making it the principal cause of mortality in the country. Globally, it ranks as the third leading cause of death, contributing to an estimated 17.8 million fatalities annually [3]. When someone has CAD, blood flow through the coronary arteries is restricted. These arteries are responsible for delivering oxygenated blood to the heart muscle, and this narrowing is primarily caused by the buildup of plaque, which is composed of cholesterol and other substances, along the arterial walls, resulting in progressive constriction.

Despite its risks, CAD is preventable if diagnosed early. For more accurate diagnosis, a method called Invasive Coronary Angiography (ICA) is used, which, as the name suggests, is an invasive imaging procedure. However, when it does not clearly indicate a severe occlusion, pressure measurements may be taken to assess the physiological significance of a narrowing. This is known as Fractional Flow Reserve (FFR), which is calculated as the ratio of the distal to proximal coronary pressure across the lesion [20]. FFR quantifies the functional severity of coronary artery narrowing and assists clinicians in determining the need for revascularization.

There are two main problems: first, ICA carries certain health risks due to its invasive nature, and secondly, it is an expensive technique. Additionally, it is known that most people who are going through invasive techniques to diagnose it do not actually need it [25]. Hence, a non-invasive method called Coronary Computed Tomography Angiography (CCTA) is used in order to eliminate the necessity for further detailed evaluation by ICA and FFR. Although it has lower resolution, CCTA provides 3D imaging of the coronary arteries, allowing radiologists to visually assess the potential stenosis [31]. In many cases, it is sufficient to exclude patients from invasive procedures, reducing unnecessary ICA referrals. However, a significant number of unnecessary referrals still occur, highlighting the need for more specific non-invasive diagnostic tools.

A major requirement for the development of non-invasive diagnostic tools is the accurate modeling of CA. Over the past few years, DL-based techniques have been promising in the area of medical image segmentation, including attempts to segment coronary arteries from CCTA scans. Some of these have already utilized convolutional neural networks (CNNs), specifically U-Net variants, owing to their good performance in learning high-resolution thin structures with fine anatomical details. In this thesis, a novel transformer-based DL model, UNETR, is utilized and compared with the established state-of-the-art CNN-based model, Basic U-Net. The aim is to evaluate whether transformer architectures have the potential for tangible gains in accuracy or generalization, especially for segmentation tasks that have highly complex vascular structures like the coronary arteries.

1.2 Contribution of the Thesis

This thesis presents a comparative evaluation of two deep learning models for the task of coronary artery segmentation from CCTA images. The first is UNETR, a transformer-based architecture, and the second is Basic U-Net, a convolutional baseline model.

First, it presents a rigorous benchmarking of transformer-based segmentation by evaluating the performance of UNETR against the Basic U-Net, on the dataset of CCTA scans. This comparison offers valuable insights into the respective strengths and limitations of transformer architectures when applied to the segmentation of small and highly vascular structures such as the coronary arteries.

In addition to this architectural comparison, the study includes both quantitative and qualitative assessments. Quantitative evaluation is carried out using the metric Dice Similarity Coefficient (DSC), while qualitative visual inspection is employed to assess the anatomical plausibility of the predicted segmentation under real clinical imaging conditions. This dual approach ensures a comprehensive understanding of model performance beyond raw numerical scores. Furthermore, the thesis examines the generalization capability of the two models across different patient samples, emphasizing the robustness of each architecture to anatomical variability and variations in image quality. This aspect is particularly important for clinical deployment, where reliability across diverse cases is critical.

Finally, by focusing on the non-invasive segmentation of coronary arteries, this work contributes to the broader effort of reducing reliance on invasive diagnostic techniques. Improved automatic segmentation methods have the potential to lower healthcare costs and enhance patient safety by streamlining diagnostic pathways.

As a whole, this thesis builds upon previous research on coronary artery segmentation through the provision of one of the few direct comparisons of CNN-based and transformer-based models for this task. These findings provide valuable insights that can guide the development of future deep learning-based, non-invasive diagnostic support systems for CAD.

1.3 Outline of the Thesis

The thesis is divided into five chapters, where each chapter discusses a main point related to the research of coronary artery segmentation based on deep learning methods.

Chapter 2 introduces the theory that is behind the methodology and findings of this research. It starts with the medical background to understand the anatomy and function of the coronary arteries, along with the clinical picture of coronary artery disease. The chapter continues to discuss the basics of deep learning with special emphasis on how deep learning is utilized for image classification and segmentation problems. Certain important concepts like activation functions, evaluation measures, and the methods of optimization in the case of

computer vision are laid down. Special emphasis is laid upon CNNs while also on transformer-based networks like the UNETR model, laying the basis for the models utilized later on.

Chapter 3 describes the data processing and model development methodology that was employed. Here, the tools and frameworks utilized across the study are introduced, namely MONAI, PyTorch, ITK-Snap, and 3D Slicer. The data pre-processing workflow is described in a detailed manner, followed by the architecture of the segmentation models. Training and validation strategies are also described in this chapter, discussing the paramount computational constraints faced in the course of the experiment.

Chapter 4 introduces the results and discussion. It is a comparison of the segmentation outputs generated by the Basic UNet and UNETR models. Model performances are quantitatively and qualitatively evaluated across different cases. A discussion of the related prior research is also added to put the findings in perspective.

Chapter 5 summarizes the primary findings and contributions of the thesis. It describes the limitations of the study and outlines the directions for future research to enhance segmentation accuracy and clinical utility.

2 Theoretical Background

This section gives the theoretical foundation that is required for the comprehension of the rest of the thesis, ranging from the medical overview of coronary arteries to the basics of deep learning.

2.1 Medical overview

2.1.1 Anatomy and Function of Coronary Arteries

The coronary arteries run along the coronary sulcus of the heart muscle, also known as the myocardium. The myocardium, like other tissues in the body, needs oxygen-rich blood to function [17]. Although their primary role is to deliver blood to the heart, they also facilitate the removal of oxygen-depleted blood. There are two main coronary arteries, the left main coronary artery (LMCA) and the right coronary artery (RCA). Both of these arise from the root of the aorta. The anatomical positioning of the arteries varies between individuals. The coronary arteries encircle the external surface of the heart, with smaller branches penetrating the myocardium to ensure adequate perfusion.

2.1.2 Coronary Artery Disease

CAD is a prevalent form of heart disease that affects the coronary arteries. In CAD, blood flow to the heart muscle is reduced due to the buildup of fats, cholesterol, and other substances in and on the walls of the arteries. This buildup, known as plaque, leads to a condition called atherosclerosis, which causes the arteries to narrow. This condition develops gradually over the years, with symptoms that include chest pain (angina), shortness of breath, and fatigue. If left untreated, a complete blockage of blood flow can result in a heart attack, heart failure, or abnormal heart rhythms.

2.1.3 Diagnosis

Accurate diagnosis is critical for optimizing treatment strategies, improving the prognosis, and reducing the burden of the disease. The diagnostic process begins with a thorough clinical evaluation, which includes a detailed history and physical examination to identify symptoms, risk factors [16]. Following clinical evaluation, various diagnostic methods can be employed, which are broadly classified into invasive and non-invasive techniques. The most commonly used diagnostic approaches include:

Invasive Diagnostic Techniques

1. Invasive Coronary Angiography (ICA)

A diagnostic procedure where a catheter is inserted through an artery to visualize the coronary arteries. This invasive technique highlights blockages or

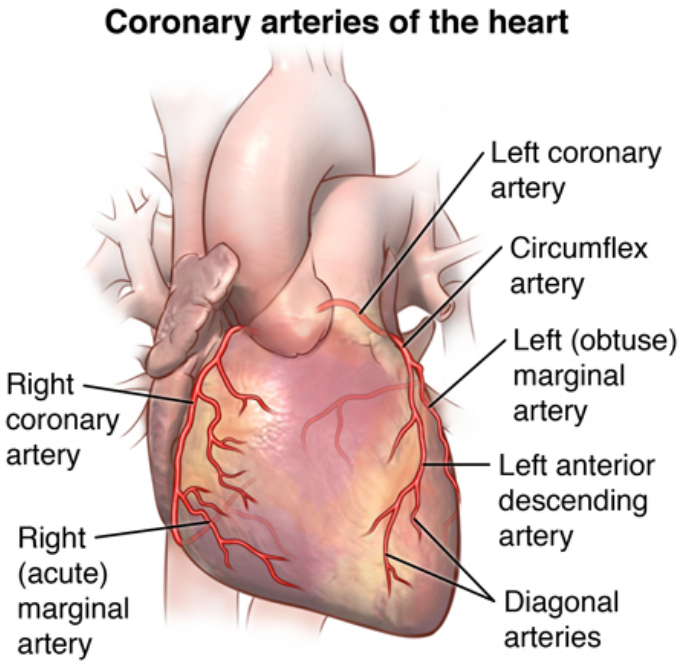


Figure 1: Anatomy of the heart's coronary arteries.

Source: [link]

abnormalities within the coronary vessels. Despite its accuracy, ICA comes with several risks, including bleeding, infection, allergic reactions, and radiation exposure. Additionally, the procedure is expensive and time-consuming, making it less ideal for routine screening. It is indicated in high-risk patients, those with severe symptoms unresponsive to medical therapy, or when non-invasive tests are inconclusive.

2. Fractional Flow Reserve (FFR)

A technique used to measure the pressure difference across a coronary stenosis to determine its functional significance. FFR is typically performed during ICA using a pressure-sensitive guide wire. By quantifying the blood flow reduction caused by the narrowing, FFR helps assess whether a stenosis is likely to cause ischemia, providing crucial functional information beyond what imaging alone can offer.

The diagnosis of CAD involves a multimodal approach combining clinical evaluation, non-invasive testing, and invasive techniques when necessary. Risk

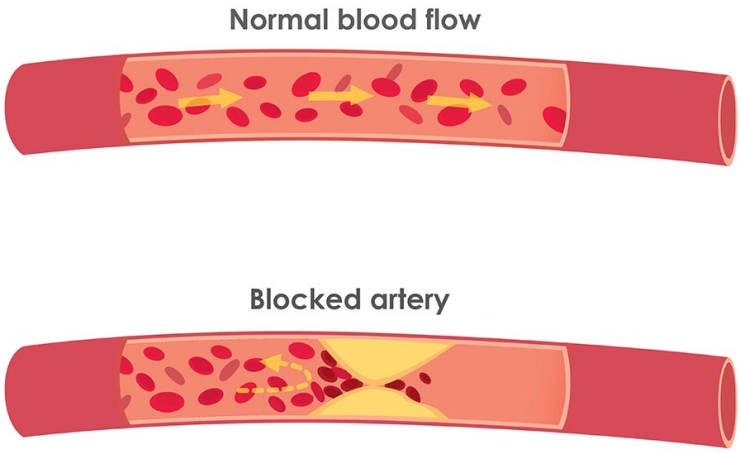


Figure 2: Illustration of plaque buildup.

Source: [link]

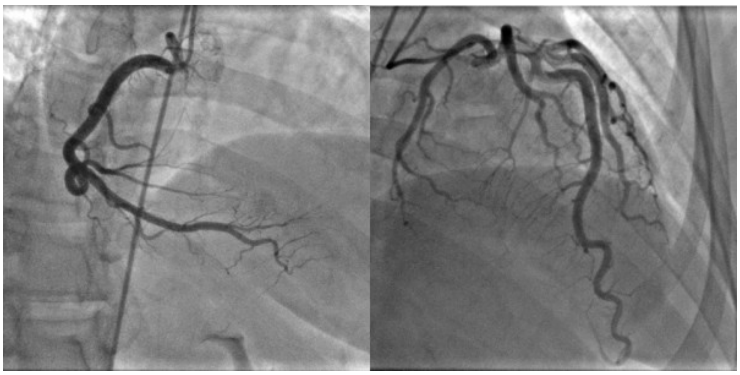


Figure 3: ICA revealing right coronary dominance with no disease in the epicardial coronary arteries.

Source: [23]

stratification frameworks guide the use of diagnostic tools and treatment pathways, ensuring personalized and evidence-based care for patients with suspected CAD.

Non-Invasive Diagnostic Tools

1. Resting Electrocardiogram (ECG)

A resting 12-lead ECG is a fundamental tool in the initial evaluation of CAD. It can reveal signs of myocardial infarction or ischemia, such as ST-segment changes. While a normal ECG cannot rule out CAD, it provides a valuable baseline for comparison in future examinations and aids in ongoing assessment.

2. Stress Testing

Stress testing assesses myocardial perfusion and cardiac function during physical exertion or pharmacological stress. It includes exercise ECG, which detects ischemic changes in patients with intermediate pre-test probability, and stress imaging techniques like echocardiography, nuclear imaging (SPECT or PET), and cardiac MRI (CMR), which offer greater sensitivity and specificity, particularly in those with baseline ECG abnormalities.

3. Coronary Computed Tomography Angiography (CCTA)

A non-invasive imaging technique that uses CT scans with intravenous contrast to create detailed 3D images of the coronary artery inner part (lumen) and anatomy in detail, enabling the detection of severe stenosis. It is especially useful for patients with intermediate pre-test probability, offering a high negative predictive value to exclude obstructive CAD. Although generally safe, coronary CTA requires specific conditions, including sufficient breath-holding ability, sinus rhythm, and a heart rate of 65 beats per minute or lower.

2.2 Deep Learning

Deep Learning (DL) is a subfield of Artificial Intelligence (AI) that employs neural networks to derive insights, recognize patterns, and make predictions from both raw and processed data. Its rapid adoption across various industries has revolutionized applications in fields such as healthcare, image recognition, natural language processing (NLP), cybersecurity, and more.

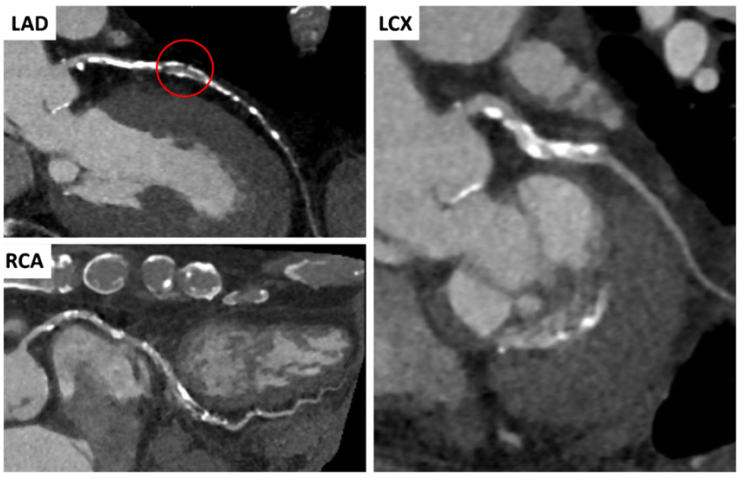


Figure 4: CCTA images in a symptomatic CAD patient showing extensive calcified plaque, complete proximal LAD stent occlusion (red circle), and severe RCA and LCx narrowing.

Source: [28]

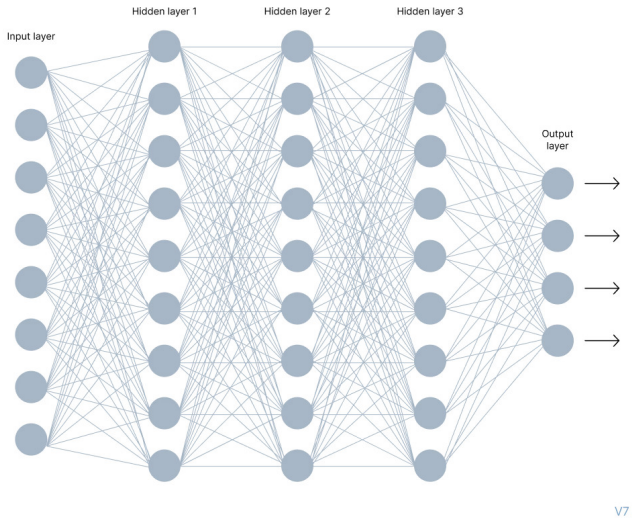


Figure 5: Schematic representation of a fully connected artificial neural network.

Source: [link]

At its core, DL is built on the concept of artificial neurons, computational units inspired by the structure and function of biological neurons. These neu-

rons are organized into layers, with each layer serving a specific purpose in the learning process. The input layer is responsible for receiving the initial data, which is then passed to one or more hidden layers. The output of these hidden layers ultimately leads to a prediction in the output layer. The whole network's architecture is portrayed in Figure 5.

To dig deeper, each neuron in the hidden layers computes a weighted sum of the inputs it receives from the previous layer's neurons. This summation function is represented mathematically as:

$$z = w_1X_1 + w_2X_2 + \cdots + w_nX_n + b$$

Where:

- $X_1, X_2, \ldots, X_n$ represent the inputs,
- $w_1, w_2, \ldots, w_n$ are the corresponding weights,
- b is the bias term.

The computed sum, z, is then passed through an activation function f to introduce non-linearity, which allows the model to capture complex patterns in the data. The final output y is then produced:

$$y = f(z)$$

The weights associated with these connections are critical, as they determine the influence of each input on the neuron's output. These weights are adjusted during the training process through optimization algorithms such as back-propagation, which minimize the error between the predicted and true values. This continual adjustment of weights helps the model improve its performance over time.

2.3 Computer Vision

Computer Vision (CV) is a popular multidisciplinary branch of Computer Science that focuses on empowering machines to interpret, analyze, and comprehend visual data (such as images or videos) from the world around us. The ultimate goal is to make it possible for computers to see and understand the objects, people, and activities in an image or video in the same way that humans do. Medical imaging, autonomous vehicles, facial recognition, object detection, and even intelligent video monitoring for emergency help and security are just a few of its uses [2].

2.3.1 Image Classification

Classification is a fundamental pattern recognition task in the field of AI and data analysis that involves learning from a dataset of labeled images to predict the label (class) of unseen images. To develop a classification model, a training dataset is required, consisting of images and their corresponding labels, which

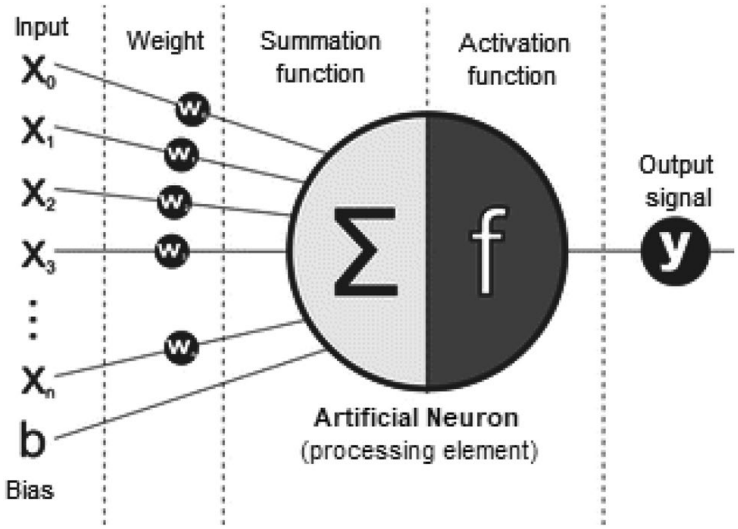


Figure 6: Schematic of an artificial neuron with weighted inputs, bias, and activation function.

Source: [26]

serve as target outputs. For example, a very common problem is that if an image classification model is trained to recognize cats, it should correctly label an image of a cat when presented with images of both cats and dogs.

2.3.2 Semantic Segmentation

Semantic segmentation is based on image classification, but rather than labeling the entire image, it assigns a class to each pixel, allowing for a more accurate and localized understanding of the visual data. As can be seen in Figure 5, the image shows a cluster of pixels that are members of the 'cow' class instead of two different 'cow' instances. In other words, a key feature of semantic segmentation is that it does not distinguish between different objects belonging to the same class.

Since it can deal with 3D volumetric data, semantic segmentation is highly helpful in medical imaging. As illustrated in Figure 6, it allows for automatic and accurate organ identification by classifying each voxel rather than each pixel.

2.3.3 Activation Functions

Activation functions, also referred to as transfer functions, are essential elements of neural networks, since they decide whether or not a neuron will be activated. For a neuron to transfer its value to the next layer, its output must be greater than the activation threshold. They also introduce non-linearity and 'break

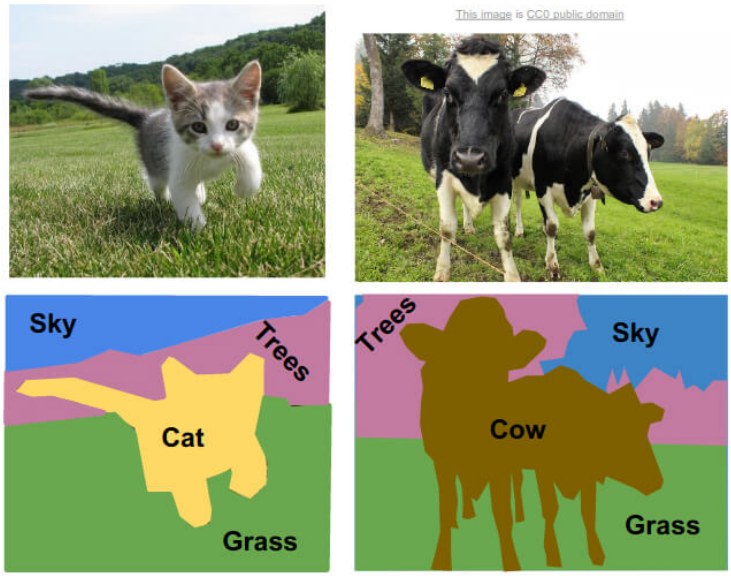


Figure 7: Labeling without instance differentiation.

Source: [link]

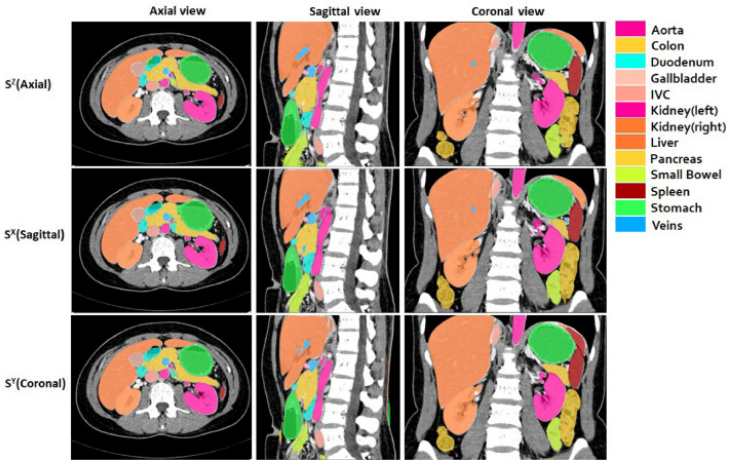


Figure 8: Semantic segmentation in multiple organs.

Source: [30]

away' from typical linear operations such as matrix multiplication. In real-world problems, data are almost always non-linear. For example, a patient's medical history, symptoms, and disease course cannot easily be modeled linearly, and this is just one example of why activation functions are essential for training

and predicting complicated data. Below is a summary of the most popular activation functions.

Linear Activation Function

The linear activation function, also known as the identity function, returns the input (x) as the output. It is defined as:

$$f(x) = x$$

Graphically, as shown in Figure 9, it looks like a straight line with a slope of 1. Essentially, if this function is applied, all the layers will be added together, making the last layer a linear function of the first one, thereby reducing the network to a single layer.

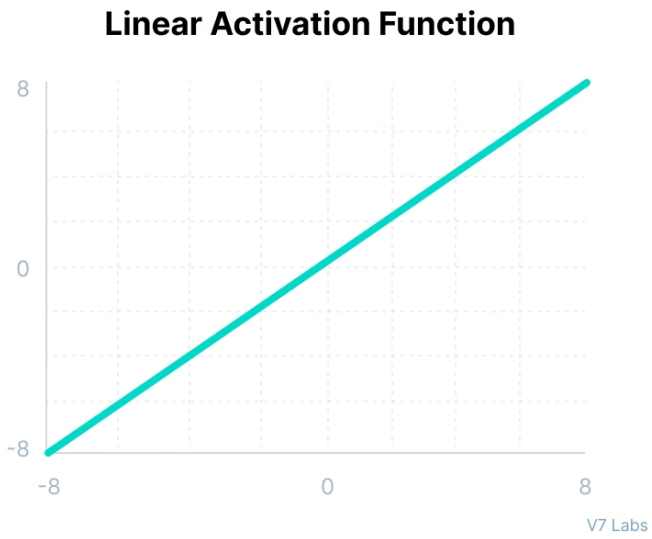


Figure 9: Linear (Identity) activation function.

Source: [link]

Binary Step Activation Function

Binary step function determines whether a neuron is 'fired' or not based on a threshold value, as shown in Figure 10. The input is compared with this value, and if it exceeds it, the neuron is activated; if not, the output is not transmitted to the subsequent hidden layer. Mathematically, it is defined as:

$$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ 1, & \text{for } x \geq 0 \end{cases}$$

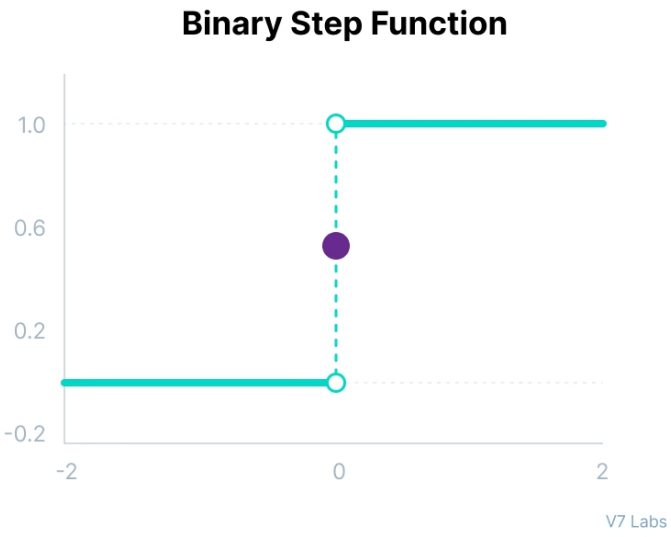


Figure 10: Binary step activation function.

Source: [link]

Sigmoid Activation Function

Sigmoid activation function, also called logistic function, as can be distinguished in Figure 11, is smooth and continuously differentiable. It accepts any real value as input and returns values between 0 and 1. The output value will be nearer 1.0 if the input is larger (more positive), and closer to 0.0 if the input is smaller (more negative). The mathematical form of the sigmoid activation function is as follows:

$$f(x) = \frac{1}{1 + e^{-x}}$$

When you utilize the derivative of this function, it is clear that its gradients will be quite small for values larger than 3 or smaller than -3. As the gradient value gets closer to zero, only slight changes occur in the weights during back-propagation, which makes the learning process extremely slow. This is known as the Vanishing Gradient problem in neural networks.

ReLU (Rectified Linear Unit) Activation Function

ReLU aims to minimize the issue of the vanishing gradient. Although it may appear to be a linear function, as shown in Figure 12, its derivative makes it both

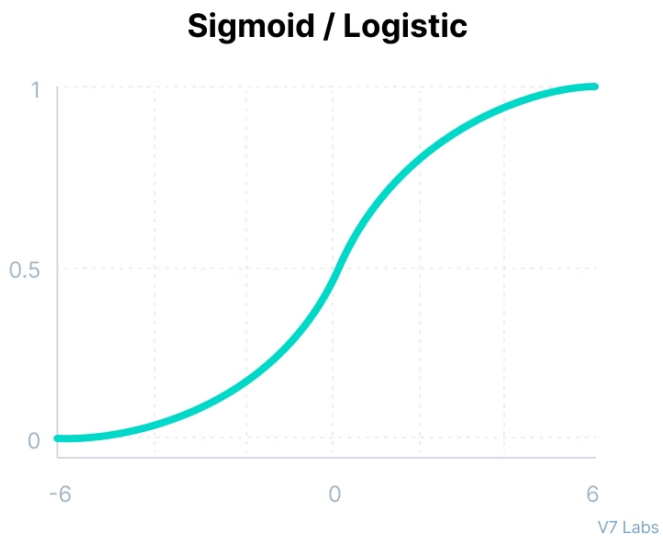


Figure 11: Sigmoid (Logistic) activation function.

Source: [link]

computationally efficient and effective for back-propagation. This is because it does not activate all neurons at once and deactivates them when the output of the linear transformation is less than zero. It is defined as:

$$f(x) = \max(0, x)$$

The issue that arises from using this function is the Dying ReLU Problem. On the negative side of the graph, the gradient value is zero. As a result, the weights and biases of some neurons are not updated during the back propagation process. This can cause neurons to become inactive, with their values remaining zero, which reduces the model's ability to train effectively on the data.

Leaky ReLU Activation Function

Finally, Leaky ReLU was introduced to address the Dying ReLU issue. Leaky ReLU has the same advantages as ReLU, plus back propagation for negative input values. With this small change for negative input values, the gradient on the left side of the graph will not be zero. Here is the mathematical expression, followed by Graph 13 for added clarity.

$$f(x) = \max(0.1x, x)$$

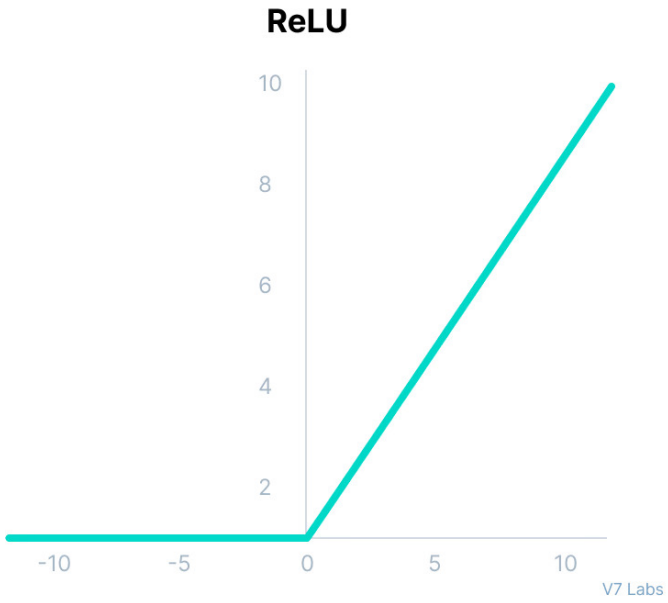


Figure 12: ReLU activation function.

Source: [link]

2.4 Optimization in Computer Vision

2.4.1 Evaluation Metrics

Dice Similarity Coefficient

The Dice Similarity Coefficient (DSC), commonly referred to also as the Dice-Sørensen coefficient, quantifies the similarity between two sets, A and B. It ranges from 0 to 1, where one indicates that the two sets are identical, and zero indicates that the two sets have no overlap [6]. It is frequently used in medical image segmentation due to its robustness to class imbalances and because it is more sensitive to small structures, as it directly measures overlap rather than counting true negatives. However, it does not take into account the spatial distance between segmented regions. It is defined as:

$$\text{DSC} = \frac{2|A \cap B|}{|A| + |B|} \tag{1}$$

Where,

- A is the predicted segmentation mask.
- B is the ground truth mask.

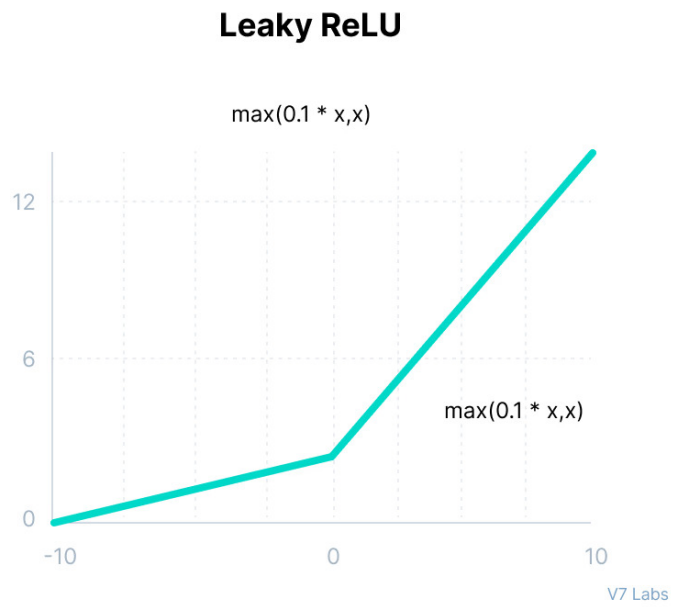


Figure 13: Leaky ReLU activation function.

Source: [link]

2.4.2 Loss Functions

A loss function is one of the most common criteria used to assess the effectiveness of statistical models, including semantic segmentation. Loss functions play an important role in shaping deep learning segmentation algorithms and help improve their overall performance. Some of the more well-known ones will be examined in this section [1].

Cross-Entropy Loss

Cross-entropy (CE) takes a random variable and measures the difference between two of its probability distributions. In segmentation tasks, it evaluates how closely the model's predictions match the target labels. By using a sigmoid function (suited for binary segmentation), the model develops pixel-by-pixel probability maps that show the likelihood of each of them falling into the targeted foreground class. Hence, the binary CE loss is calculated as follows:

$$L_{BCE} = -\sum_{i=1}^{N} [g_i \log p_i + (1 - g_i) \log(1 - p_i)]$$

Where,

- g_i represents the ground truth probability
- p_i is the predicted one for the foreground class.

Focal Loss

Focal loss can be considered a variation of BCE loss and was designed to address highly imbalanced class scenarios [14]. It focuses more on pixels that are hard to classify and reduces the weights of examples that are considered easy. Those easy pixels are the ones categorized with high confidence, whereas the hard ones are predicted by the model with low confidence. To achieve this, a modulating factor that down-weights easy samples is used, and the parameter $\gamma \geq 0$ controls how much the loss concentrates on hard samples. When $\gamma = 0$, focal loss simplifies to binary cross-entropy loss. The function is defined as:

$$L_{\text{Focal}} = -\sum_{i=1}^{N} \alpha_i \left[g_i (1 - p_i)^{\gamma} \log p_i + (1 - g_i) p_i^{\gamma} \log(1 - p_i) \right]$$

Where,

- $g_i \in \{0, 1\}$ is the ground truth label for pixel i.
- p_i is the predicted probability of the pixel being foreground (1) after applying sigmoid activation.
- α_i is an optional weighting factor to further adjust class imbalance.
- γ is the focusing parameter that reduces the weight of easy samples.

Dice Loss

Dice Loss is inspired by the Dice Similarity Coefficient (DSC), a metric that is widely used in medical image segmentation since it can handle class imbalance. It measures the overlap between predicted and ground truth masks, where the foreground, which is the object of interest, is significantly smaller than the background. However, it is non-differentiable and unsuitable for direct optimization in deep learning models since it operates in binary masks only. Mathematically, it is defined as:

$$L_{\text{Dice}} = 1 - \text{DSC} \tag{2}$$

Soft Dice Loss

Soft Dice Loss is a variant of Dice Loss that is differentiable and enables gradient-based optimization. Instead of using hard binary values, it uses predicted probabilities, ensuring smooth gradients for back-propagation. This is done by applying a sigmoid activation to the predicted values and then using them to calculate the DSC. A small smoothing factor ε is also added to prevent division by zero.

It is widely used in deep learning-based segmentation models as it directly optimizes for overlap, making it more effective in scenarios where foreground regions are sparse. It is defined as:

$$L_{SoftDice} = 1 - \frac{2\sum_{i=1}^{N} p_i g_i + \epsilon}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} g_i + \epsilon} \tag{3}$$

Where,

- p_i is the predicted probability of pixel i being in the foreground.
- g_i is the ground truth (0 for background, 1 for foreground).
- N is the total number of pixels.
- ϵ is the small smoothing factor.

2.5 Convolutional Neural Networks (CNNs)

CNNs are among the most advanced types of ANN architectures that exist today. They are primarily used for image recognition pattern tasks, and possess an easy-to-use design that makes implementing them much easier compared to other types of ANN architectures. CNNs surpass other techniques in image classification, object detection, and segmentation due to their powerful spatial data hierarchy detection. They consist of three types of layers: convolutional layers, pooling layers, and fully connected layers [18]. A simplified CNN architecture for classification is shown in Figure 14, where the following layers are:

- The input layer that holds the pixel values of the image.
- The convolutional layer that determines the output of neurons connected to local parts of the input by computing the scalar product between the weights of the neurons and the corresponding region in the input volume. An element-wise activation function, such as ReLU, is applied to the activation output from the preceding layer.
- The pooling layer that reduces the number of parameters in that activation by performing down-sampling along the input's spatial dimensionality.
- The fully connected layer that aims to produce class scores from the activations, to be used for classification. ReLU may be used between these layers to improve performance.

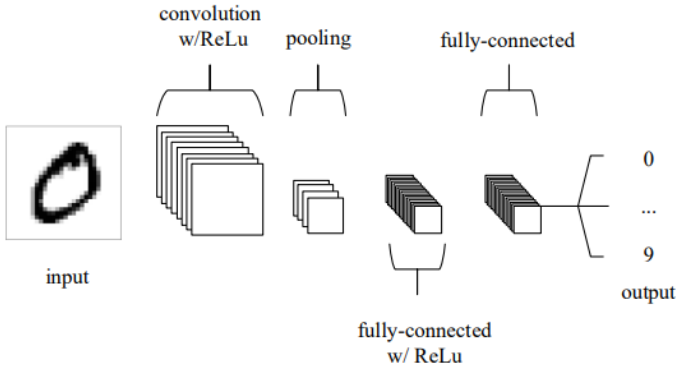


Figure 14: A simple CNN architecture, comprised of five layers.

Source: [18]

2.5.1 U-Net

U-Net is a CNN architecture that was designed specifically for medical image segmentation. Unlike traditional CNNs, it is designed to carry out dense pixel-wise segmentation. It is based on a fully convolutional network (FCN) design, and it can analyze images of different sizes, while it lacks fully connected layers [24].

The network architecture is illustrated in Figure 15. It is made of two parts, a contracting path (on the left side), which is the encoder, and an expansive path (on the right side), which is the decoder. The contracting path adheres to a convolutional network's standard architecture. Two 3x3 convolutions (which are unpadded convolutions) are applied repeatedly, each followed by a ReLU and a 2x2 max pooling operation with a stride of 2 for down-sampling. At every down-sampling stage, the number of feature channels is doubled. In the expansive part, each step involves up-sampling of the feature map, followed by a 2x2 convolution ("up-convolution") that reduces the number of feature channels by half. This is then concatenated with the appropriately cropped feature map from the contracting path. Afterward, two 3x3 convolutions are applied, each followed by a ReLU activation. Cropping is required because border pixels are lost during each convolution. In the final layer, a 1x1 convolution maps each 64-component feature vector to the required number of classes. In total, the network consists of 23 convolutional layers.

In Figure 15, note that the blue boxes represent a multi-channel feature map. The number of channels is displayed at the top, while the x-y size can be found at the lower left corner. The white boxes correspond to copied feature maps, and the arrows show the different operations that happen each time.

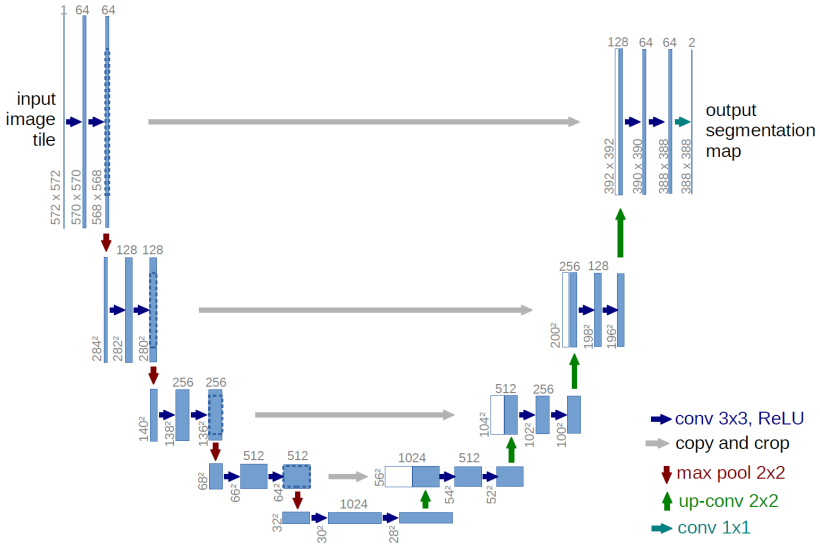


Figure 15: U-Net architecture.

Source: [24]

2.5.2 3D U-Net

The 3D U-Net model is an extension of the original U-Net architecture, designed to handle the complexities of volumetric data, such as CT and MRI scans. To achieve this, it replaces all 2D operations with their 3D equivalents, allowing for direct processing of volumetric data. In particular, 3D convolutions, 3D max pooling, and 3D up-convolutional layers are used. One important advantage is that it can learn from limited annotations, which is especially valuable in medical imaging, where fully labeled 3D datasets are rare and costly to create [35].

The network architecture is shown in Figure 16. Following the standard U-Net, it has an analysis (encoder) and a synthesis (decoder) path, each with four resolution steps. In the analysis path, every layer performs two 3×3×3 convolutions followed by ReLU activation, then applies 2×2×2 max pooling with a stride of two in all dimensions. In the synthesis path, each layer contains a 2×2×2 up-convolution by strides of two in each dimension, and then it is followed by two 3×3×3 convolutions and by a ReLU activation. The final layer is a 1×1×1 convolution that reduces the output channels to match the number of all labels. Finally, it is known that the network contains 19 million parameters and avoids bottlenecks by doubling the number of feature channels before each pooling operation.

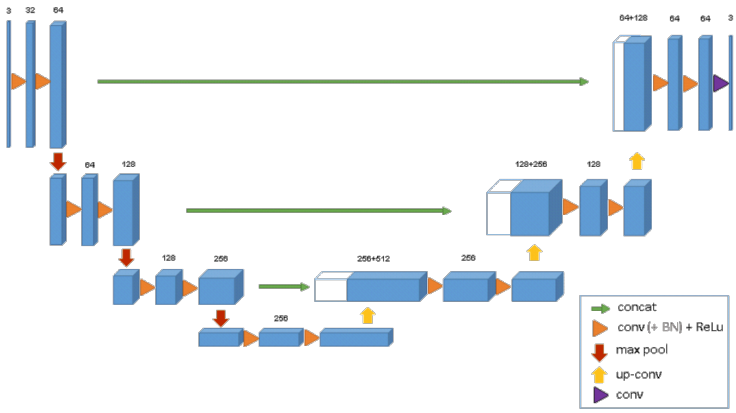


Figure 16: 3D U-Net architecture.

Source: [35]

In Figure 16, note that the blue boxes represent feature maps, and the number of channels is displayed above each feature map.

2.6 Transformer Architectures

Transformer architectures were first presented in 2017 in a revolutionary paper called *"Attention is All You Need"*, which was written by Google researchers [29]. This paper aims to provide a solution to natural language processing (NLP) problems, which had been tackled using traditional techniques such as RNNs and convolutional models up until that moment. Transformers rely exclusively on attention mechanisms, substituting the recurrent layers often employed in encoder-decoder designs with multi-headed self-attention. Due to this innovation, parallel processing is possible, which makes GPU utilization better. Therefore, the model can be trained significantly faster than before, and leads transformer architectures to be considered state-of-the-art for various NLP tasks.

The Transformer architecture is illustrated in Figure 15. It consists of two stacks: the encoder (on the left side) and the decoder (on the right side), both of which utilize self-attention and point-wise fully connected layers. The encoder processes the input through a stack of six identical layers, while the decoder, which also has six layers, includes an additional mechanism to be able to focus on the encoder's outputs. This particular attention mechanism is called "Scaled Dot-Product Attention" in the paper and is represented in this equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

The input is composed of queries (Q) and keys (K), and values (V). In the output, the dot product of the query of all keys (QK^T) is computed, efficiently

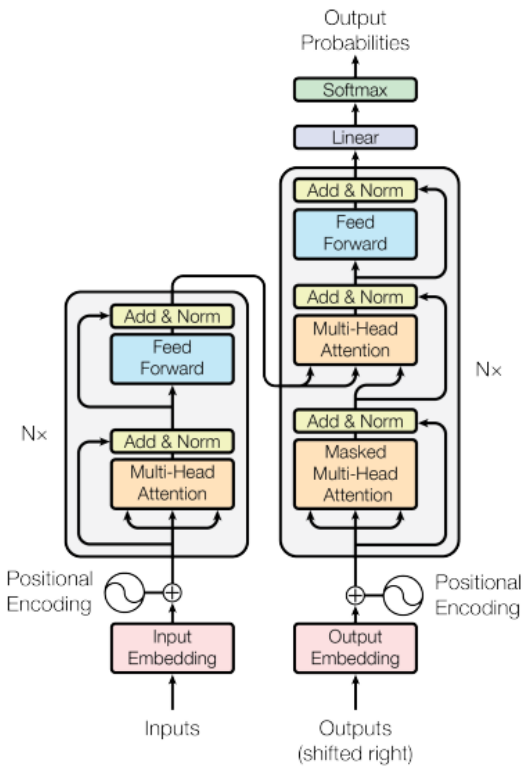


Figure 17: Transformer model architecture.

Source: [29]

calculating the degree of compatibility between each query and key pair. Then, it is divided by $\sqrt{d_k}$, in which d_k represents the dimensionality of the key vectors. This 'scaling' step helps in stabilizing the range of the dot products, especially when d_k is large. The softmax function is applied to acquire the weights on the values.

The self-attention technique enables the model to focus on different segments of the input sequence at different steps by dynamically generating queries, keys, and values for each input sequence. Transformers also utilize multi-head attention, which, as the name implies, introduces multiple parallel attention blocks into this process. Thus, each attention head learns various linear projections of the Q, K, and V matrices, allowing the model to capture different facets of words and their relationships. Overall, this enhances the model's ability to make more accurate predictions.

2.6.1 Vision Transformer

In 2020, Google researchers recognized that transformers could also be effectively applied to image processing, leading to the development and release of the Vision Transformer (ViT) [7]. In this model, as shown in Figure 18, the input image is split into fixed-size patches. A linear embedding is then applied to each patch, along with the addition of positional embeddings. The resulting sequence is passed into a standard Transformer encoder. For classification, the conventional approach of appending a learnable "classification token" to the sequence is used. So, ViT managed to process images as a series of patches and to apply self-attention mechanisms to capture global dependencies, unlike CNNs, which rely on local receptive fields and hierarchical feature extraction. Unfortunately, it doesn't include CNN-like spatial biases, which makes it incapable of training effectively with a limited dataset. Another challenge they face is computational complexity, as self-attention exhibits quadratic scaling with respect to image resolution.

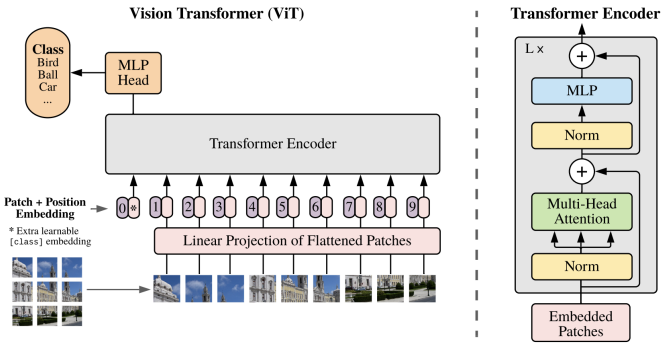


Figure 18: Vision Transformer model architecture.

Source: [7]

Researchers have developed a number of models with the aim to overcome the limitations of ViT. These models, by integrating convolutional processes with transformer architectures, seek to capture both local and global data and therefore, try to improve performance and efficiency in tasks like medical image segmentation. UNETR is one such model, which will be discussed in more detail below.

2.6.2 UNETR Transformer

UNETR (UNEt TRansformers) were developed to tackle the challenge of 3D medical image segmentation [10]. Its core innovation is the use of a pure transformer as an encoder, which leverages embedded 3D volumes to effectively cap-

ture long-range relationships and global context. It features a skip-connected decoder that combines the extracted representations at various resolutions and predicts the segmentation output.

The architecture of the model can be found in Figure 19. In the encoder, the input images are partitioned into non-overlapping patches, which are then flattened and linearly projected into an embedding space. All these embeddings are processed through multiple Transformer layers. Using the skip connections at various resolutions, the decoder extracts representations from the encoder and predicts the segmentation outputs. This design aids in preserving spatial information and improving the accuracy of the results.

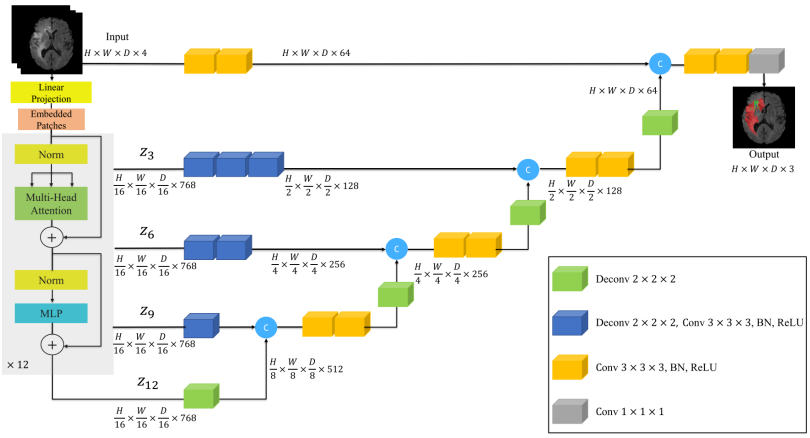


Figure 19: The UNETR model architecture.

Source: [10]

Note that this framework uses a CNN-based decoder instead of transformers. This is due to the fact that transformers are unable to properly capture localized information, despite their great capability of learning global information.

3 Methodology

In this section, a detailed explanation of all the tools and frameworks used, as well as the entire methodology of pre-processing, training, and validating the model that was carried out to achieve our goal, will be provided.

3.1 Frameworks and Tools

3.1.1 PyTorch

PyTorch is an open-source Python library that has grown to be a popular tool within the deep learning research community by combining user-friendliness with accurate performance optimization. One of the most important features is interoperability and extensibility, allowing data to be exchanged within Python libraries easily and efficiently. Moreover, it features automatic differentiation, which means that it can execute dynamic tensor operations immediately. Combined with its C++ core, both CPU and GPU achieve a high-performance, and this was the reason that it was chosen for this thesis [19].

3.1.2 MONAI

Medical Open Network for Artificial Intelligence (MONAI) is an open-source PyTorch-based platform that revolutionizes medical imaging research. Some of the key functionalities include several pre-processing techniques for multi-dimensional data and APIs for higher-level applications. Data transformation and augmentation are also made simple by MONAI for the successful training of the models. It has various loss functions and evaluation metrics, and a Model Zoo that includes several pre-trained models. It also supports multi-GPU and multi-node parallelism for data processing, making it highly useful for tasks such as segmentation, classification, and registration. It consists of three components:

- **MONAI Core**: This is the prime library of the project, which contributes to deep learning model research and development. Especially for training AI models, it provides medical-specific transforms, advanced 3D segmentation algorithms, metrics, losses, and other training parameters. MONAI Core is trusted by researchers in all over the world, making breakthrough discoveries in medical AI.
- **MONAI Label**: This is a clever tool for labeling images that lets you build AI annotation models and train datasets to speed up the development of AI applications in medical imaging [5]. The annotation process is interactive using real-time assistance, AI, and active learning for improving the models. It also offers multi-user collaboration support and direct interaction with 3D viewing programs.
- **MONAI Deploy**: This component is used for the simplification of the deployment of AI models into medical imaging workflows, with a particular emphasis on radiology [9]. Particularly, it has a Pythonic SDK in

order to build standardized, portable AI applications that are ready for deployment in Healthcare. Moreover, it provides a workflow manager that coordinates the different components. Ultimately, it acts as a bridge between AI tools and healthcare systems, making sure data can flow back and forth smoothly using standard formats, like DICOM (Digital Imaging and Communications in Medicine) and FHIR (Fast Healthcare Interoperability Resources).

In this thesis, MONAI Core was utilized for the entire workflow, from data loading and pre-processing to the whole process of model training and validation.

3.1.3 ITK-Snap

ITK-Snap is an open-source software program that enables users to visualize three-dimensional medical images, manually sketch the anatomical parts of interest, and perform image segmentation automatically [33]. In Figure 20, the application interface is shown with a random image from the dataset used in this thesis. ITK-Snap provides a multi-planar view of the CT scan by showing the axial, coronal, and sagittal planes in the top right, top left, and bottom right Panels, respectively. Note that you can navigate through all slices of the image. In the bottom-left panel, the segmentation becomes visible when a labeled image is added, and it also appears with the red color in the CT images. The intuitive interface, fast and clear visualization of segmentation, is why I chose this tool to gain a better understanding of the dataset images and to review my results.

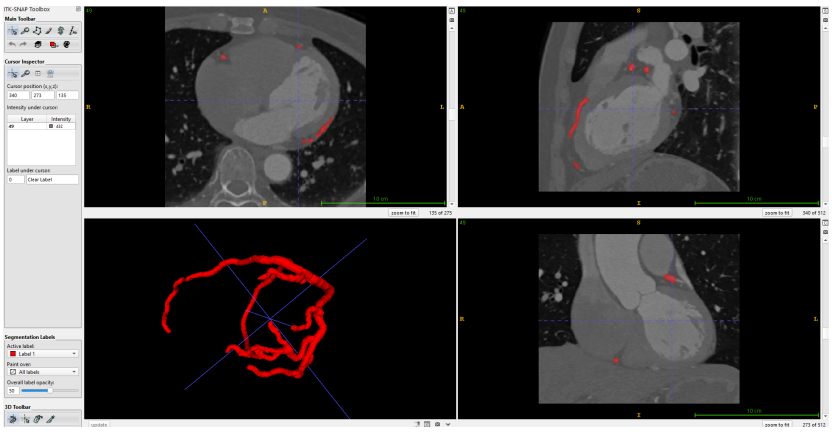


Figure 20: The interface of ITK-Snap.

3.1.4 3D-Slicer

3D Slicer is another free, open-source software framework for processing medical images and visualizing them in three dimensions. It is a clinical research tool,

which is why it offers advanced features, such as supporting versatile visualizations, automated segmentation, and registration for various application domains [8]. It also provides the tools for quantitative analysis and surgical planning. Its interface is similar to the one that ITK-Snap has and is illustrated in Figure 21. The axial, coronal, and sagittal views are portrayed in the top left, bottom left, and bottom right panels, respectively, with the segmentation result displayed in the top right one. A key difference that led me to also use 3D Slicer is its ability to load multiple images and masks simultaneously and switch between them.

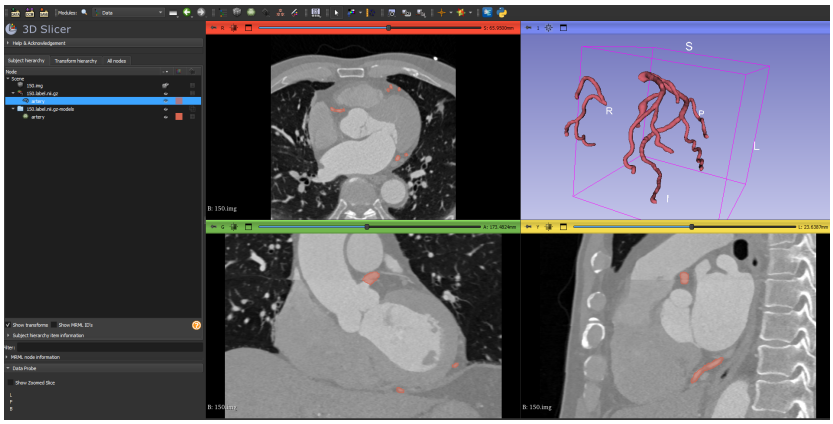


Figure 21: The interface of 3D-Slicer.

3.2 Dataset

The dataset used in this thesis to achieve coronary arteries segmentation is the publicly available ImageCAS database, which has also been employed in previous research, which will be explained in Section 4.4.1. The ImageCAS dataset comprises 3D CTA images of 1,000 patients, obtained from a Siemens 128-dual source scanner. Images were obtained from real clinical cases from Guangdong Provincial People's Hospital between April 2012 and December 2018. The dataset includes patients over 18 years old with known medical histories of ischemic stroke, transient ischemic attack, and/or peripheral artery disease. The dataset also covers patients with CAD who underwent revascularization within 90 days. The dataset has 414 females and 586 males with mean ages of 59.98 and 57.68 years, respectively. The left and right coronary arteries in each image are annotated separately by two radiologists, with cross-validation of their results. In case of disagreements, a third radiologist performs the annotation, and the final result is by consensus [34].

In Table 1, some key details of the dataset are displayed:

Details	ImageCAS Dataset
Samples	1000
Min Shape	512 x 512 x 206
Max Shape	512 x 512 x 275
Voxel Resolution (mm^2)	0.29-0.43
Voxel Spacing (mm)	0.25-0.45

Table 1: Dataset Statistics.

Dataset Division

The dataset was partitioned using a common split, known as the 80/20 rule. However, due to computational limitations, which are explained in Section 3.5, only half of the dataset was used in certain cases instead of the entire set. Nonetheless, the splitting strategy remained the same. Initially, the maximum available number of images was divided into 80% for training and validation, and 20% for testing. Within the training and validation subset, 20% of the data was designated for validation purposes, leaving the remaining 80% for actual training. Below, in Table 2 provides an overview of the dataset splits used in the experiments:

Dataset Division	Number of Samples	
	Original	Limited
Total Dataset	1000	500
Training + Validation	800	400
Training	640	320
Validation	160	80
Testing	200	100

Table 2: Dataset division in original and limited scenarios.

3.3 Pre-Processing

The pre-processing phase is where the medical imaging data are prepared for training and validation. Initially, since the images were significantly large and the coronary arteries are relatively small compared to the entire image, the first step was to remove all unnecessary spatial padding along all axes that was not relevant to the mask. To achieve this, a script that automatically detects and crops the relevant region using the label was developed. Using SimpleITK, a library in Python, the code computes the minimal bounding box around nonzero mask voxels (with optional margin), and then crops both the image and mask accordingly, while also maintaining the original image metadata such as spacing, direction, and orientation. This removes empty slices and reduces data dimensionality, which leads to improved efficiency. To ensure that the code worked and the labels remained intact, the images were visually inspected using

ITK-Snap. As shown in Figure 22, the original image is displayed, while the corresponding cropped version is presented in Figure 23.

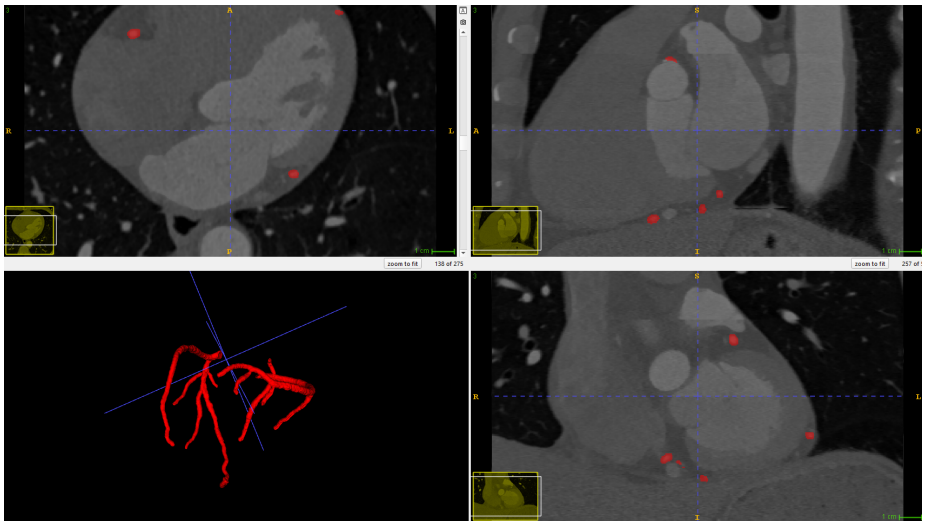


Figure 22: Original image before applying the cropping.

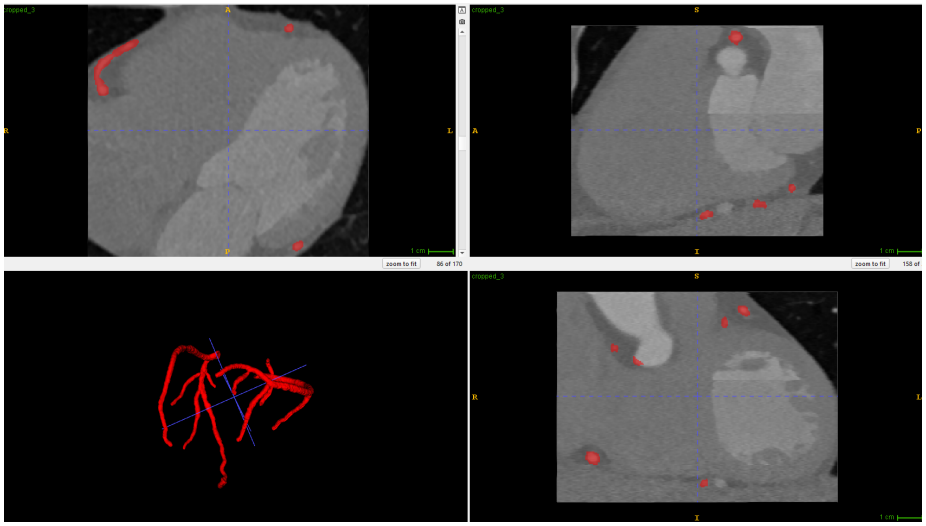


Figure 23: Cropped image.

Subsequently, the data needs to be prepared to fit the model input. The transformation pipelines that are defined for the training and validation sets are almost identical, with a minor difference regarding the voxel spacing, and are applied to both models used in the following experiments. These transfor-

mations include:

1. **Loading images and labels:**
   The transform called `LoadImaged` is responsible for loading both the 3D image and its corresponding segmentation labels and metadata. This step is essential because it ensures that both the image and its label are correctly read into memory as NumPy arrays, while maintaining the spatial metadata for further processing.

2. **Ensuring channel-first format:**
   Many medical imaging datasets store data in the shape (D, H, W), representing depth, height, and width. However, deep learning frameworks like PyTorch expect tensors in the format (C, D, H, W), where C is the channel dimension. `EnsureChannelFirstd` ensures that the image and label tensors include an explicit channel dimension, even if they are grayscale CT scans, so single-channel volumes like this dataset.

3. **Scaling image intensity:**
   CT images often contain a wide range of intensity values, including outliers. To improve model training stability, `ScaleIntensityRanged` transform is used to clip the intensity values from the original range of [-1024, 1500], which is also fixed, since all the images are different, and was found by calculating the mean of all images. Then, it scales them to a more normalized range of [0, 1]. This enhances contrast while retaining meaningful anatomical information.

4. **Resampling to uniform voxel spacing:**
   `Spacingd` transform is applied to make sure that spatial consistency across all volumes is preserved, as all medical images include voxel spacing. This step resamples both the images and their labels to a common spacing of (0.38, 0.38, 0.5) mm, which is calculated as the mean across all data, using bilinear interpolation for the images and nearest-neighbor interpolation for the labels. This step is crucial for accurate spatial learning and preservation of the anatomical features.

5. **Resizing to a fixed region of interest:**
   `Resized` resizes both the image and the mask to a predefined region of interest (ROI), in this case to (128,128,128) spatial size. This is important because deep learning models often require input volumes to be of fixed size. The interpolation mode is chosen as ('area', 'nearest'), where the first computes the average of pixel areas during down-sampling and the second one selects the closest pixel value from the original image. These preserve the overall intensity and brightness better than others and preserve the segmentation masks.

6. **Converting data to PyTorch tensors:**
   `ToTensord`, converts the processed NumPy arrays into PyTorch tensors.

This step is crucial for compatibility with PyTorch-based models and ensures the data can be moved efficiently to the GPU during training and inference.

Although voxel spacing plays a significant role in medical image pre-processing, the desired results can still be achieved without its application. By omitting voxel spacing, computational complexity can be reduced, and the overall processing time is accelerated. For reasons that will be elaborated upon in subsequent sections, this transformation was excluded from the pre-processing pipeline of the UNETR model.

Overall, each of these transformations contributes to a robust and reproducible pre-processing workflow. By aligning image formats, orientations, and intensity scales, the pipeline minimizes variability in the data and allows the model to focus on learning the anatomical features relevant to the segmentation task.

To verify the effectiveness of the transformations, a verification step was implemented to check the transformed images and labels. A selected sample from the training dataset was processed and saved as a NIfTI file for manual inspection. This was especially important to validate the spatial alignment, intensity normalization, and shape conformity of the preprocessed data. In Figure 24 the result of the transformations with the added voxel spacing is portrayed, while in Figure 25, the result without voxel spacing is presented. It is clear that the differences are minor. Also, note that it is the same image used in the previous Figures 22 and 23.

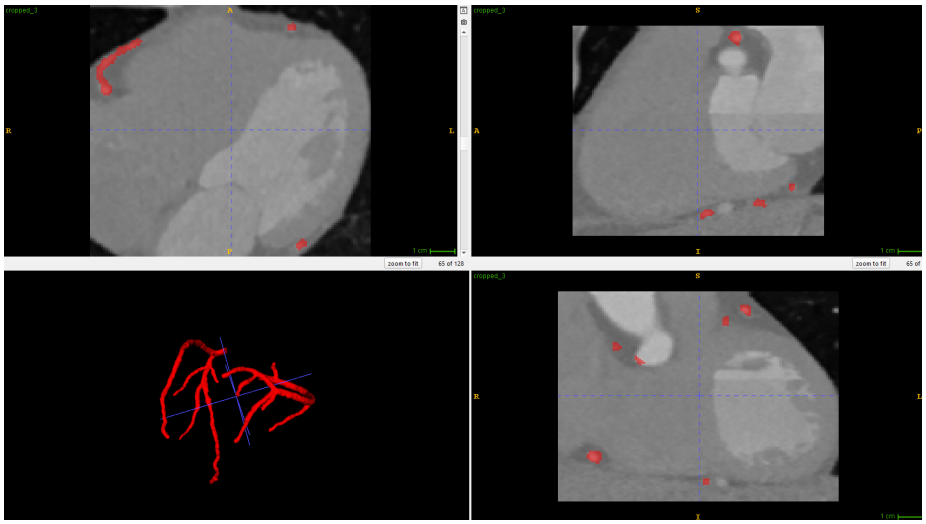


Figure 24: Image after applying all transformations.

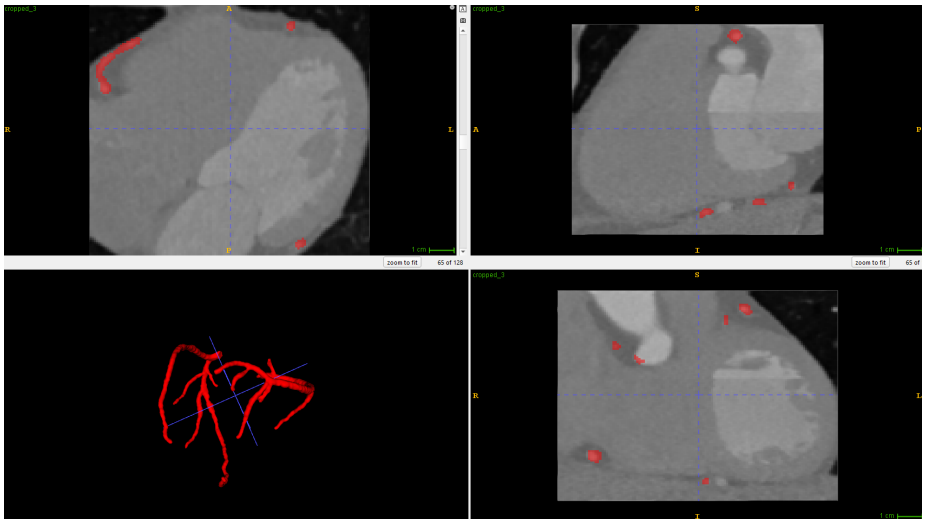


Figure 25: Image after applying all transformations besides voxel spacing.

3.4 Model Architectures

The data was trained using two different models from the MONAI framework, Basic U-Net and UNETR. These models were chosen after experimenting with a few other options MONAI offers. The purpose is to compare the performance of both convolutional-based and transformer-based approaches for the specific task of coronary artery segmentation.

3.4.1 Basic U-Net model

The model that was chosen to evaluate the performance of a convolutional model is Basic U-Net by MONAI. The reason it was preferred over U-Net is its adequacy for rapid experiments and its efficiency even with limited GPU power. Additionally, it demonstrated slightly better performance in validation scores. The model has been set up with the following configuration:

- The **spatial dimensions** (`spatial_dims`) of the input are set to 3, since our dataset consists of 3D CT scans.
- The **input channels** (`in_channels`) are set to 1, as the input data consists of grayscale CT scans, each containing a single channel.
- The **output channels** (`out_channels`) are defined as 2, which is normal for binary segmentation, as this approach allows the background and coronary arteries to be distinguished in two classes.
- The **feature sizes** (`features`) are established at their default value (32, 32, 64, 128, 256, 32), where the first five numbers represent the five-level

encoder feature sizes and the last value relates to the feature dimension after the final up-sampling.

- The **activation function (`act`)** is Leaky ReLU, but with a negative slope of 0.1 added. This way, small gradients are allowed to flow even when the input is negative, helping preserve thin and low-contrast structures like the coronary arteries. Moreover, `inplace` was set to "True" to optimize memory usage during training.
- The **normalization type (`norm`)** is set to the default option, instance normalization. By doing so, the data are normalized across each channel for each sample individually. Moreover, the affine transformation is enabled since it is very useful in medical imaging, where batch sizes are small, as in this case.
- The **dropout rate (`dropout`)** is initialized at 0.1, which means that 10% of the neurons are randomly dropped during each training iteration. This prevents the model from overfitting too early and enhances its ability to generalize to new data.
- The **upsampling method (`upsample`)** is set to "deconv", which applies transposed convolutions in the decoder path. This allows the model to learn how to upsample data in a more optimal way and helps it recover and reconstruct fine spatial details that were lost during downsampling. This improves the resolution and precision of the final segmentation and is better for small and complex structures, like coronary arteries.

For a clearer view of all the configuration settings of the Basic U-Net model, Table 3 is presented.

Parameters	**Values**
Spatial Dimensions	3
Input Channels	1
Output Channels	2
Feature Sizes	(32, 32, 64, 128, 256, 32)
Activation Function	Leaky ReLU
Negative Slope	0.1
Inplace Activation	Enabled
Normalization	Instance
Dropout Rate	0.1
Upsampling Method	Deconvolution

Table 3: BasicUNet model configuration.

3.4.2 UNETR model

The transformer-based model selected for this project is UNETR, as it offers a compromise between computational efficiency and acceptable performance in

medical imaging tasks, such as in BTCV segmentation [10]. Note that some parameters are similar to the previous model and were set to the same values, as both models were trained on the same dataset. A detailed explanation of the configuration settings is provided below:

- The **input channels** (`in_channels`) are also set to 1 in this case, since each input volume is a single-channel grayscale image.
- The **output channels** (`out_channels`) are set to 2, which is equal to the number of classes that the model predicts. As mentioned before, the task is binary segmentation, so the foreground and background represent the two classes to be identified.
- The **image size** (`img_size`) is initialized at roi size. It defines the spatial shape of the model input to make sure it is compatible with the model architecture.
- The **feature size** (`feature_size`) was increased from the default value of 32 to allow a larger feature representation capacity. This parameter defines the number of output channels from the first convolutional layer and also establishes the base number of feature maps that are derived in the initial encoding layers and then are resized in subsequent network layers.
- The **hidden size** (`hidden_size`) was kept to its default size, 768. This parameter refers to the number of hidden units that the hidden layers have.
- The **MLP dimension** (`mlp_dim`) was also kept to the default value, 3072. This represents the dimension of the transformer's feed-forward layer.
- The **number of heads** (`num_heads`) were left unchanged to the value of 12. As the name suggests, this parameter defines the number of attention heads within each Transformer block. The higher the number of heads is, the easier it is to focus simultaneously on multiple features.
- The **projection type** (`proj_type`) defines the kind of layer that is applied for embedding the image patches before they are processed by the transformer encoder. The default value was changed to "perceptron", which uses a learnable projection with a linear or convolutional perceptron block.
- The **normalization method** (`norm_name`) is set to the default option, like in the previous model, instance normalization.
- The **residual blocks** (`res_block`) parameter is set to "true", which enables the use of residual blocks in the decoder. Residual connections enhance the gradients' flow, enabling the training of deeper networks and enhancing the overall performance through identity mappings.

- The **dropout rate** (`dropout_rate`) is set to 0.1, the configuration of the previous model, for the same reason, the fact that it's an approach that helps preserve fine, low-contrast structures such as coronary arteries.

Table 4 presents an overview of all the parameters used to configure the UNETR model.

Parameters	**Values**
Input Channels	1
Output Channels	2
Feature Size	32
Hidden Size	768
MLP Dimension	3072
Number of Heads	12
Projection Type	Perceptron
Normalization	Instance
Residual Blocks	Enabled
Dropout Rate	0.1

Table 4: UNETR model configuration.

3.5 Training and Validation

Overall, the whole code, but particularly the training and validation process, were inspired by MONAI's official tutorials available on GitHub. The final code was a combination of "3D Brain Tumor Segmentation with Swin UNETR (BraTS 21 Challenge)" [22] and "3D Multi-organ Segmentation with UNETR (BTCV Challenge)" [21].

3.5.1 Training Process

The same training strategy was followed for both deep learning models to ensure fair results. The code is epoch-based, which means that training happens in cycles, where one epoch means one full pass through the entire training dataset. Also, gradient clipping was applied for training stability. All the steps of the training process are explained below.

Forward Pass

The forward pass or forward propagation is the initial phase in training. In this phase, the model generates the predicted segmentation masks (logits) based on the input.

Loss Calculation

Then, a loss function computes the prediction error between the logits and the target label, which is the ground truth. The loss function that was used is the DiceCELoss from MONAI, which returns the weighted sum of both Dice loss and Cross Entropy loss. Before selecting this loss function, several alternatives available in the framework were tested. However, none yielded results as effective as the one ultimately chosen. The specific implementation of the code is as follows:

- The background class is included in the loss calculation. This setup was preferred since it gave more accurate results (`include_background=True`).
- Target labels are converted into one-hot encoding format (`to_onehot_y=True`).
- The sigmoid activation is used internally in the loss function (`sigmoid=True`).
- Squared versions of targets are used to penalize the less confident predictions (`squared_pred=True`).

Backward Pass and Optimization

Moving on, the gradients of the computed loss are propagated backwards in order to allow the optimizer to adjust the model parameters, consequently [15]. The optimizer utilized is the AdamW algorithm, which implements the Adam algorithm with weight decay regularization to prevent overfitting. Although the Adam and Stochastic Gradient Descent (SGD) optimizers were also tested, they did not perform as well in comparison. The learning rate (LR) of the AdamW optimizer was set at 0.0002, and the weight decay at 0.0001 after numerous experiments of observing how the loss and dice score progress.

Learning Rate Scheduling

After the optimizer was defined, a learning rate scheduler was used to adjust the learning rate at the end of each epoch. A cosine annealing scheduler was selected, which modifies the learning rate during training by following a cosine-shaped curve, as the name suggests. This scheduling strategy encourages smoother convergence and can lead to improved generalization. The total number of training epochs was set as the cycle length for the scheduler.

3.5.2 Validation Process

The validation strategy, like the training process, was kept identical for both models to ensure a fair and consistent comparison of their performance. One key difference between the two processes is that validation is performed periodically, with the frequency determined by a parameter (`val_every`). All the steps for validating the model's performance on unseen data are outlined below.

Inference Method

Sliding window inference was employed as the inference method to enable the trained models to process and make predictions on previously unseen data. This approach is particularly effective for architectures that require heavy memory, like UNETR, since it processes fixed-size patches. Specifically, each input volume is divided into smaller, overlapping regions that are independently processed by the model. Afterwards, the predictions are combined to reconstruct the full segmentation map. More details regarding the code are shown below:

- The window size (`roi_size`) used for sliding window evaluation is set equal to the ROI size, matching the input size expected by the models.
- The batch size (`sw_batch_size`) defines how many input patches can be analyzed at once. Due to limited GPU power, it was set to the minimal value of 2, even though 4 was also tested.
- The overlap ratio (`overlap`) between adjacent regions along each spatial dimension was set to 0.5. This ensures smoother transitions between patches and minimizes boundary artifacts, leading to more accurate segmentation predictions.

Post-processing of Predictions

During the validation phase, the raw model outputs that are in the form of continuously valued logits were post-processed. This was essential to ensure more reliable evaluation results. In particular, a sigmoid activation function was used to project the logit output to [0, 1] probability values. Then, these probability maps were binarized at a fixed threshold of 0.5 to obtain the final segmentation masks, where each voxel is classified as foreground or background if the predicted probability is greater than the threshold.

Performance Evaluation

The evaluation metric that was chosen to measure segmentation accuracy is `DiceMetric`, which is the equivalent of the Dice coefficient in MONAI. Generally, it measures the overlap between the predicted segmentation masks and the ground truth. In this implementation, the background pixels of the image were included in the DSC calculation, as this resulted in higher scores and more accurate results. Notably, the dice score is computed as the mean value over each batch, and any NaN (Not-a-Number) values are excluded from the metric calculation in order to ensure numerical stability.

In Table 5, all the training and validation hyperparameters are outlined for better clarity.

Training	
Image Size	(128, 128, 128)
Learning Rate	0.0002
Weight Decay	1×10^{-3}
Epochs	100
Batch Size	1
Optimizer	AdamW
Loss Function	Dice Cross-Entropy Loss
Scheduler	Cosine Annealing
Gradient Clipping	Max norm = 1.0
Validation	
Validation Frequency	Every 5 epochs
Inference Method	Sliding Window Inference
ROI Size	(128, 128, 128)
Sliding Window Batch Size	2
Overlap Between Patches	0.5
Activation Function	Sigmoid
Thresholding	0.5
Evaluation Metric	Dice Coefficient
Include Background	True
Reduction Method	Mean over batch
Ignore NaNs	True

Table 5: Training and Validation hyperparameters.

Final Model Selection

During training, the model achieving the highest validation Dice score was saved using a checkpoint strategy. This best-performing model was used afterwards for the visualization and evaluation of the final results.

3.6 Testing Process

For the testing process, the best model checkpoint of each model was obtained during training for the testing, which was reloaded, and the model was set to evaluation mode. The testing datasets were comprised of 100 images for limited-data cases and 200 images for the original dataset. All test volumes and the respective segmentation labels were preprocessed in the same 3D transformation pipeline used during training and validation to maintain consistency between all

stages. This consistency is necessary since the model has only been trained to process input images with particular spatial and intensity properties; presenting raw, unprocessed images for generalization will lead to poor results.

For inference, each of the models generated raw logits for the test inputs and, after applying a sigmoid activation function, was thresholded at 0.5 to generate binary segmentation masks. For the final comparison, both the predicted outputs and ground truth labels were converted to one-hot encoded and discretized. The main metric to measure the segmentation accuracy was DSC, with the background class included. The Dice scores were then computed on the entire test set after excluding any invalid values, such as NaNs, to guarantee a fair and solid evaluation. This quantitative analysis allowed obtaining reliable estimates on the performance of each model when applied to a set of unseen data during the training, as would be the case when deploying any of the models in practice.

3.7 Computational Constraints

Training 3D segmentation models on volumetric medical data is computationally demanding, especially when using transformer-based architectures such as UNETR. Due to GPU memory limitations, the input size was restricted to $128 \times 128 \times 128$, and the batch size was set to 1. A sliding window inference strategy with a sub-volume batch size of 2 was used, as well as gradient clipping to ensure numerical stability during training. As previously mentioned, voxel spacing was not applied during training for the UNETR model, as its inclusion consistently led to training crashes.

Regarding training time, the U-Net model, trained on 320 training and 80 validation volumes, required approximately 6 hours and 10 minutes to complete. When using the original full dataset, the training time increased to approximately 10 hours and 40 minutes. Under the same conditions, the UNETR model required over 12 hours to train on the limited dataset and 17 hours when trained on the full dataset. This underscores the considerably increased computational requirements of the transformer-based architectures. To counter this challenge, the early stopping technique was used, effectively bringing the time taken for the limited one to more than 9 hours and the full dataset to 11 hours and 15 minutes.

All experiments took place on a locally based workstation with Windows 11 Home installed on it, consisting of an Intel Core i7 processor with a speed of 3.5 GHz and 32 GB RAM. An NVIDIA GeForce RTX 3060 with 12 GB of VRAM supported the CUDA computations. This setup allowed 3D medical image segmentation tasks but required compensations like lower batch size, gradient clipping, and sliding-window inference due to the restrictions in memory and GPU size.

4 Results and Discussion

In this chapter, the 3D predictions and results are presented, along with the validation and test Dice scores. Evaluation was performed using two separate test datasets of 100 and 200 images to examine the generalization capabilities of the models under varying test conditions. A comparative analysis of the two models and previous related work is also included to evaluate their performance on the segmentation task.

The experiments presented in this section are part of a sensitivity study that aims to assess the performance of the two models under practical limitations of real-world medical imaging workflows. These include limited data for training, the application of particular transformation strategies that were previously introduced in the pre-processing pipeline, and the potential need for early stopping due to extended training durations and limited computational resources. By comparing the results under these conditions, this section evaluates the robustness, precision, and feasibility of the model deployment in a clinical environment. The experiments below summarize the impact of these factors on segmentation performance across validation and test scenarios.

4.1 Segmentation using the Basic U-Net model

First of all, the results of two experiments conducted using the convolutional-based model, Basic U-Net, will be presented.

First Experiment

The first experiment with this model was performed using the previously described pre-processing pipeline, with a limited dataset consisting of 320 images for training, 80 for validation, and 100 for testing. Upon completion of each training, two performance graphs are generated, one illustrating the average training loss per epoch and another depicting the mean Dice score on the validation set. Following that, visualizations of the predicted segmentation results are provided in 2D, while the outputs are also saved in 3D format to offer a more comprehensive view. All relevant results are presented below.

Metric	Value
Epochs	100
Validation Frequency	5
Training Loss	0.0639
Validation Mean Dice	90.11%
Test Mean Dice	89.94%

Table 6: Summary of BasicUNet performance metrics using the limited dataset.

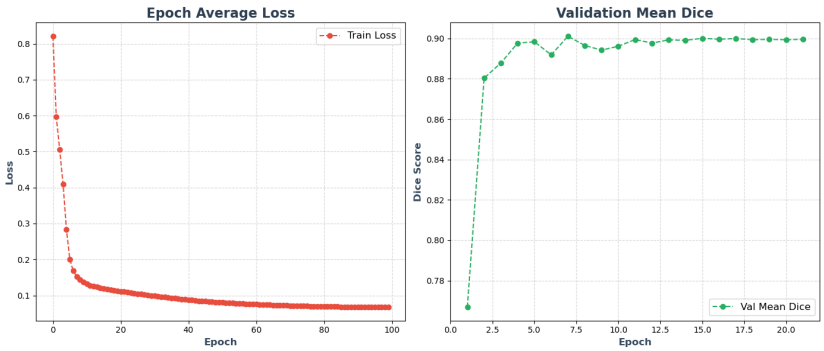


Figure 26: Training loss and validation Dice score curves for BasicUNet trained on the limited dataset.

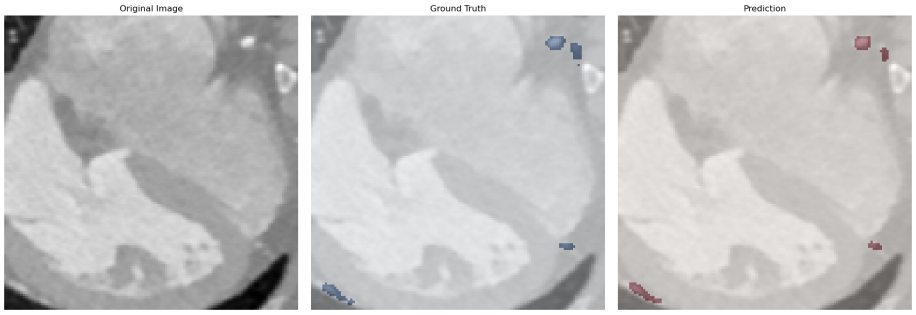
Initially, the Dice scores are presented in Table 6, along with other relevant training metrics. In Figure 26, the left-hand plot illustrates the training loss per epoch, while the right-hand plot shows the validation mean Dice score, which in this experiment is calculated every 5 epochs.

For the training loss, it is noticeable that the curve is decreasing rapidly in the first epochs, which indicates that the model adapts very quickly and learns the data. After the 15th epoch, the curve begins to flatten, suggesting a deceleration in the learning rate. This implies that the model is still improving its performance, but much more slowly. Upon epoch 40 and onwards, the loss stabilizes at a slightly lower value, implying that the model has converged and is no longer significantly improving its performance.

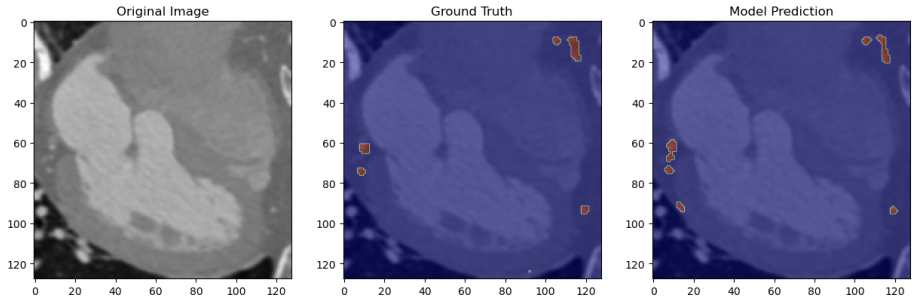
The other plot depicts how effectively the model performs on unseen validation data. The increase is rapid in the early evaluation points, with scores going above 90% at epoch 5 of validation, which is roughly at the 15th training epoch because validation is performed every 5 training epochs. The performance improves gradually and stabilizes at a plateau from about epoch 50 and above. The curve is high with little fluctuation throughout, with values ranging from 89% to 90%. This is an expression that demonstrates that maximum generalization performance has been achieved early by the model and is retained through later epochs. Even though, stability of the curve and lack of downward trend in the validation metric are indicative that overfitting is not exhibited within the training window that is under observation.

Proceeding further, the segmentation results will be presented in 2D and 3D images. The test dice score was 90.11%. In the 2D visualizations, the differences between the predicted and ground truth segmentation are not always easily distinguishable, as they tend to be very subtle. In Figure 27, the 2D axial view of predictions can be seen. These images show the original CT slice, its corresponding ground truth annotation, and its predicted vessel mask. In both, the model is able to identify correctly the vascular structures with high spatial accuracy and continuity. Of note, even in regions with less contrast or smaller diameter vessels, the predictions have anatomical plausibility and

high alignment with ground truth. In some instances, the predicted area is slightly reduced and is missing detail, whilst in others, there is added area to the prediction. These discrepancies become greater in the 3D images.



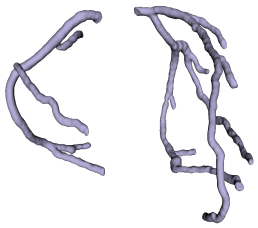
(a) Original CT slice, ground truth, and predicted segmentation. The ground truth annotation is shown in blue, while the predicted segmentation is shown in red.



(b) Overlay of ground truth and predicted segmentation on the same CT slice.

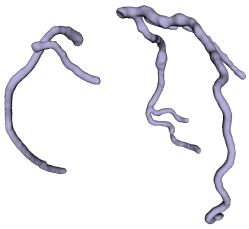
Figure 27: 2D visual comparison of BasicUNet predictions on two different samples from the limited dataset. The top row shows a raw axial view with original CT, ground truth, and prediction. The bottom row presents a different sample with ground truth and prediction masks overlaid.

On Figure 28, sub-figures (a) and (c) display the original ground truth label prior to any spatial or intensity transformations, while sub-figures (b) and (d) show the same label after being incorporated into the training, validation, and testing pipeline. Despite the transformations applied, the anatomical integrity and structure of the vessels are clearly preserved since there are only minor and unnoticeable changes. This confirms the correctness of the pre-processing strategy and its ability to maintain alignment between input images and their corresponding labels.

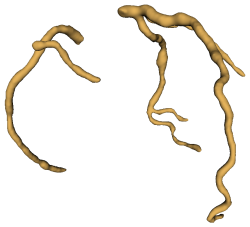


(a) Original label (Sample 842).

(b) Pre-processed label (Sample 842).



(c) Original label (Sample 847).



(d) Pre-processed label (Sample 847).

Figure 28: 3D comparison of ground truth labels before and after pre-processing transformations for two representative samples (842 and 847). Left: original labels; Right: labels used during training and validation.

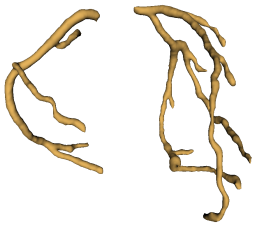
Moving on, Figure 29 provides a 3D visual comparison between the ground truth, which is rendered in yellow, and the corresponding model predictions, which are rendered in red for two representative test cases.

Sample 842 presents a more complex vascular topology compared to Sample 847, yet the model prediction demonstrates a high degree of anatomical alignment. The major vessels and bifurcations are clearly delineated, and the segmentation extends to the distal ends of the coronary tree with notable precision. Slight under-segmentation is observed in some of the thinner branches, consistent with known challenges in segmenting small-caliber vessels due to reduced contrast and limited representation during training. Also, it can be observed that at the ends of the arteries, the model missed a portion of the structure on the left side, while on the right side, it incorrectly added regions that were not part of the actual anatomy.

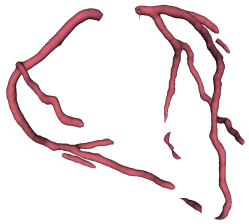
In Sample 847, the predicted segmentation closely follows the morphology of the ground truth, accurately capturing the main coronary branches with high spatial fidelity. The vessel contours are smooth and continuous, and the branching structures are well preserved. Minor deviations are noted in some peripheral regions, likely attributable to the model's sensitivity to partial volume effects. Nevertheless, the prediction remains topologically coherent, with no evidence of disconnection or over-segmentation.

In both cases, the predicted outputs maintain the global orientation and

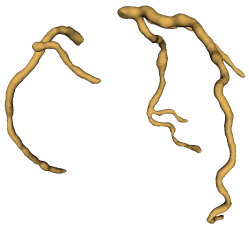
spatial distribution of the coronary arteries, indicating that the model has effectively generalized the geometric patterns present in the dataset. These qualitative findings complement the quantitative Dice scores reported earlier, reinforcing the model's capability to produce near-accurate coronary artery segmentation with strong anatomical precision.



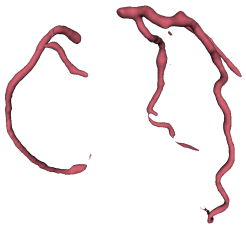
(a) Ground truth after transformations (Sample 842).



(b) 3D prediction (Sample 842).



(c) Ground truth after transformations (Sample 847).



(d) 3D prediction (Sample 847).

Figure 29: Visual comparison between pre-processed ground truth labels and model predictions in 3D for two representative samples (842 and 847). Each row shows the transformed label (left) and the corresponding prediction (right).

Also, visualizations of the transformed ground truth labels and the corresponding model predictions are provided to allow clearer observation of segmentation errors. In Figure 30, the ground truth is shown in yellow and the predictions in red, highlighting regions of under-segmentation and over-segmentation more clearly.

(a) Overlay of ground truth (yellow) and prediction (red) for sample 842.

(b) Overlay of ground truth (yellow) and prediction (red) for sample 847.

Figure 30: Visual overlay comparison of ground truth and prediction masks for two representative samples from the limited dataset. Yellow indicates the ground truth, while red highlights BasicUNet's prediction.

Second Experiment

The second experiment utilized the same pre-processing pipeline and model structure as the first experiment. The difference came about through the use of a much larger dataset of 640 images for training, 160 images for validation, and 200 images for testing. It was the intent of the experiment to determine if using more training data would improve segmentation.

Metric	**Value**
Epochs	100
Validation Frequency	5
Training Loss	0.0657
Validation Mean Dice	90.56%
Test Mean Dice	90.14%

Table 7: Summary of BasicUNet performance metrics using the full dataset.

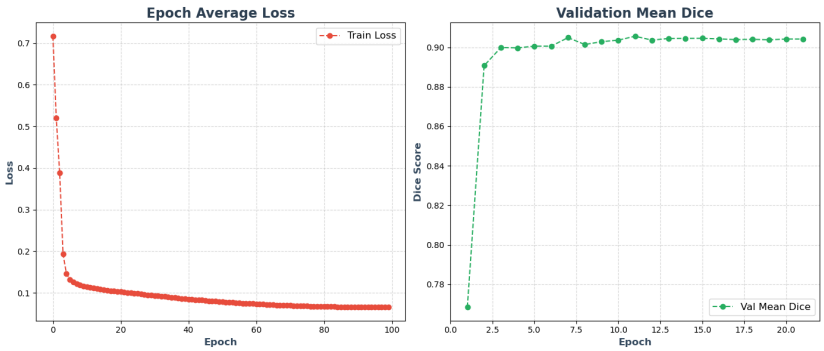
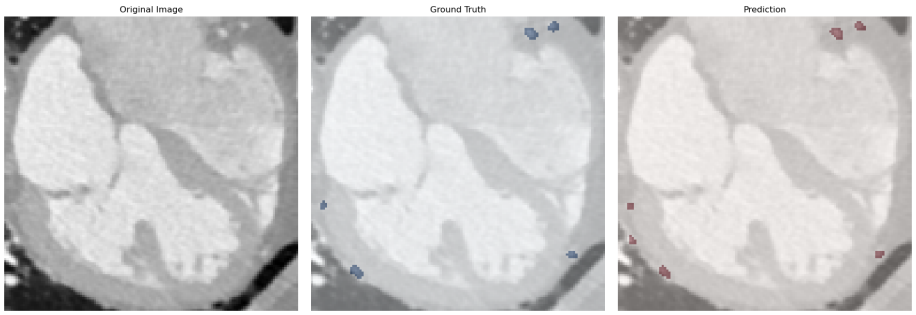


Figure 31: Training loss and validation Dice score curves for BasicUNet trained on the full dataset.

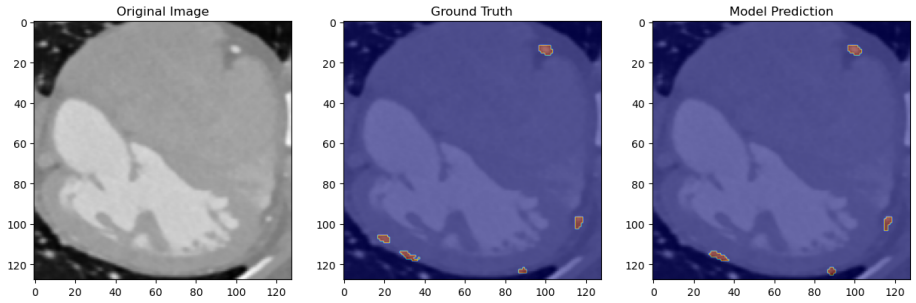
Similar to the initial experiment, the model also demonstrated fast convergence during the initial epochs, such that the training loss dropped sharply and converged towards a stable low. The validation Dice rose rapidly throughout the initial evaluations and stabilized at a high level around a plateau, producing values between 89% and 90% with very small movement, reaching the highest mean at 90.56%. The model achieved generalization early in the training process and does not exhibit signs of overfitting throughout this training period.

The Dice score obtained is 0.9014, which is slightly above the initial experiment's value, indicating a modest improvement due to the extra availability of training sets. Subsequently, 2D and 3D example cases are provided below to investigate further whether the difference between segmentation quality may also be visually apparent beyond the Dice scores.

Figures 32a and 32b display 2D axial slices from the test set for visual evaluation of the segmentation quality in Experiment 2. As shown, the model accurately delineates and identifies the coronary arteries, even within regions of poor contrast, as well as within small vessel structures. The ground truth annotations and the predictions have excellent spatial correlation, and the model captures the overall shape and location of the vessels. Small discrepancies between the ground truth and prediction are visible within finer vessel parts where subtle under-segmentation or over-segmentation might take place. The overall structural integrity remains intact, again affirming the good quantitative quality evident. The overlay visualizations outline these subtle differences while also enhancing the anatomical plausibility of the model's output.



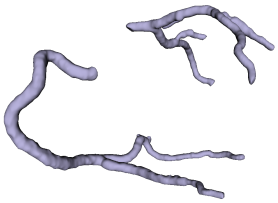
(a) Original CT slice, ground truth, and predicted segmentation. The ground truth annotation is shown in blue, while the predicted segmentation is shown in red.



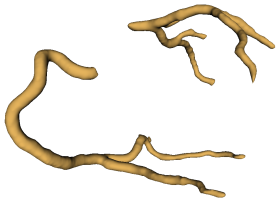
(b) Overlay of ground truth and predicted segmentation masks on the same CT slice.

Figure 32: 2D visual comparison of BasicUNet predictions on two different samples from the full dataset. The top row shows a raw axial view with original CT, ground truth, and prediction. The bottom row presents a different sample with ground truth and prediction masks overlaid.

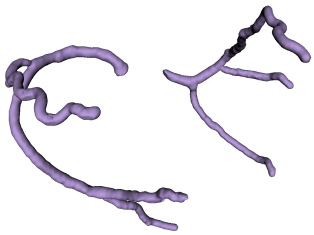
Then, in Figure 33, two representative examples are presented, similar to the previous ones. The ground truth label is first shown in its original form, followed by the transformed version in yellow, illustrating the effect of the applied preprocessing steps.



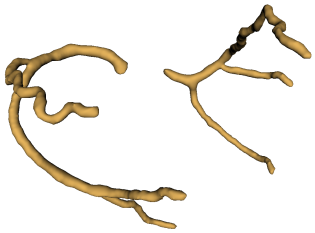
(a) Original label (Sample 873).



(b) Pre-processed label (Sample 873).



(c) Original label (Sample 982).



(d) Pre-processed label (Sample 982).

Figure 33: 3D comparison of ground truth labels before and after pre-processing transformations for two representative samples (873 and 982). Left: original labels; Right: labels used during training and validation.

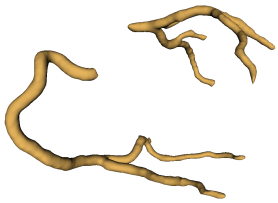
Proceeding further, Figure 34 shows a 3D visualization of the comparison between the ground truth depicted in yellow and the corresponding model predictions depicted in red of two representative samples.

In Sample 873, the model is able to reconstruct the overall vessel topology successfully and to reproduce accurately the major arterial paths and their branching organization. The segmentation closely follows the orientation and spatial arrangement of the ground truth with slight variations in vessel thickness, especially towards the proximal areas. One extra branch is visible on the left artery, though this may be representative of a branch called the SA nodal artery, which is visible in multiple other examples of the dataset. Towards the ends of the vessels, there exist some missing and gap-like segments that reflect the limited capture of finer detail. There is some minor smoothing artifact also seen, however, the anatomical structure of the coronary tree is mostly maintained with no significant disconnections and spurious branches created.

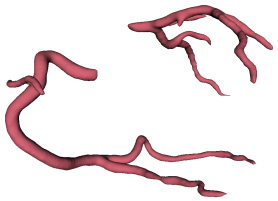
In Sample 982, the model is able to capture the global shape and curvature of the coronary arteries and is also able to detect the principal arterial trees successfully. Nonetheless, there is an additional branch in the distal part of the left artery, and also a spurious additional branch in the right coronary artery. In addition, certain bifurcations either get shortened or do not display enough detail in comparison to the ground truth. Even though the global topology is maintained, localized deviations introduce slight geometrical inaccuracies in the

model. Despite these constraints, the segmentation is structurally coherent and presents a fairly faithful representation of the internal anatomy.

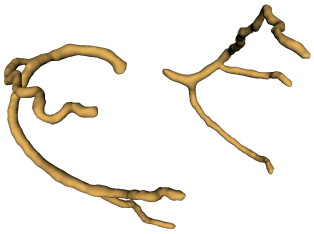
Together, these examples demonstrate that whereas the model remains to excel on the larger dataset employed in Experiment 2, there can be local inaccuracies in more irregular and underrepresented anatomical configurations. However, the overall vessel morphology, organization of branches, and orientation are successfully reconstituted in both instances in accordance with the robust quantitative performance seen.



(a) Ground truth after transformations (Sample 842).



(b) 3D prediction (Sample 873).

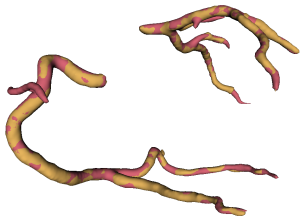


(c) Ground truth after transformations (Sample 982).

(d) 3D prediction (Sample 982).

Figure 34: Visual comparison between pre-processed ground truth labels and model predictions in 3D for two representative samples (873 and 982). Each row shows the transformed label (left) and the corresponding prediction (right).

Finally, combined visualizations of the transformed ground truth and respective model predictions to better observe segmentation inconsistencies are provided. In Figure 35, the prediction is displayed in red and the ground truth in yellow within the same rendering.



(a) Overlay of ground truth (yellow) and prediction (red) for sample 873.

(b) Overlay of ground truth (yellow) and prediction (red) for sample 982.

Figure 35: Visual overlay comparison of ground truth and prediction masks for two representative samples from the limited dataset. Yellow indicates the ground truth, while red highlights BasicUNet's prediction.

4.2 Segmentation using the UNETR model

Moving forward, the results of the two experiments conducted using UNETR, which is the transformer-based model, will be presented below, following a structure similar to the previous analyses.

First Experiment

The first experiment with UNETR was conducted using the same pre-processing pipeline that was described already, with one minor change: the fact that voxel spacing was not applied. This was because applying voxel spacing made the training extremely slow and caused it to stall, even with the limited dataset. It is also important to note that early stopping was applied during training with the UNETR model to prevent overfitting and ensure optimal generalization. The average training loss per epoch alongside the mean Dice score on the validation set are presented in Figure 36 with the performance metrics summarized in Table 8, similar to earlier results.

Metric	Value
Epochs	74
Validation Frequency	5
Training Loss	0.0350
Validation Mean Dice	89.06%
Test Mean Dice	88.91%

Table 8: Summary of UNETR performance metrics using the limited dataset.

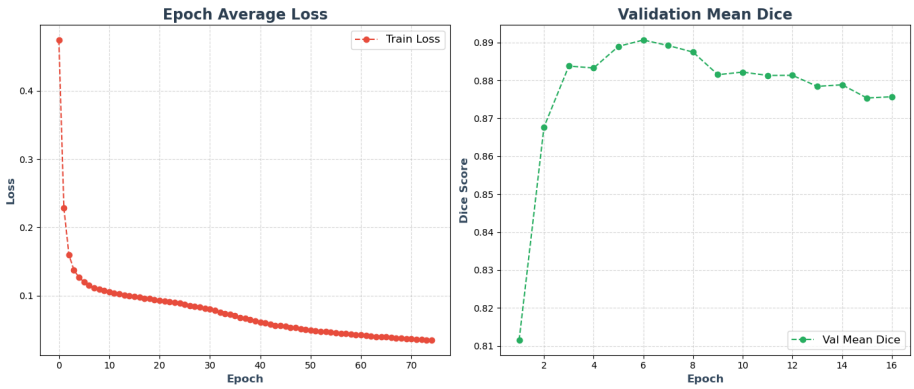
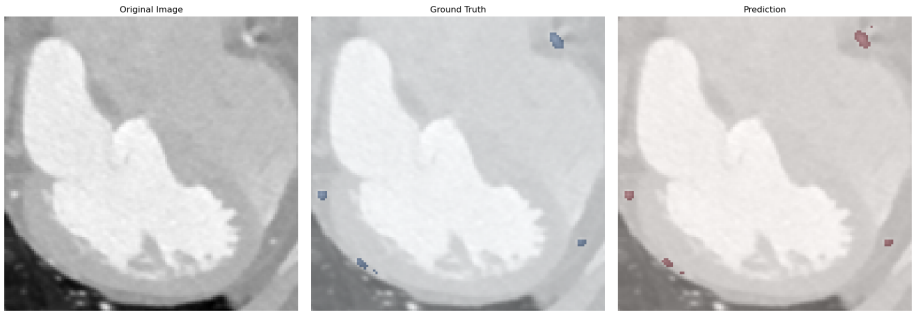


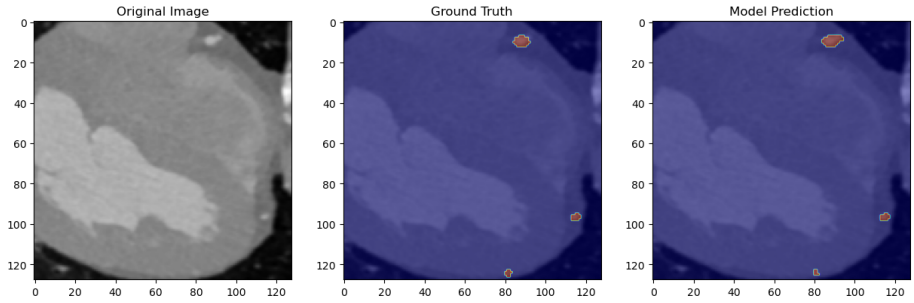
Figure 36: Training loss and validation Dice score curves for UNETR trained on the limited dataset.

The left subplot depicts a gradual and steady decline in the mean training loss, which implies effective learning and convergence of the model. In contrast, the right subplot does indeed demonstrate a spike on validation with Dice score gaining improvement peaking at 89.06% on the 7th validation epoch (so on epoch 35), only to show signs of stagnation and very gradual shortcomings of performance. This behavior indicates possible overfitting beyond this point. To counter this, early stopping was used, which led to the selection of the model checkpoint that provided optimal performance. This way, the finalized model maintained high segmentation accuracy on new data without incurring additional validation performance degradation resulting from excessive training.

Figure 37 presents 2D axial slices of two random samples of the test set for visual assessment of segmentation quality. The model accurately outlines the coronary arteries with good spatial correspondence between ground truth and predicted segmentation once again. The vessel geometries are well delineated, including in low-contrast areas, with good preservation of anatomical form. Although minor misalignment at vessel boundaries is visible for thinner or peripheral segments, the overall structural coherence of the segmented vessels is maintained.



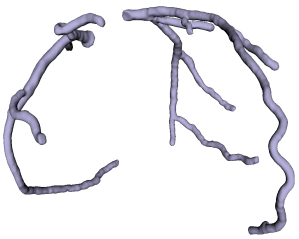
(a) Original CT slice, ground truth, and predicted segmentation. The ground truth annotation is shown in blue while the predicted segmentation is shown in red.



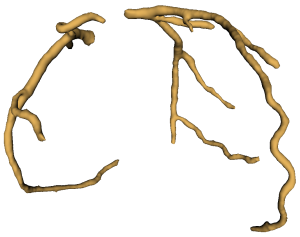
(b) Overlay of ground truth and predicted segmentation masks on the same CT slice.

Figure 37: 2D visual comparison of UNETR predictions on two different samples from the limited dataset. The top row shows a raw axial view with original CT, ground truth, and prediction. The bottom row presents a different sample with ground truth and prediction masks overlaid.

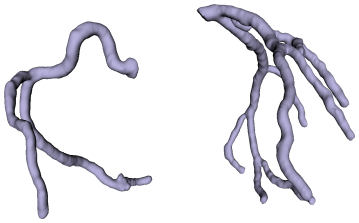
In the upcoming figures, ground truth prior to any transformations are shown colored in lavender, the transformed label is shown in yellow and red is used to display the output of the model, which is a color commonly associated with arteries in medical imaging.



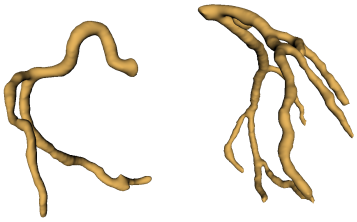
(a) Original label (Sample 839).



(b) Pre-processed label (Sample 839).



(c) Original label (Sample 900).



(d) Pre-processed label (Sample 900).

Figure 38: 3D comparison of ground truth labels before and after pre-processing transformations for two representative samples (839 and 900). Left: original labels; Right: labels used during training and validation.

In Sample 839, the model reconstructs the overall vascular topology well, maintaining the global curvature and orientation of the coronary arteries. Closely following the spatial course of the ground truth, its segmentation match well along the larger branches. Yet, some of the peripheral segments are partially disconnected or appear incomplete, particularly in the distal branches. Those gaps indicate minor under-segmentation, probably due to low vessel thinness or low contrast in the segments. In spite of these problems, the overall primary anatomy is not compromised, and no anatomically implausible structures are inserted. This prediction is spatially and structurally coherent and provides a fairly accurate representation of the coronary anatomy.

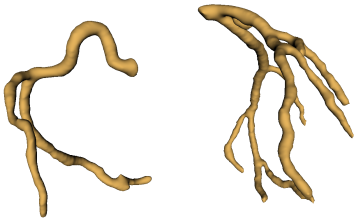
In Sample 900, just like previously, strong anatomical correspondence to ground truth is shown with the major coronary artery branches and bifurcations well-exhibited. The prediction has maintained the global orientation and curvature of the arterial tree and overall vessel topology. The segmentation of the right coronary artery is almost perfect, only failing at the very tip of one branch. The left coronary artery, on the other hand, has more apparent inaccuracies, with many terminal branches not appearing in the prediction. Despite these issues, the coronary arteries are generally well represented in the segmentation results.

These examples illustrate that even with the terminal branch omissions, the overall vessel topography is well preserved. Even with the limited data

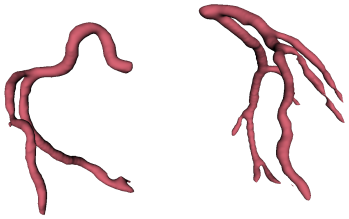
conditions, the model is shown to have a reliable performance, accurately reconstructing the fundamental anatomical geometries to high fidelity.

(a) Ground truth after transformations (Sample 839).

(b) 3D prediction (Sample 839).



(c) Ground truth after transformations (Sample 900).

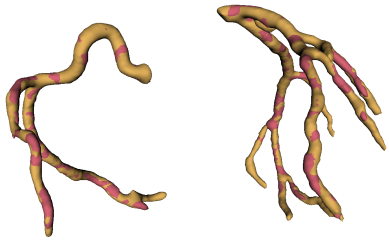


(d) 3D prediction (Sample 900).

Figure 39: Visual comparison between pre-processed ground truth labels and model predictions in 3D for two representative samples (839 and 900). Each row shows the transformed label (left) and the corresponding prediction (right).

At the end, to assess the alignment between the ground truth and the predictions of the model, and to better observe segmentation inconsistencies as well as vessel thickness, Figure 40 is provided.

(a) Overlay of ground truth (yellow) and prediction (red) for sample 839.



(b) Overlay of ground truth (yellow) and prediction (red) for sample 900.

Figure 40: Visual overlay comparison of ground truth and prediction masks for two representative samples from the limited dataset. Yellow indicates the ground truth, while red highlights UNETR's prediction.

Second Experiment

Continuing to the second experiment with the UNETR, all the images from the dataset were fed to the model. The graphs are presented below, along with the metrics table.

Metric	**Value**
Epochs	69
Validation Frequency	5
Training Loss	0.0394
Validation Mean Dice	89.57%
Test Mean Dice	89.56%

Table 9: Summary of UNETR performance metrics using the full dataset.

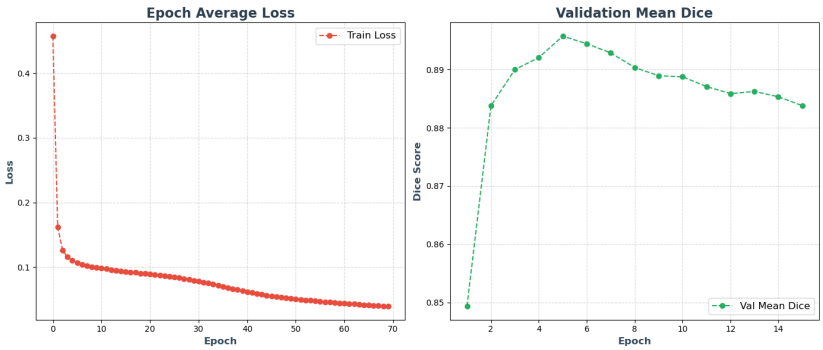
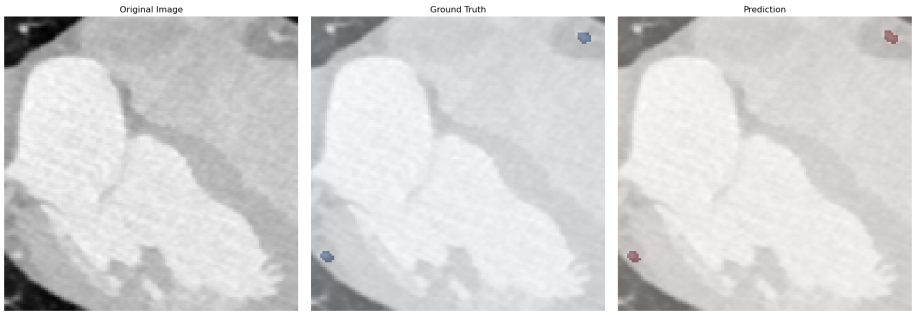


Figure 41: Training loss and validation Dice score curves for UNETR trained on the full dataset.

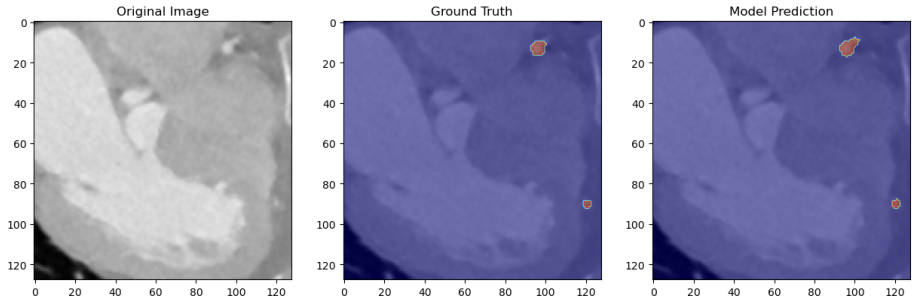
As shown in Figure 41, the training loss in the left subplot consistently decreases throughout the 69 epochs, ultimately reaching a value below 0.04. This indicates that the model was able to learn effectively from the data and maintained stable convergence across the full training duration.

The right subplot displays the mean Dice score of the validation set. A rapid rise is seen early in the epochs, with the peak at 89.64% occurring at validation epoch 5, which is around epoch 15 for the whole training process. A steady decrease is seen from there, indicating the onset of overfitting. To stop the generalization performance from decreasing any further, early stopping was again utilized. This enabled the best checkpoint to be preserved, marking the highest validation Dice score. Consequently, segmentation performance was maintained strongly.

Figure 42 displays additional 2D axial slices from the test set to further evaluate the segmentation quality of the UNETR model. As illustrated, the model continues to demonstrate reliable localization of vascular structures with strong alignment to the ground truth. Vessel shapes are preserved, and segmentation accuracy remains high, even in regions with limited contrast. While very small discrepancies are noticeable at vessel boundaries, particularly in smaller or less defined areas, the global structure and continuity of the vessels are well maintained, underscoring the model's robustness under constrained training conditions.



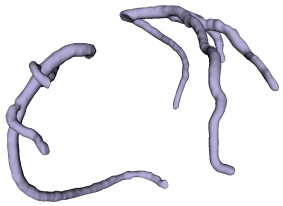
(a) Original CT slice, ground truth, and predicted segmentation. The ground truth annotation is shown in blue, while the predicted segmentation is shown in red.



(b) Overlay of ground truth and predicted segmentation masks on the same CT slice.

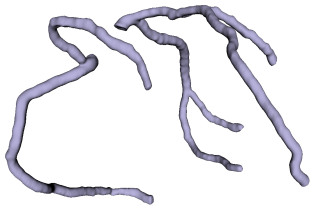
Figure 42: 2D visual comparison of UNETR predictions on two different samples from the full dataset. The top row shows a raw axial view with original CT, ground truth, and prediction. The bottom row presents a different sample with ground truth and prediction masks overlaid.

In the following figures, the original label before any transformations is depicted in lavender, while the transformed label is shown in yellow.

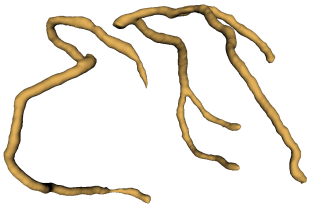


(a) Original label (Sample 825).

(b) Pre-processed label (Sample 825).



(c) Original label (Sample 830).



(d) Pre-processed label (Sample 830).

Figure 43: 3D comparison of ground truth labels before and after pre-processing transformations for two representative samples (839 and 900). Left: original labels; Right: labels used during training and validation.

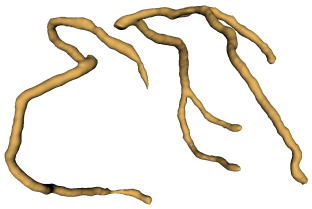
In Sample 825 segmentation demonstrates accurate anatomical correspondence with the vessel outlines, supporting strong spatial agreement with the target anatomy. All of the major vessels are well captured. Some over-segmentation is observed at the vessel tips, where small regions extend beyond the actual arterial boundaries. Additionally, an extra-segmented region appears at the end of one branch of the left coronary artery. Still, the predicted segmentation remains a realistic depiction of the coronary tree. This further strengthens confidence in the ability of the model to generalize to different morphologies of vessels within the dataset.

In Sample 830, the predicted segmentation again seems to represent the primary elements of the coronary arteries quite well, in terms of preserving both vessel connectivity and general topology. Most bifurcations are well reconstructed, and the alignment of the vessel paths with the ground truth is quite satisfactory. However, there is some over-segmented noise found in small regions approximately around the main arteries, which is falsely predicted as structures. These small segments do not affect the overall anatomical interpretation. Several outlying segments exhibit a marginal increase in thickness or a smoother appearance relative to the ground truth, likely due to the fact that the model tries to reduce detailed or redundant signals. Overall, no major concerns of disconnected or spurious branches exist, and the shape remains anatomically realistic.

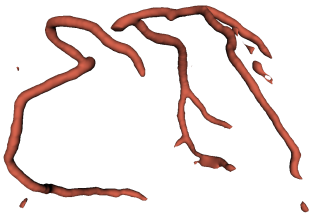
These samples prove that, even with occasional over-segmentation of small parts, the model is still able to capture the coronary arteries with sufficient accuracy.

(a) Ground truth after transformations (Sample 825).

(b) 3D prediction (Sample 825).



(c) Ground truth after transformations (Sample 830).



(d) 3D prediction (Sample 830).

Figure 44: Visual comparison between pre-processed ground truth labels and model predictions in 3D for two representative samples (825 and 830). Each row shows the transformed label (left) and the corresponding prediction (right).

For the final step, to observe the over-segmented regions as well as the thickness of the vessels, the following Figures present the alignment between the ground truth and the predictions.

(a) Overlay of ground truth (yellow) and prediction (red) for sample 825.

(b) Overlay of ground truth (yellow) and prediction (red) for sample 830.

Figure 45: Visual overlay comparison of ground truth and prediction masks for two representative samples from the limited dataset. Yellow indicates the ground truth, while red highlights UNETR's prediction.

4.3 Model Performance Comparison

At this point, a comparison between the two models used in this study, the BasicUNet and UNETR from MONAI, is conducted to evaluate their performance. The comparison begins with an examination of the DSC metrics obtained during training and testing.

Model and Experiment	**Validation DSC (%)**	**Test DSC (%)**
BasicUNet – 1st Experiment	90.11	89.94
BasicUNet – 2nd Experiment	90.56	90.14
UNETR – 1st Experiment	89.06	88.91
UNETR – 2nd Experiment	89.57	89.56

Table 10: Dice Similarity Coefficients (DSC) expressed as percentages for BasicUNet and UNETR models across two experiments.

From Table 10, it is evident that BasicUNet outperforms UNETR in both ex-

periments, with the highest DSC scores being 90.56% for validation and 90.14% for testing in the experiment using the full dataset. The difference in performance stems from the underlying architecture difference between the two models. BasicUNet is a convolutional model that has fewer parameters than UNETR and a more localized receptive field, which encourages better general performance under sparse data conditions. This is better for tasks such as medical imaging segmentation. On the other hand, the transformer-based encoder in UNETR requires a considerable amount of data to sufficiently capture long-range dependencies and spatial hierarchies. The consistently lower DSC scores for UNETR in both experiments may suggest the model's suboptimal performance, bound by the small dataset's constraints. At the same time, however, the second experiment confirms that increasing available training data improves performance and ensures that both models benefit from strengthened generalization enabled through more data. These observations emphasize the need to augment the dataset in building transformer-based segmentation models, while suggesting that UNETR, despite its untapped potential, poses a limitation in reliability, in contrast with BasicUNet, which achieves robust outcomes even with less data.

To further understand the models' performance on specific samples of the CCTA testing dataset, as selected after the data split. The sample numbers correspond to the test set indexing used in this study and do not reflect the original dataset identifiers. Table 11 presents sixteen cases along with their respective test DSC scores.

Sample	**BasicUNet**		**UNETR**	
	1st	**2nd**	**1st**	**2nd**
15	92.61%	93.72%	91.86%	91.88%
16	91.57%	91.35%	89.98%	90.64%
17	89.95%	91.39%	87.33%	89.56%
18	91.33%	92.22%	90.30%	91.40%
19	91.11%	91.90%	90.28%	91.13%
20	90.59%	89.07%	89.22%	89.71%
21	91.64%	92.83%	90.53%	91.34%
22	92.63%	93.02%	91.55%	91.99%
23	88.37%	90.48%	88.24%	88.82%
24	91.52%	91.47%	90.36%	91.38%
25	86.67%	88.10%	85.69%	86.68%
26	88.55%	89.64%	88.05%	88.58%
27	90.81%	92.43%	91.78%	92.28%
28	89.30%	89.95%	88.50%	89.18%
29	90.83%	91.68%	90.32%	91.20%
30	91.40%	92.59%	90.30%	91.16%

Table 11: Dice Similarity Coefficient (DSC) for samples 15–30 using BasicUNet and UNETR models across two experiments.

To look beyond the numbers, three different samples representing good, median, and poor DSC scores are illustrated in the Figures below for our two models, allowing for a side-by-side comparison of their performance.

(a) (b) DSC = 94.8% (c) DSC = 92.9%

Figure 46: Best-case segmentation results. Visual comparison between ground truth, BasicUNet, and UNETR predictions.

(d) (e) DSC = 91.9% (f) DSC = 90.0%

Figure 47: Median-case segmentation results. Visual comparison between ground truth, BasicUNet, and UNETR predictions.

(g) (h) DSC = 83.9% (i) DSC = 81.8%

Figure 48: Worst-case segmentation results. Visual comparison between ground truth, BasicUNet, and UNETR predictions.

As shown, both models excel in the best case, with close agreement with the ground truth. In the median case, segmentation is accurate, with only slight structural irregularities such as a few holes in the branches or slight over-segmentation. In the worst case, however, both models find it difficult to reconstruct smaller branches and are affected by fragmentation, and UNETR is somewhat more discontinuous. This is indicative of the sensitivity of both models to difficult vascular geometry and low-contrast areas.

4.4 Previous Works

In this subsection, previous works on coronary artery segmentation are presented and compared to our results.

4.4.1 ImageCAS: A large-scale dataset and benchmark for coronary artery segmentation based on computed tomography angiography images

The paper by Zeng et al. presents a public benchmark and a unique dataset for coronary artery segmentation from CTA images [34]. In contrast to the majority of previous work that used small datasets or proprietary datasets, ImageCAS consists of 1000 high-resolution 3D CTA scans and offers a structured benchmarking framework that addresses a range of different methods of segmentation, such as direct segmentation, patch-based methods, tree and graph-based methods, and a specially commissioned baseline method.

In their benchmark, deep learning techniques were compared across various configurations using DSC, Hausdorff Distance (HD), and Average Hausdorff Distance (AHD) as the evaluation criteria. Out of these, their method that combines patch-based U-Net++ along with coarse segmentation resulted in the best Dice score of 82.96%. The direct 3D FCN segmentation method scored 80.58%, and the methods based on a graph and tree scored around 70%. For comparison, our models used patch-wise inputs of 128×128×128. The BasicUNet implementation resulted in a DSC of 90.56%, outperforming all the models that ImageCAS evaluated. Likewise, the UNETR implementation resulted in a DSC of 89.57%, still higher among the rest. A comparison of all the models is provided in Table 12.

This performance is an affirmation of the argument that convolutional networks, when properly constructed and trained, remain highly competitive. Furthermore, these findings justify our pre-processing and training methods, especially under challenging 3D patch-based partitioning situations, and underscore that good architecture selection is more crucial than mere complexity.

Method	Input type	Input size	DSC (%)
Direct segmentation (3D FCN) [27]	Full image	$512 \times 512 \times 256$	80.58
Patch segmentation (3D U-net) [11], [4]	Patch	$64 \times 64 \times 64$	72.01
Tree-based segmentation (3D TreeConvGRU) [12]	Tree	$N \times 16 \times 16 \times 4$	68.78
Graph-based segmentation (GCN) [32]	Graph	$N \times 32$	70.61
Baseline method (3D U-net & U-net++) [34]	Patch	$128 \times 128 \times 64$, 16^3, 32^3, 64^3	82.96
BasicUNet (ours)	Patch	$128 \times 128 \times 128$	**90.14**
UNETR (ours)	Patch	$128 \times 128 \times 128$	**89.56**

Table 12: Performance comparison of the methods in the benchmark of ImageCAS and our models in Dice score (%).

Source: [34]

4.4.2 Segmentation of Coronary Arteries using Transformers

The Master's thesis in Computer Science by Michael Staff Larsen examines the usage of Transformer-based models for the task of coronary artery segmentation on three different datasets, one of which is the ImageCAS dataset. A comparative analysis between a standard convolutional neural network, the nnU-Net, and a transformer-based architecture, the Swin UNETR, was performed in his work. Both models were trained and validated on a clinical dataset from a hospital that cannot be accessed freely. The DSC and the 95th percentile Hausdorff Distance (HD95) were used as the fundamental performance metric of this evaluation. The outcomes revealed that the Swin UNETR performed a DSC of 87.66%, marginally better when compared with the nnU-Net architecture with a value of 83.95% on the test set after post-processing.

For the ImageCAS dataset, our BasicUNet implementation obtained a DSC of 90.14%, with the transformer-based UNETR model obtaining a 89.56%. Both of our models performed better than post-processed Swin UNETR with a DSC of 86.14%. These findings show evidence of the excellent segmentation potential of our models, particularly the lightweight CNN-based BasicUNet, supporting that convolutional networks can be highly competitive even against the more advanced transformer-based approaches.

Method	Input type	Input size	DSC (%)
SWIN UNETR [13]	Patch	$160 \times 160 \times 160$	86.14
BasicUNet (ours)	Patch	$128 \times 128 \times 128$	**90.14**
UNETR (ours)	Patch	$128 \times 128 \times 128$	**89.56**

Table 13: Dice score comparison between SWIN UNETR [13] (external benchmark) and our implementations on ImageCAS.

5 Conclusion and Future Work

This thesis presented a comparative evaluation of two deep learning architectures for coronary artery segmentation in computed tomography angiography (CTA) scans. The two selected models are the BasicUNet, a convolutional neural network (CNN), and UNETR, a transformer-based model from the MONAI Framework. They were both trained, validated, and tested under identical experimental conditions to ensure a fair and direct comparison.

Despite the growing interest in the use of transformer architectures, the experimental findings identified that the BasicUNet model consistently outperformed UNETR based on both validation and test criteria. Precisely, BasicUNet achieved the best Dice Similarity Coefficient (DSC) score of 90.14%, while UNETR scored 89.56%. Also, qualitative evaluations, consisting of 2D slices and 3D visualizations, established the high anatomical fidelity of the models, with the CNN-based model delivering more solid predictions over a broader set of test samples. Notably, the fact that UNETR showed slightly more segmentation fragmentation for suboptimal or peripheral vessel locations is a behavior that can be attributed to its greater data dependency and architectural complexity.

These results bring to light an important conclusion: CNNs continue to exhibit robust performance when it comes to medical image segmentation, especially when computational resources and training data are limited. Our BasicUNet model, with its architectural simplicity and computational efficiency, generalized effectively on a relatively modest dataset, performing better than both the UNETR transformer-based model and previously published benchmarks, such as the Swin UNETR and official ImageCAS baseline. It is significant to note, however, that our system's computational constraints may have played an important role. Transformer-based models like UNETR generally have a requirement for more GPU resources and larger datasets to demonstrate their true capabilities, which may explain their comparatively poor performance in this situation.

5.1 Future Work

Although the current results are encouraging, several limitations remain, offering important directions for future research. First, the UNETR architecture may be able to reach its full potential by using a larger and more varied training set. For example, it could incorporate cases with greater anatomical variability, as the current dataset already includes some anomalies. Alternatively, with access to a more powerful GPU, it would be possible to experiment using more computationally demanding transformers, for example, Swin UNETR or TransUNet, which attained state-of-the-art values in various medical image segmentation tasks. These models, when integrated with our carefully designed pre-processing framework, could take advantage of and further enhance performance. Additionally, training and evaluating these models over different datasets, especially those involving more variation in anatomical structures and

imaging protocols, would help establish the robustness and domain's ability to generalize the presented method.

Moreover, potential areas for future research could expand the current binary segmentation approach to multi-class segmentation, where different anatomical and pathological structures, such as plaque, calcification, and the vessel lumen, are segmented separately. This would provide more detailed and more clinically actionable information, particularly for coronary artery disease (CAD) diagnosis and treatment planning. With segmentation of these components, the clinician can assess more accurately the severity of the disease, stenosis, and risk for rupture.

In addition, integrating segmentation output through the use of functional diagnostic methods, like Fractional Flow Reserve (FFR) estimation or lesion classification models, would greatly add clinical utility to the system. As an example, precise lumen segmentation could be utilized directly in computational calculations for FFR, thus delivering a non-invasive but physiologically relevant measurement of stenosis. This form of integration bridges the gap between anatomical and physiologic understanding, and a more complete diagnostic pathway for CAD is gained.

In summary, this thesis provides strong evidence for the feasibility of automatic coronary artery segmentation using deep learning and highlights the continued value of convolutional models, while also opening the door to exploring more advanced architectures under improved data conditions.

6 Bibliography

References

[1] Reza Azad, Moein Heidary, Kadir Yilmaz, Michael Huttemann, Sanaz Karimijafarbigloo, Yuli Wu, Anke Schmeink, and Dorit Merhof. Loss functions in the era of semantic segmentation: A survey and outlook. *ArXiv*, abs/2312.05391, 2023.

[2] Athanasios Bimpas, John Violos, Aris Leivadeas, and Iraklis Varlamis. Leveraging pervasive computing for ambient intelligence: A survey on recent advancements, applications and open challenges. *Computer Networks*, 239:110156, 2024.

[3] James C. Brown, Thomas E. Gerhardt, and Eunice Kwon. Risk factors for coronary artery disease. In *StatPearls [Internet]*. StatPearls Publishing, Treasure Island (FL), 2023. PMID: 32119297, updated 2025.

[4] Yo-Chuan Chen, Yi-Chen Lin, Ching-Ping Wang, Chia-Yen Lee, Wen-Jeng Lee, Tzung-Dau Wang, and Chung-Ming Chen. Coronary artery segmentation in cardiac ct angiography using 3d multi-channel u-net, 2019.

[5] Andres Diaz-Pinto, Sachidanand Alle, Vishwesh Nath, Yucheng Tang, Alvin Ihsani, Muhammad Asad, Fernando Pérez-García, Pritesh Mehta, Wenqi Li, Mona Flores, Holger R. Roth, Tom Vercauteren, Daguang Xu, Prerna Dogra, Sebastien Ourselin, Andrew Feng, and M. Jorge Cardoso. Monai label: A framework for ai-assisted interactive labeling of 3d medical images. *Medical Image Analysis*, 2024.

[6] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[8] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dave Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James V. Miller, Steve Pieper, and Ron Kikinis. 3d slicer as an image computing platform for the quantitative imaging network. *Magnetic Resonance Imaging*, 30(9):1323–1341, Nov 2012.

[9] Vikash Gupta, Barbaros Erdal, Carolina Ramirez, Ralf Floca, Bradley Genereaux, Sidney Bryson, Christopher Bridge, Jens Kleesiek, Felix Nensa, Rickmer Braren, Khaled Younis, Tobias Penzkofer, Andreas Michael

Bucher, Ming Melvin Qin, Gigon Bae, Hyeonhoon Lee, M Jorge Cardoso, Sebastien Ourselin, Eric Kerfoot, Rahul Choudhury, Richard D White, Tessa Cook, David Bericat, Matthew Lungren, Risto Haukioja, and Haris Shuaib. Current state of community-driven radiological ai deployment in medical imaging. *JMIR AI*, 3:e55833, 2024.

[10] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation, 2021.

[11] Weimin Huang, Lu Huang, Zhiping Lin, Su Huang, Yanling Chi, Jiayin Zhou, Junmei Zhang, Ru-San Tan, and Liang Zhong. Coronary artery segmentation by deep learning neural networks on computed tomographic coronary angiographic images. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 608–611, 2018.

[12] Bin Kong, Xin Wang, Junjie Bai, Yi Lu, Feng Gao, Kunlin Cao, Jun Xia, Qi Song, and Youbing Yin. Learning tree-structured representation for 3d coronary artery segmentation. *Computerized Medical Imaging and Graphics*, 80:101688, 2020.

[13] Michael Staff Larsen. Segmentation of coronary arteries using transformers. Master's thesis, Norwegian University of Science and Technology (NTNU), Department of Computer Science, June 2023.

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. pages 2999–3007, 2017.

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[16] Task Force Members, Gilles Montalescot, Udo Sechtem, Stephan Achenbach, Felicita Andreotti, Chris Arden, Andrzej Budaj, Raffaele Bugiardini, Filippo Crea, Thomas Cuisset, Carlo Di Mario, J. Rafael Ferreira, Bernard J. Gersh, Anselm K. Gitt, Jean-Sebastien Hulot, Nikolaus Marx, Lionel H. Opie, Matthias Pfisterer, Eva Prescott, Frank Ruschitzka, Manel Sabaté, Roxy Senior, David Paul Taggart, Ernst E. van der Wall, Christiaan J.M. Vrints, ESC Committee for Practice Guidelines (CPG), Jose Luis Zamorano, Stephan Achenbach, Helmut Baumgartner, Jeroen J. Bax, Héctor Bueno, Veronica Dean, Christi Deaton, Cetin Erol, Robert Fagard, Roberto Ferrari, David Hasdai, Arno W. Hoes, Paulus Kirchhof, Juhani Knuuti, Philippe Kolh, Patrizio Lancellotti, Ales Linhart, Petros Nihoyannopoulos, Massimo F. Piepoli, Piotr Ponikowski, Per Anton Sirnes, Juan Luis Tamargo, Michal Tendera, Adam Torbicki, William Wijns, Stephan Windecker, Document Reviewers, Juhani Knuuti, Marco Valgimigli, Héctor Bueno, Marc J. Claeys, Norbert Donner-Banzhoff, Cetin Erol, Herbert Frank, Christian Funck-Brentano, Oliver Gaemperli, José R.

Gonzalez-Juanatey, Michalis Hamilos, David Hasdai, Steen Husted, Stefan K. James, Kari Kervinen, Philippe Kolh, Steen Dalby Kristensen, Patrizio Lancellotti, Aldo Pietro Maggioni, Massimo F. Piepoli, Axel R. Pries, Francesco Romeo, Lars Rydén, Maarten L. Simoons, Per Anton Sirnes, Ph. Gabriel Steg, Adam Timmis, William Wijns, Stephan Windecker, Aylin Yildirir, and Jose Luis Zamorano. 2013 esc guidelines on the management of stable coronary artery disease: The task force on the management of stable coronary artery disease of the european society of cardiology. *European Heart Journal*, 34(38):2949–3003, 08 2013.

[17] I Ogobuiro, CJ Wehrle, and F Tuma. *Anatomy, Thorax, Heart Coronary Arteries*. StatPearls Publishing, Treasure Island (FL), January 2025. [Updated 2023 Jul 24].

[18] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458v2*, 2015.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[20] Nico H.J. Pijls and Jan-Willem E.M. Sels. Functional measurement of coronary stenosis. *Journal of the American College of Cardiology*, 59(12):1045–1057, 2012.

[21] Project MONAI. Unetr btcv segmentation 3d tutorial, 2021. Accessed: 2025-03-27.

[22] Project MONAI. Swin unetr brain tumor segmentation (brats'21) tutorial, 2022. Accessed: 2025-03-27.

[23] Subha Raman, Jennifer Dickerson, and Roula Al-Dahhak. Myocardial ischemia in the absence of epicardial coronary artery disease in friedrich's ataxia. *Journal of cardiovascular magnetic resonance : official journal of the Society for Cardiovascular Magnetic Resonance*, 10:15, 02 2008.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[25] Alan R. Roth, Andy Lazris, and Sonal Ganatra. Overuse of cardiac testing. *American Family Physician*, 98(10):561–563, November 2018. PMID: 30365287.

[26] Iqbal H. Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(5):1–20, 2021.

[27] Ye Shen, Zhijun Fang, Yongbin Gao, Naixue Xiong, Cengsi Zhong, and Xianhua Tang. Coronary arteries segmentation based on 3d fcn with attention gate and level set function. *IEEE Access*, 7:42826–42835, 2019.

[28] Vasvi Singh and Marcelo Di Carli. Spect versus pet myocardial perfusion imaging in patients with equivocal ct. *Current Cardiology Reports*, 22, 05 2020.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[30] Yan Wang, Yuyin Zhou, Wei Shen, Seyoun Park, Elliot K. Fishman, and Alan L. Yuille. Abdominal multi-organ segmentation with organ-attention networks and statistical fusion. *Medical Image Analysis*, 55:88–102, 2019.

[31] Robert J. Widmer, Zachary P. Rosol, Subhash Banerjee, Yader Sandoval, and Jeffrey M. Schussler. Cardiac computed tomography angiography in the evaluation of coronary artery disease: An interventional perspective. *J Soc Cardiovasc Angiogr Interv*, 3(3Part B):101301, March 2024.

[32] Jelmer M. Wolterink, Tim Leiner, and Ivana Išgum. Graph convolutional networks for coronary artery segmentation in cardiac ct angiography, 2019.

[33] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage*, 31(3):1116–1128, 2006.

[34] An Zeng, Chunbiao Wu, Guisen Lin, Wen Xie, Jin Hong, Meiping Huang, Jian Zhuang, Shanshan Bi, Dan Pan, Najeeb Ullah, Kaleem Nawaz Khan, Tianchen Wang, Yiyu Shi, Xiaomeng Li, and Xiaowei Xu. Imagecas: A large-scale dataset and benchmark for coronary artery segmentation based on computed tomography angiography images. *Computerized Medical Imaging and Graphics*, 109:102287, 2023.

[35] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation, 2016.