ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A NOVEL APPROACH IN 3D RECONSTRUCTION OF CORONARY ARTERY TREE FROM 2D X-RAY ANGIOGRAMS

Ph.D. Thesis by

İbrahim ATLI

Department of Computer Engineering

August, 2020

ANKARA

A NOVEL APPROACH IN 3D RECONSTRUCTION OF CORONARY ARTERY TREE FROM 2D X-RAY ANGIOGRAMS

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfilment of the Requirements for the Degree of Doctor of
Philosophy in Computer Engineering, Department of Computer Engineering

by

İbrahim ATLI

August, 2020

ANKARA

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A NOVEL APPROACH IN 3D RECONSTRUCTION OF CORONARY ARTERY TREE FROM 2D X-RAY ANGIOGRAMS"** completed by **İBRAHİM ATLI** under the supervision of **ASST. PROF. DR. O. SERDAR GEDİK** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Asst. Prof. Dr. Osman Serdar GEDİK

Supervisor

Assist. Prof. Dr. M. Abdullah BÜLBÜL   Assist. Prof. Dr. Elif VURAL

Jury Member   Jury Member

Assist. Prof. Dr. Fatih NAR   Prof. Dr. A. Aydın ALATAN

Jury Member   Jury Member

Prof. Dr. Ergün ERASLAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information, and documents are obtained in the framework of academic and ethical rules,

- All information, documents, and assessments are presented in accordance with scientific ethics and morals,

- All the materials that have been utilized are fully cited and referenced,

- No change has been made on the utilized materials,

- All the works presented are original,

and in any contrary case of the above statements, I accept to renounce all my legal rights.

ACKNOWLEDGMENTS

A NOVEL APPROACH IN 3D RECONSTRUCTION OF CORONARY ARTERY TREE FROM 2D X-RAY ANGIOGRAMS

ABSTRACT

Cardiovascular diseases are common health problems in both developing and developed countries and responsible for one-third of all deaths around the globe. Therefore, accurate diagnosis of vascular diseases plays a critical role in decreasing the rate of deaths and improving the quality of life of the whole world. Although there are many imaging modalities used to diagnose coronary diseases such as DSA, MRA, CTA, etc., X-Ray angiography is used as a gold standard technique among them in clinics. However; it is limited by inherent two-dimensional (2D) representation of three-dimensional (3D) structures and the diagnosis relies heavily on the experience of cardiologists. In this manner, 3D imaging technologies provide an objective, operator-independent tool for an accurate assessment, especially in visualization and quantification of blood vessels. In addition, this technology has the potential to guide clinical decisions. Therefore, the development of an automated and accurate vessel-tree reconstruction from angiograms is highly desirable. In literature, several methods exist for 3D reconstruction of coronary artery trees, but they propose analytical methods that require camera calibration or feature extraction to match correspondences in images for the utilization of epipolar geometry properties. Some of these methods require manual assistance. However; recent developments in technology and tremendous effort drawn in learning-based 3D reconstruction methods enable objects to be reconstructed without any camera calibration or feature extraction steps. In this thesis, we develop an end-to-end fully automated pipeline using deep learning architectures for 3D reconstruction of coronary artery tree from X-Ray angiograms. The pipeline contains mainly 2 steps: (1) blood vessel segmentation and (2) 3D reconstruction of vessels segmented in the first part. We propose a novel fully convolutional deep learning architecture, called Sine-Net, for blood vessel segmentation and multiple deep learning architectures for 3D reconstruction. The input to 3D reconstruction networks is the segmented image extracted in the first part, and the output is a well-defined 3D representation of connected cylinders. This structured

definition for 3D tubular shapes is novel to the literature. As deep learning architectures need plenty of training data with ground-truth, we populate 3D coronary arterial trees synthetically from real data of 9 subjects. After generating relatively enough data for training, we propose multi-view CNN, LSTM and GRU based architectures to predict 3D model of given multiple segmented vessel images. We have validated our method and the data structures defined for 3D tubular shapes in the 3D reconstruction of coronary arteries.

Keywords: Blood Vessel Segmentation; 3D Reconstruction; X-Ray Angiography; Deep Learning; Fully Convolutional Neural Networks

2D X-RAY ANJİYOGRAMLARDAN KORONER ARTER AĞACININ 3D YENİDEN YAPILANDIRMASINDA YENİ BİR YAKLAŞIM

ÖZ

Kardiyovasküler hastalıklar gelişmiş ve gelişmekte olan ülkelerde yaygın olup, dünyadaki tüm ölümlerin üçte birinden sorumludur. Bu yüzden vasküler hastalıkların doğru teşhisi, ölüm oranını azaltmada ve tüm dünyanın yaşam kalitesini artırmada kritik bir rol oynamaktadır. Koroner hastalıkların teşhisi için MRA, CTA, DSA gibi birçok görüntüleme yöntemi olmasına rağmen, X-Ray anjiyografi görüntüleme tekniği kliniklerde en çok kullanılan (altın standart) yöntem olarak karşımıza çıkmaktadır. Ancak, bu görüntüleme tekniği üç boyutlu (3B) yapıların iki boyutlu (2B) temsili ile sınırlı kalmakta ve doğru tanı büyük ölçüde kardiyologların tecrübesine dayanmaktadır. Bu bağlamda 3D görüntüleme teknolojileri, özellikle kan damarlarının görüntülenmesi ve nicelendirilmesinde doğru bir değerlendirme için objektif, operatörden bağımsız bir araç sunabilmektedir. Ayrıca bu teknoloji klinik kararlara rehberlik edebilme potansiyeline sahiptir. Bu nedenle, anjiyogramlardan otomatik ve doğru bir damar ağacı geriçatımının geliştirilmesi oldukça gereklidir. Literatürde, koroner arter ağaçlarının 3B geriçatımı için çeşitli yöntemler mevcuttur, ancak bu yöntemler analitik çözümler sunmakta ve epipolar geometri özelliklerinin kullanılabilmesi için kamera kalibrasyonuna veya öznitelik çıkarımlarına ihtiyaç duymaktadır. Bu yöntemlerden bazıları manuel yardım gerektirmektedir. Teknolojideki son gelişmeler ve öğrenmeye dayalı 3B geriçatım yöntemlerinde gösterilen büyük efor, nesnelerin herhangi bir kamera kalibrasyonuna veya öznitelik çıkarma adımlarına ihtiyaç duymadan da 3B modelinin oluşturulmasını sağlamaktadır. Bu tezde, X-Ray anjiyogramlarından koroner arter ağacının 3B geriçatımı için derin öğrenme mimarileri kullanarak uçtan uca tam otomatik bir yöntem geliştirilmiştir. Bu yöntem temelde 2 adım içermektedir: (1) kan damarı bölütlemesi ve (2) birinci bölümde bölütlenmiş damarların 3B geriçatımı. Kan damarı bölütlemesi için Sine-Net adlı yeni ve tamamen evrişimsel derin öğrenme mimarisi ve 3B geriçatım işlemi için

birden fazla derin öğrenme mimarisi önerilmiştir. 3B geriçatım ağlarına giriş sinyali olarak ilk bölümde çıkarılan bölütlenmiş kan damarı görüntüleri verilmekte olup ve çıktı olarak 3B silindirik yapıları tutabilen bizim tanımladığımız bir veri yapısı bulunmaktadır. 3B tübüler şekillerin tutulmasını sağlayan veri yapısı literatürde yenidir. Derin öğrenme mimarilerinin eğitimi gerçekleştirilirken çok sayıda etiketlenmiş eğitim verisine ihtiyaç duyduklarından, 9 hastaya gerçek verilerden sentetik olarak çok sayıda 3B koroner damar ağacı ürettik. Daha sonra, bölütlenmiş damar görüntülerinin 3B modelini tahmin etmek için CNN, LSTM ve GRU tabanlı derin öğrenme mimarileri geliştirdik. Koroner arterlerin 3B geriçatımında 3B tübüler şekiller için tanımladığımız veri yapısının geçerliliğini ve önerdiğimiz yöntemi deneysel olarak doğruladık.

Anahtar Kelimeler: Kan Damarı Bölütlemesi; 3B Yeniden Geriçatım; X-Ray Anjiyogram; Derin Öğrenme; Tamamen Evrişimsel Sinir Ağları

CONTENTS

NOMENCLATURE

Acronyms

CVD	Cardiovascular Disease
WHO	World Health Organization
DSA	Digital Subtraction Angiography
MRI	Magnetic Resonance Imaging
XRA	X-Ray Coronary Angiography
CTA	Computed Tomography Angiography
CLAHE	Contrast Limited Adaptive Histogram Equalization
CPU	Central Processing Unit
GPU	Graphic Processing Unit
RCA	Right Coronary Artery
LCA	Left Coronary Artery
LAD	Left Anterior Descending
NN	Neural Network
CNN	Convolutional Neural Network
FCNN	Fully Convolutional Neural Network
RNN	Recurrent Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Term Short Memory
2D	2-Dimensional
3D	3-Dimensional
MSE	Mean Squared Error
MTHT	Multi-scale Top Hat Transform
SGD	Stochastic Gradient Descent
DL	Deep Learning
GPU	Graphic Processing Unit
CD	Chamfer Distance
FOV	Field of View
MvFCNN	Multi-view Fully Convolutional Neural Network

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Cardiovascular diseases (CVDs) are serious health problems that lead to death if not treated. According to the research conducted by World Health Organization (WHO), it is reported that an estimated 17.9 million people die (31% of all deaths worldwide) each year from CVDs [1]. There exist many imaging methods utilized for the diagnosis of CVDs such as Magnetic Resonance Imaging (MRI), Digital Subtraction Angiography (DSA), Computed Tomography Angiography (CTA) and X-Ray Angiography (XRA). Among them, XRA imaging modality is one of the most widespread techniques used in clinics [2]. It is a technique to visualize blockages, aneurysms, narrowing, malformations and other blood vessel problems in veins, arteries and organs. Through this procedure, a hollow thin tube, known as catheter, is placed into an artery from somewhere in the arm or the groin and directed toward to the desired position. When it reaches the area to be examined, the contrast matter is injected to make blood vessels visible in X-Ray machine [3]. The machine provides 2-Dimensional (2D) images to monitor vessels in screens as seen in Figure 1.1. Interventional cardiologists capture images from different views of angle [4] and examine these images to make a diagnosis. They decide on treatment planning by investigating the anatomical features of arteries. At this point, a 3D reconstruction system integrated to X-Ray device that enables 3D visualization of arteries in real-time would be beneficial in order to provide clinical assistance to operators.

In some cases, it is possible to treat blocked vessels during the angiogram, but further examination is recommended after surgery to diagnose a variety of vascular problems including aneurysm, vascular malformations, deep vein thrombosis, aortic arc (problems in the arteries that branch off the aorta), peripheral artery disease and renovascular conditions [3]. Since X-Ray angiography is an invasive operation and it is not feasible to repeat the procedure, a tool that allows us to reproduce 3D model of arteries becomes valuable.

Figure 1.1 X-Ray Angiogram image of cardiac veins from [3].

1.1 Motivation

CVD is a common health problem in both developed and developing countries. In Europe 1 of every 5 deaths [5], in the USA 1 of every 7 deaths [6] are caused by this disease. It is responsible for one-third of all deaths around the globe [1]. X-Ray coronary angiography is utilized as a gold standard technique for its diagnosis and treatment [7]. Cardiologists examine multiple 2D images (20-30 images) produced from XRA to reach a decision about the treatment of patients. However, it is not a trivial task due to the following factors: (1) The images are generally in low quality and contrast; (2) X-Ray angiography imaging is limited by an inherent 2D representation of 3D structures. It creates 2D projection images and the projection operation causes a substantial amount of 3D/4D anatomy of the coronary artery information to be lost. (3) Cardiologists need to combine all images mentally in their mind to resolve the problem and its location; (4) The accurate diagnosis heavily relies on the experience of cardiologists, and (5) this creates a subjective evaluation; (6) Finally, the procedure of X-Ray imaging is not convenient to be repeated multiple times because it needs surgical intervention and injection of contrast (radioactive) material directly to arteries. Although tackling these problems is possible manually, it takes hours of processing and evaluation for each patient. It is a cumbersome, repetitive and operator dependent task. Decreasing intra or inter-operator variability

and improving the quality of diagnosis in terms of time and accuracy is possible with an automated tool. In this manner, a system generating a 3D model of heart vessels can provide an objective, reproducible and relatively easier evaluation. Such a system offers a new and accurate approach in diagnosis and has the potential to guide clinical decisions. Therefore, the development of automatic and accurate vessel-tree reconstruction from angiograms is highly desirable. The aim of this study is to develop fast, robust and automated tool for 3D reconstruction of coronary artery tree from X-Ray angiograms to provide clinical assistance.

1.2 The Contribution of the Thesis

The pipeline of the overall system is given in Figure 1.2. Since angiogram images are in low quality and poor contrast with noisy background, the first step in this study is to segment vessels from the background. The second step is to use these segmented images in the 3D reconstruction network for generating the corresponding 3D representation of vessels given in XRA images. We have novelties in both segmentation and 3D reconstruction phases. In order to present them clearly, we divide this section into two parts: Section 1.2.1 introduces novelty in the segmentation of blood vessels and section 1.2.2 introduces novelty in 3D reconstruction. To the best of our knowledge, the overall system proposed in this study is novel to the 3D reconstruction literature specifically in XRA.
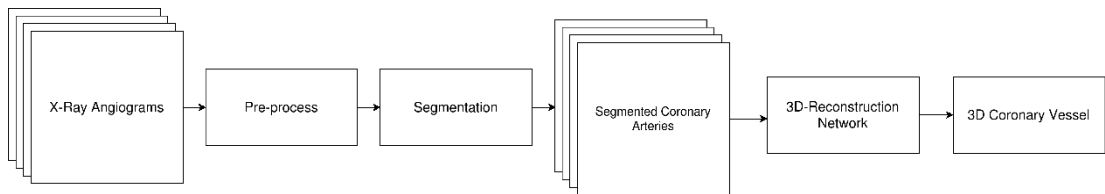
Figure 1.2 Overall pipeline for 3D reconstruction of coronary arteries

1.2.1   Segmentation of Blood Vessels

Blood vessel segmentation plays a critical role in medical imaging because it is a key item for the diagnosis of many diseases in a variety of fields including ophthalmology, neurosurgery, oncology, cardiology and laryngology [8, 9, 10, 11]. A remarkable effort

has been shown to improve the segmentation performance of blood vessels from hand-designed filters to supervised methods. Among all of them, deep network architectures typically achieve the-state-of-the-art performances. We have proposed two deep learning (DL) architectures for the segmentation of coronary arteries. The first architecture is derived from U-Net [12] and presented in our previous study [13]. The second architecture, presented in this thesis, is Sine-Net. It is novel to the literature in its construction style. Although DL based methods perform well on raw input data, appropriate preprocessing operations may improve the segmentation performance. Thus, we conduct additional tests on our networks in order to show their robustness and the effect of preprocessing clearly.

Another contribution of the proposed method is the selection of input size for networks. Since DL methods need plenty of data during training, patch-wise solutions are typically preferred in the related literature. In this solution, full resolution image is divided into smaller pieces, called patches, of definite sizes (i.e. 27x27, 50x50). However, we choose a larger patch size (448x448) to increase local connectivity and reduce computation for overlapping regions. Correspondingly, we present a data augmentation technique for the generation of the training set. Section 4.1 discusses the selection of a larger input size and data augmentation method utilized in the thesis.

1.2.2   3D Reconstruction

3D reconstruction is a topic of high interest in computer vision, robotics, autonomous vehicles and medical fields. Many methods based on analytical calculations are proposed in the literature for obtaining a 3D model of environments, scenes and objects. However, analytical solutions have limitations depending on the method they use such as images should be in color and textural quality should be decent, camera parameters should be known, camera calibration should be handled, a number of images needed to obtain relations and features, etc. Recently, DL based methods have been proposed for 3D reconstruction of objects. DL has shown superior performance on the generation of 3D model of objects; even single image is supplied, but the quality of reconstruction varies depending on the representation of the 3D model. More information related to 3D representations can be found in Section 4.2.1. We have

proposed a novel 3D representation for tubular shapes (Section 4.2.1) and validated the method on coronary vessel trees in this study. The other contribution we have made is the data augmentation procedure (Section 4.2.2) for 3D coronary vessel model generation. We can produce synthetically left and right coronary arteries with the software we developed. Another contribution is that the system we proposed is applicable to use in practice due to its decent running time, fully-automated pipeline and quality of reconstruction.

1.3 Outline of the Thesis

This thesis is organized in 6 chapters. Chapter 1 presents the problem and motivation. The contribution of the thesis is also introduced in this chapter. Chapter 2 presents background knowledge about cardiovascular anatomy, related diseases, imaging modalities, convolutional and recurrent neural networks to warm-up readers and increase understanding of the work presented here. In the next following (CHAPTER 3), a literature review is presented separately for the segmentation of blood vessels and 3D reconstruction. Chapter 4 describes the proposed methods and introduces the implementation details. In Chapter 5, the experimental results of the proposed method are demonstrated and compared with other state-of-the-art works presented in the related literature. Finally, in Chapter 6, a conclusion is drawn that is reached from the results of this study.

CHAPTER 2

BACKGROUND

This thesis is an interdisciplinary study that includes medical and computer science concepts and terminologies. In order to provide a better understanding for readers, this chapter presents medical background along with theoretical information about neural networks in computer engineering.

2.1. Medical Background for Coronary Artery

2.1.1. Anatomy of Coronary Arteries (Heart Vessel Anatomy)

Coronary arteries supply blood that contains necessary nutrients and oxygen to the heart muscle. Also, oxygen-depleted blood is carried away through coronaries. The coronaries have two branches mainly: (1) Right Coronary Artery (RCA) and (2) Left Coronary Artery (LCA). These two main branches exit from aorta and continue to divaricate for covering the surface of the heart. Branches of RCA and LCA are demonstrated from an anterior projection view in Figure 2.1a and 3D volume in Figure 2.1b.

(a)   (b)
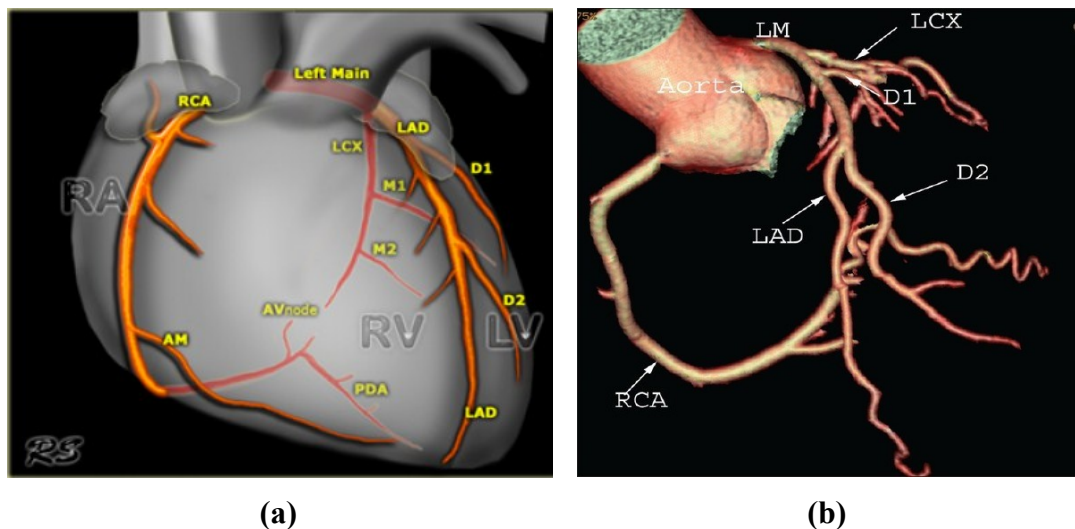
Figure 2.1 Coronary vessel anatomy in the anterior projection on the left [14] and 3D volume rendered from CT angiogram on the right [15].

LCA, alternatively called as left main artery (LMA), originates from the left coronary cusp. At the very beginning, LCA branches into the Left Circumflex Artery (LCX) and the Left Anterior Descending (LAD) artery [14]. Typically, the length up to the first bifurcation point is between 10 and 25 mm. Sometimes (15% of subjects) an external artery occurs at the first bifurcation point and creates a trifurcation; this additional artery is called intermediary artery (IMA) [14, 16]. The LCX artery, which is branched from LMA, supplies blood to the lateral and posterior wall of the left ventricle [17] with obtuse marginals (see M1, M2 in Figure 2.1 or OM1, OM2 in Figure 2.2) vessels. The LAD, which is another branch of LMA, passes through the anterior interventricular groove and proceeds to the apex of the heart. It branches diagonals (D1, D2) and supplies blood to the greater part of left ventricle [14].
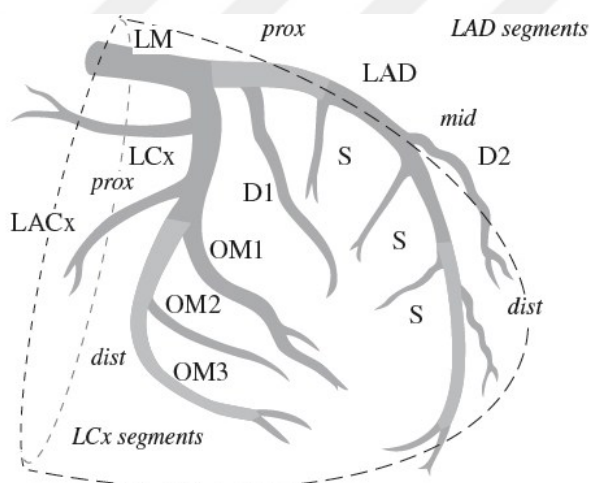
Figure 2.2 LCA labels from the right anterior oblique view in detail [18].

RCA originates from the aorta at the right sinus and travels through the right atrioventricular (AV). It rounds the acute margin of the heart toward crux. RCA can have the first branch, called conus and continues to give off a variable number of branches towards the right atrium and ventricle. The acute marginal branches (M1, M2) emerges with an acute angle to provide blood to the right ventricle. RCA supplies blood to the left ventricle besides providing blood to the right ventricle. For the right dominant circulation, posterior descending artery (PDA) comes off to supply inferior

part of the septum and left ventricle [14]. Detailed labeling belongs to RCA is delineated in Figure 2.3.
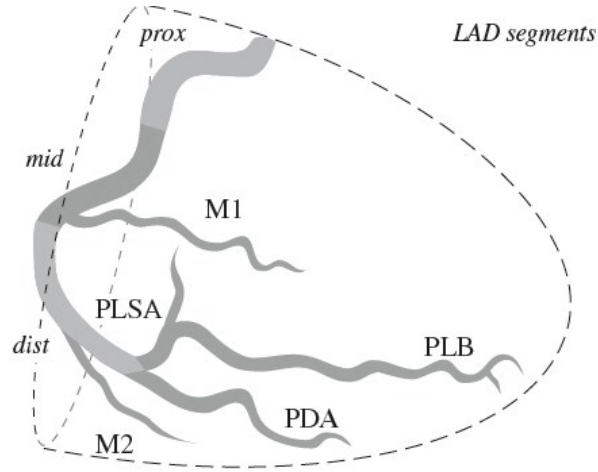
Figure 2.3 RCA labels from the right anterior oblique view in detail [18].

We follow branches in the coronaries to produce 3D synthetic data. Left and right coronaries are examined separately. Section 4.2.2 introduces details about how we generate random synthetic 3D coronary artery models.

2.1.2. Coronary Artery Diseases

Coronary artery diseases (CADs) occur when major blood vessels flowing through the heart muscles (myocardium) is reduced or blocked due to the existence of plaques (fat or cholesterol) or clots in the arteries [19, 20]. This disease can cause a heart to fail because coronary arteries carry fewer nutrients and oxygen to the myocardium. It is one of the most common types of CVDs [21]. Discomfort or pain that may progress towards arms, shoulders, back, neck or chin is widespread symptoms of CADs. While some people have severe symptoms, others may not experience these symptoms at all until they have a complication or a heart attack [22]. CADs have different types that occur in various ways and influence different parts of the organ. Types of CADs can be listed according to [20] as follows:

- Dilated cardiomyopathy: The heart cannot effectively pump blood because chambers are dilated (enlarged) as a result of the weakness developed in

myocardium. There are many reasons behind it, but the most common one is that myocardial ischemia (lack of oxygen supply) caused by CADs. It usually influences the left ventricle.

- Myocardial infarction: This type can alternatively be called heart attack, coronary thrombosis or cardiac infarction. The spasms, partly or complete blockages of coronary arteries by clots or plaques are usually underlying mechanisms of myocardial infarction. These reasons result in abnormal or absence blood flow through myocardium and cause irreversible damage to the part of heart muscle, which weakens the heart. Discomfort or the pain in chest is a widespread symptom of myocardial infarction.

- Heart failure: Congestive heart failure or heart failure occurs when the heart is incapable of pushing blood into the body effectively due to damage caused by the weakness of the heart or other conditions like viruses. High blood pressure, obesity, diabetes or CADs are the main causes of heart failure. The left, right or rarely both ventricle of the heart is affected. The heart becomes too stiff and not fill properly between beats and over time, it cannot respond to blood demands of the body [23].

- Hypertrophic cardiomyopathy: This disease is caused by the thickness of left ventricle and mostly inherited from a person's parents (genetic disorder) by a chance of 50%. The thickness of the wall makes it hard to pump blood vessels out of the heart. Friedreich's ataxia, Fabry disease, and certain medications such as tacrolimus are other causes of hypertrophic cardiomyopathy [24].

- Mitral regurgitation: It is also known as mitral failure, mitral incompetence, mitral insufficiency, or mitral valve regurgitation. As names infer, the mitral valve does not close firmly, and it causes an irregular blood flow from the left atrium to the left ventricle. It is the opposite direction for regular blood flow. As a result of the reverse blood flow caused by the mitral valve, blood cannot leave the heart and be pumped efficiently to the body. Patients with this form of heart disease frequently feel exhausted and breathless.

- Mitral valve prolapse: It occurs when mitral valve between left ventricle and left atrium does not close completely. In many cases of patients, this disorder does not need to be treated if it is not diagnosed as mitral regurgitation.

- Pulmonary stenosis (Pulmonic stenosis): The valve between the right pulmonary artery and right ventricle is called pulmonary valve. Pulmonary stenosis occurs when this valve is tight or stiff which makes blood flow through right ventricle to pulmonary artery hard. To overcome this problem, the right ventricle of the heart must work harder. If the pressure becomes too high in the right ventricle, a procedure is needed to treat pulmonary valve.

2.1.3. Imaging Modalities for Coronary Artery Disease

There are many imaging modalities used in both diagnosis and treatment of CADs. Some studies categorize imaging modalities as invasive/non-invasive [25, 26], while some studies categorize them as anatomical and functional imaging [27, 28]. In our case, we classify imaging modalities as invasive and non-invasive because X-Ray angiogram can be examined both functionally and anatomically which is not convenient to the latter categorization. Table 2.1 summarizes imaging modalities, their corresponding categories and working principles of the modality.

Table 2.1 The classification of imaging modalities used in the diagnosis of coronary artery diseases

Test	Category	Working Principle
Electrocardiogram	Non-invasive	Measures the electrical activity of the heart using electrodes placed on the skin and provide information about the rate, and the regularity of heartbeats [29].
Echocardiogram	Non-invasive	It produces images of the heart using sound waves. This test usually enables doctors to gather information about heartbeats and pumping blood [30].

Cardiac Magnetic resonance imaging (Cardiac MRI)	Non-invasive	Cardiac MRI utilizes magnetic field and radio waves to produce images of the structure within and around the heart. It can be used to evaluate the effect of CAD such as myocardium damages caused by limited blood flow [29].
Computed tomography coronary angiography (CTCA)	Non-invasive	It is used to visualize coronary arteries for plaque and calcium buildup [30] and determine whether the coronaries are narrowed by the injection of contrast material containing iodine [29]. CTCA produces multiple cross-sectional images that can be utilized for 3D transformation. Unlike X-Ray coronary angiography, CTCA is an only diagnostic test, treatment cannot be applied concurrently.
Intravascular ultrasound	Invasive	It produces images of vessels utilizing sound waves for the evaluation of plaque and its composition. It provides significant information about the type of plaque and its amount.
X-Ray coronary angiography (XRA) / Catheter angiography	Invasive	A thin, flexible tube called catheter, is inserted into vein or artery to reach coronary arteries. When it arrives, contrast material is injected to monitor blood vessels by using X-ray beams. The advantage of this modality is that it is possible to diagnose and treat problems in a single procedure. This technique is used as gold standard due to its very detailed, accurate pictures of coronaries [29].

Among imaging modalities explained in Table 2.1, X-Ray coronary angiography is used as gold standard in clinics [7] due to the following advantages:

- The technique may eliminate the need for surgery,
- It presents accurate, clear and detailed coronary artery pictures,
- Non-invasive methods may not produce as much detail as this technique,
- In contrast to CTCA or MRI, it is possible to combine diagnosis and treatment [29].

Since this study aims to reconstruct 3D anatomy of coronary arteries from 2D XRA images, we provide additional information about the types of X-Ray angiography systems. The systems for X-Ray coronary imaging are typically the same and based on C-arm machines. These machines can be in type of single-plane, bi-plane, rotational and dual-axis rotational coronary angiography (DARCA) according to the movement ability of X-Ray source and flat-detector [7] as shown in Figure 2.4. This movement ability in different directions makes the assessment of coronary artery diseases easier. In single-plane angiography, several images are collected from a few fixed views. Choosing the views is up to the operator, therefore; the success rate depends on operator skills. In bi-plane angiography, the system has two C-arms. The position of C-arm is generally adjusted orthogonal and the operator collects images from 2 X-ray sources. In rotational angiography, capturing images are standardized and automated with pre-defined C-arm rotations. It provides panoramic views of coronary arteries independently from operators. Finally, DARCA is an enhanced version of rotational angiography. Rotations of C-arm are optimized to minimize occlusions and foreshortening of coronary vessels. No matter the types of C-arm machines, they produce 2D projection of coronary arteries on an image plane (flat-detector) as shown in Figure 1.1.
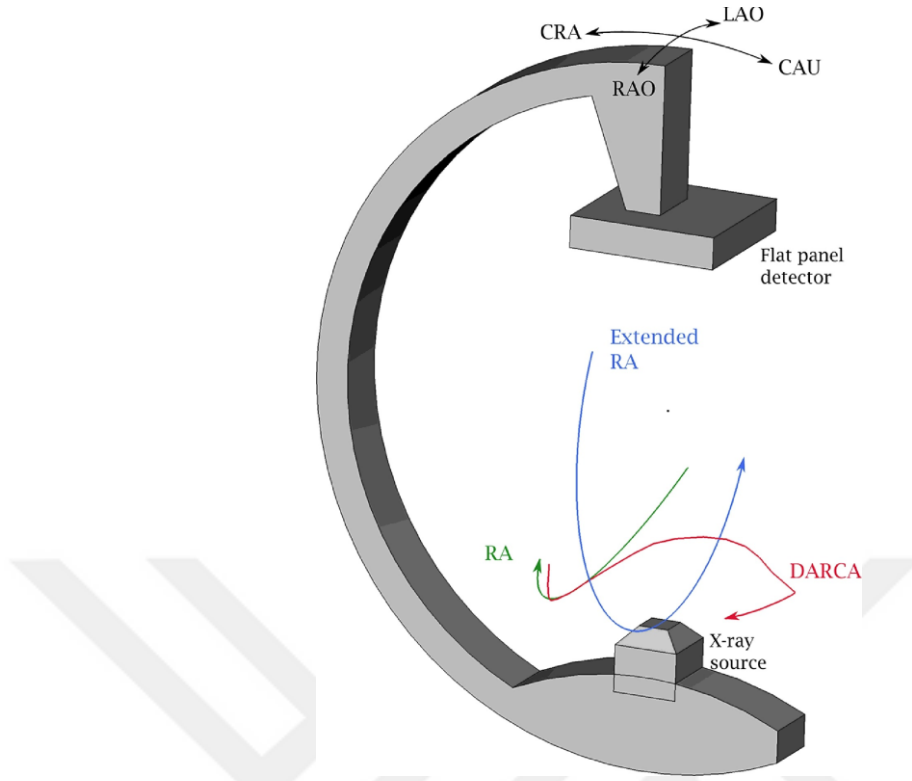
Figure 2.4 Different types of C-arm systems for X-Ray coronary imaging from [7]. Green and blue curves show the movement ability of rotational angiography while red curve shows DARCA systems.

2.2. Theoretical Background for Deep Networks

2.2.1. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a special type of Neural Networks (NN) and still contains cells, called neurons, that have learnable weights and biases. CNN can produce features automatically to create complex relations between input and output. The architectures in CNN mostly assume that the inputs are images so that we see neurons are arranged in a 3-dimensional grid (height, width, depth/channels) as seen in Figure 2.5. However, this does not mean CNN only accepts images, it has also applicable to 1-dimensional, 2-dimensional or 3-dimensional data. Basically, there are five types of layer exist in current CNN architectures: Convolutional layer, de-convolutional (transpose convolution) layer, pooling layer (max-pool, average pool etc.), sampling layer (up-sampling and down-sampling), and fully connected layer.

CNN architectures are made up of stacking these layers. Each layer of CNNs receive 3D volumes and apply convolutional dot product with each learned filter and produce 3D output volumes. In deep learning, many combinations of the CNN layers are stacked to obtain deep network architecture for extracting important features from the input. These features may also be supplied to NN as an input.
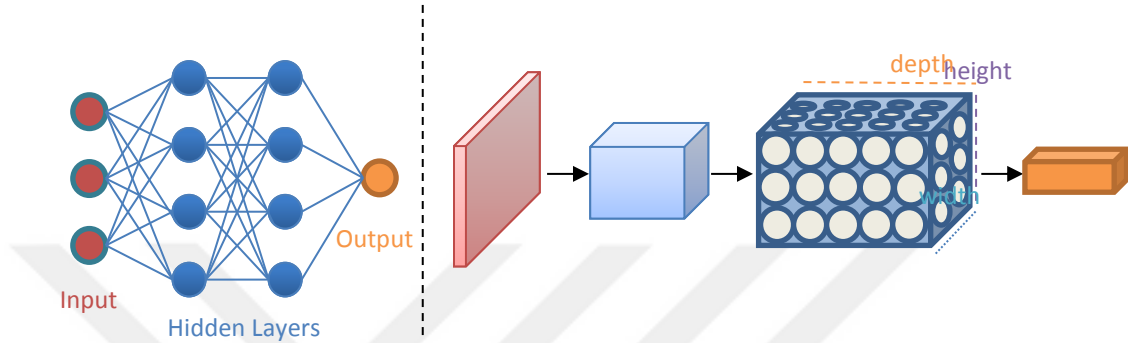
Figure 2.5 Traditional 3-layer neural network structure (left) and 3D CNN structure (right).

Convolution operation, in image processing field, is defined differently what it applied in signals and systems. It is defined as dot product (element by element) of **mxn** matrix by **kxf** filter. There is no flipping operation on filter or matrix. In deep learning architectures, nonlinear transformation is achieved by the convolution operation as in (2.1).

$$y_i^l = g_i^l\left(W_i^l * g^{l-1}\right) \tag{2.1}$$

where * denotes convolution operation. W_i^l denotes the weights of ith filter of lth convolution layer, g^{l-1} denotes the input comes from layer $(l-1)$, and finally g_i^l and y_i^l are non-linear activation function and output of the lth layer for the ith filter, respectively.

2.2.2. Recurrent Neural Networks

Recurrent Neural Networks (RNN) are significantly preferred in processing of sequential data because it keeps an internal state within cells. It is a common method used in natural language processing such as text-generation. It has the potential to be applied in many fields especially if you need to keep a hidden state for the solution via its internal memory. Very popular examples that utilize this method behind are Apple Siri and Google Voice Search. RNNs can be formed in many types according to data, but the principles are the same for all. In RNN, the information is sequentially processed though the network occurring a loop for the data as shown in Figure 2.6. Unlike feed-forward networks, it takes both the current input and what it has learned from the previous inputs into consideration while it is producing an output. RNNs have short term-memory and apply recurrence formula at every step given as:

$$h_t = f_W(h_{t-1}, x_t) \tag{2.2}$$

where h_t is the new state, f_W is some function with parameters W, h_{t-1} is the old state and finally, x_t is the current input vector at time step t. The same function and the same set of parameters are used at every time step as delineated in Figure 2.6.

If an input has a large number of sequences, there are two major problems most likely to encounter with RNNs such as exploding and vanishing gradients. Exploding gradients occur when the algorithm repeatedly assigns a higher number to the weights while in backpropagation due to chain rule (multiplication of partial derivatives through a network). Fortunately, defining a clip limit or truncate gradients can overcome this problem. On the other hand, multiplication of weights due to chain rule causes gradients to vanish when partial derivates are too small and this problem is harder to solve than exploding gradients. Long Short Term Memory (LSTM) [31] proposed by Sepp Hochreiter and Juergen Schmidhuber solves this issue using gates. Details are discussed in Section 2.2.2.1 for LSTM.
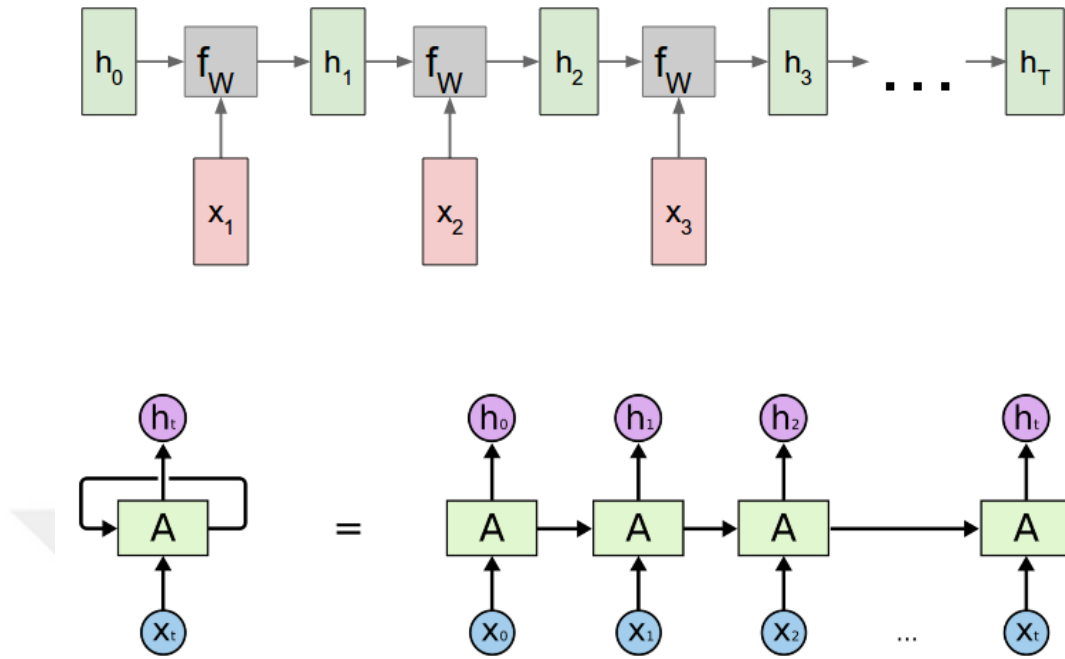
Figure 2.6 RNN sequence process illustration. Top figure [32] shows in every time step t, it applies the same function f_W with its internal state h_{t-1} and the input x_t. The bottom figure [33] demonstrates black box representation of RNN.

2.2.2.1. Long Term Short Memory Network

LSTM [31] is an expanded version of RNNs by extending its memory, meaning that LSTMs have larger memory than RNNs. It becomes a good fit method if very long sequences exist and expect networks to remember previous information for a long period of time. LSTMs resemble a memory in a computer because it can handle read, write and delete operations with the use of gates. LSTM network learns whether to store or delete information by the weights at each gate. Typically, four gates are defined in LSTM as demonstrated in Figure 2.7. RNNs recurrence formula turns into (2.3) for LSTM.

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \tag{2.3}$$

$$c_t = f \odot c_{t-1} + i \odot g \tag{2.4}$$

where c_t represents the cell state, f is forget gate that helps networks to decide whether cell to be erased, i is the input gate decides whether to write information to the cell, g is the gate gate decides how much information write to cell and o is output gate that decides how much information to be revealed from the cell. The gates in LSTM are in the form of sigmoid (σ) which makes the output of gates in the range between 0 to 1. LSTM architectures cope with the vanishing gradient problem presents in RNNs.
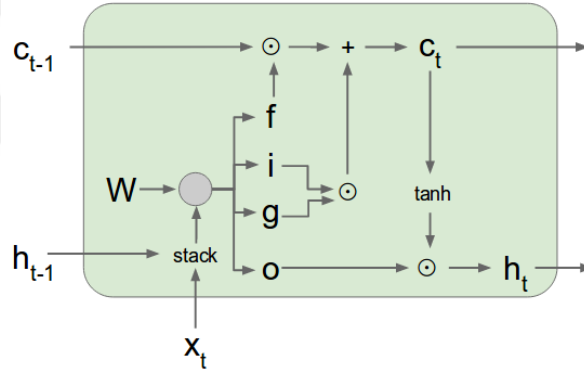
Figure 2.7 Internal gate structure of a typical LSTM [32].

It is noted that the activation function is modified and tested for a variety of problems in the literature. Meaning that there is no exact LSTM recurrence formula for gate activations, but the structure is almost the same as in Figure 2.7. What differs in LSTMs are generally the activation functions.

2.2.2.2. Gated Recurrent Neural Network

Gated recurrent unit (GRU) is introduced by Cho *et al.* [34], and it is very similar to LSTM architecture with fewer parameters. The main differences between GRUs and

LSTMs are the number of gates and maintenance of cell states. It captures adaptively dependencies of different time scales. It controls the flow of information with gating mechanism like LSTM, but without having to use a memory unit [35]. Due to lower number of gating, GRU is computationally more efficient. Typical GRU unit at time instance *t* can be examined in Figure 2.8. The gating mechanism in Figure 2.8 is a commonly used version and can be varied/changed for different applications.
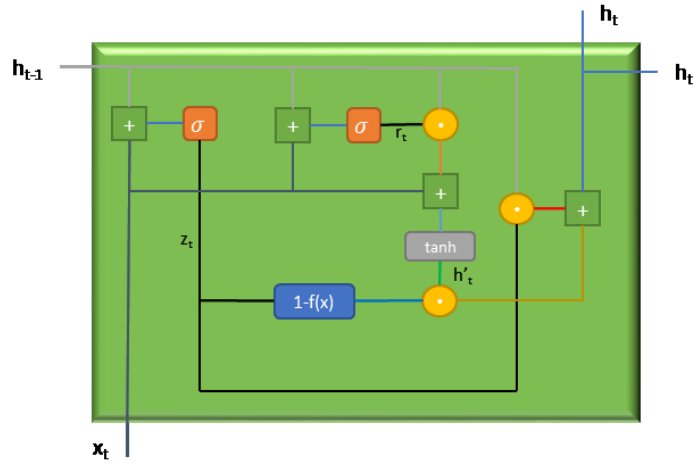
Figure 2.8 Internal gate structure of a typical GRU. Z_t and r_t denote update gate and reset gate, respectively. The current memory content is held in h'$_t$.

The update gate (z_t) decides how much of preceding information to be moved on to the future and it is calculated using (2.5). This equation can eliminate the problem of vanishing gradients.

$$z_t = \sigma(W^z x_t + W^h h_{t-1}) \qquad (2.5)$$

The reset gate (r_t) determines how much of the preceding information to be forgotten by using:

$$r_t = \sigma(W^r x_t + W^f h_{t-1}) \qquad (2.6)$$

The content of current memory is updated using reset gate and current input x_t as following:

$$h'_t = tanh\big(Wx_t + W(r_t \odot h_{t-1})\big) \qquad (2.7)$$

Finally, the current output of GRU or the current hidden state h_t in Figure 2.8 is calculated as:

$$h_t = W\big((1 - z_t) \odot h'_t\big) + W^u(z_t \odot h_{t-1}) \qquad (2.8)$$

2.3. Theoretical Background for 3D Reconstruction from Images

3D reconstruction can be achieved either by active or passive methods. Active methods, which actively interfere with objects and acquire depth maps info with sensors, are beyond the scope of this study. On the other hand, passive methods do not interfere with objects, it only uses reflected or emitted light from the surface of objects with the help of sensors. We present passive methods in two main categories; (1) Analytical methods and (2) Learning-based methods. Coronary artery tree reconstruction is achieved using the first method in the literature. The latter is a relatively new field that emerged with object reconstruction and extended to many areas such as robotics, autonomous driving, augmented reality applications, etc. To the best of our knowledge, currently, there is no learning-based 3D reconstruction method for coronaries. This section introduces analytical and learning-based 3D reconstruction approaches briefly.

2.3.1. Analytical 3D Reconstruction from Images

Various types of systems for XRA have resulted in different strategies to be developed for coronary artery reconstruction. However; they typically follow the same methodology during the reconstruction process. This methodology uses the principle of a finite projective camera model that is extensively covered in computer vision tasks [36]. X-ray angiography imaging can be adapted to this model by defining three

coordinate systems: X-ray source (camera), patient coordinate system and X-Ray detector (image-plane). The parameters which define the relation between camera, image and coordinate system form a matrix called camera calibration matrix or projection matrix. These matrices together with epipolar geometry properties enable mapping points in 3D to their 2D projections. All analytical derivations of these methods can be examined in [36].

All the methods for X-ray coronary reconstruction in the literature utilize the variations of the methodology described in the above paragraph. Projection matrix, fundamental matrix or camera calibration matrix are formed or estimated from given images to achieve 3D reconstruction. Section 3.2 presents the methodologies followed in the literature in more detail.

2.3.2. Learning-based 3D Reconstruction from Images

3D reconstruction is extensively explored in computer vision, robotics and computer graphics communities. It is a longstanding problem and remains challenging due to some factors such as ill-posed views, occlusion, lack of textural quality or image quantity, etc. Camera calibration and multiple images are often required for an effective 3D reconstruction solution. However; the rapid development of learning-based solutions provides a chance to overcome these problems. Although it is a relatively new area, it is possible to reconstruct an object from a single view [37] as humans do by leveraging their prior knowledge about the object. Humans can achieve such tasks easily because they have seen objects previously. Thus, they can recognize the object. In learning-based 3D reconstruction, a similar approach is followed. We treat the problem as a recognition task, and we introduce samples to networks for building prior knowledge in the training phase. The networks proposed in the literature demonstrated promising results in 3D object reconstruction (details can be found in the literature review presented in Section 3.2). We here focus on main architecture mechanisms proposed in learning-based 3D reconstruction of objects.

As we dig the related literature, typically few architectures are proposed according to 3D representations. These are auto-encoder decoder, variational auto-encoders,

generative adversarial networks. Among them, auto-encoder decoder architecture construction is mostly preferred. The auto-encoder part encodes input features into a latent space, and decoder decodes these embedded feature vectors into the desired output. There are various latent space representations and they are determined according to how objects are represented in 3D space. We utilize auto-encoder decoder architecture for 3D coronary artery reconstruction with our 3D representation of tubular shapes presented in Section 4.2.1.

2.4. Challenges in 3D Reconstruction from XRA Images

Dealing with XRA images in order to obtain corresponding 3D models is not an easy task due to the following factors:

- Generally, XRA images are in low contrast and have noisy background because of tissues and bones (ribs) around the heart. Such cases make establishing feature correspondences hard for 3D reconstruction methods.
- The heart is non-rigid, which beats 60 to 100 per minute [38]. This behavior causes shape differences of arteries in time and increases the complexity of 3D reconstruction process.
- Coronary arteries divided into many branches and surround the surface of the heart. To visualize coronaries clearly, images are captured from different views of angles. However, this brings an occlusion problem (vessel overlapping).
- Vessels may be visualized shorter in XRA images than they really are due to different viewing angles. This is called foreshortening effect and this effect is demonstrated in Figure 2.9.
- X-Ray images are generally ill-posed [39]. Camera parameters are mostly unknown, but some X-Ray machines provide camera parameters with the help of external software.
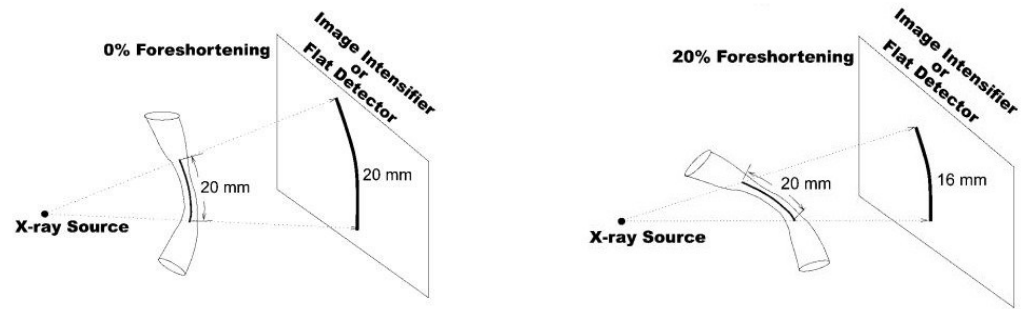
Figure 2.9 Foreshortening effect due to different views of angle [40].

Learning-based 3D reconstruction has the potential to overcome these challenges and eliminate the number of parameters that are used in the analytical 3D reconstruction process.

CHAPTER 3

LITERATURE REVIEW

This chapter presents literature reviews separately for blood vessel segmentation and 3D reconstruction. We first present literature review for the segmentation of blood vessels and continue with 3D reconstruction published works for both coronaries and objects. Finally, we present the location of our methods besides other works in the related literature.

3.1 Literature Review for Vessel Segmentation

A tremendous amount of work has been proposed for segmentation of blood vessels and it is still challenging task due to following factors: (1) Vessel shapes do not follow a simple pattern; (2) Existence of other structures (lesions and optic disk) [41] and (3) various imaging conditions (low image contrast, noise and pathological reasons) make hard to detect vessels. Since the related literature is broad, some reviews [8, 9, 42] categorize proposed methods according to the way they achieve the segmentation for a better understanding of the literature. [9] divides the methods into 4 main categories, [42] divides the methods into 6 main categories, and finally [8] divides the methods into 7 main categories. Further categorizations are provided as subcategories of the main categories. However, it may not be trivial to categorize some of the methods that combine various approaches from different categories together. Therefore, it is better to examine related literature broadly in two big categories: supervised and unsupervised solutions as provided in a recent review [43]. Firstly, we present unsupervised methods literature which mostly covers hand-designed filters. Then, we provide supervised methods of literature which exploits prior labeling information.

3.1.1. Unsupervised Methods

In this category, handcrafted filters that rely on some assumptions have been proposed for the extraction of blood vessels. These filters are mostly based on second-order local structures (Hessian) of an image. Frangi *et al.* [44], the most cited paper in the

literature, define multiscale vesselness measure using eigenvalues of the Hessian matrix. The weakness of the Hessian-based approaches is to find the right scale (Gaussian blur) for vessels especially when they are thick. For example, if the scale is big, thick vessels may disappear or be affected by other adjacent structures. If the scale is small, Hessian-based approaches may fail this time due to second derivative at that point [44]. Another problem with Frangi filter is that it does not produce satisfactory results for images with low contrast. Optimally Oriented Flux (OOF) [45] is proposed to decrease the effect of adjacent objects and detect curvilinear structures. It depends on localized boundary of the local spherical region. The difference between OOF and Hessian matrix is that OOF does not consider closely located objects. However; it is sensitive to edges and performance drops when the structures become very irregular [46].

Several pipelines have been proposed in the literature for the segmentation of blood vessels. Multiscale and region growing approaches are combined in [47]. The first step in the proposed algorithm is to set the initial seed and sigma for Gaussian blur. In each step, Hessian features (Frangi vesselness filter and direction information) are calculated and added to the region if that pixel labeled as a vessel. In the next iteration, the pixel found in the previous step is considered as the seed point for the following step. The algorithm runs for each sigma value in the range $\sigma_{min} \leq \sigma \leq \sigma_{max}$. In this way, proposed method produces a continuous vessel tree. In [48], multiscale graph cut algorithm is used to segment coronary tree. The proposed method introduces direction information into energy function with the combination of Frangi vesselness measure. As it can clearly be seen from literature review, many feature detection algorithms rely on scale selection of Gaussian blur (sigma). To enhance vessels, multi-scale based approaches have been proposed including the evaluation of orientation of eigenvalue of Hessian matrix [49] and filters [50]. The proposed method in [51] adapts Markov Random Field (MRF) multi-label optimization for scale-selection by formulating it as a graph labeling problem. It improves Firangi's scale selection and reduces scale selection error by 15% on average in the Digital Retinal Images for Vessel Extraction (DRIVE) dataset [52].

There are also active contour models addressed in the literature for image segmentation [53]. They find the boundary of an object with different forces defined: external forces, model-based forces, internal and constraining the contour geometry. Active contour models can be examined in two main classes [54]: parametric [55, 56, 57] and implicit active contours [58, 59]. Parametric active contours known as snakes rely on Lagrangian formulation. On the other hand, implicit active contours rely on partial derivatives of contours. [55] proposes a deformable tubular model with the optimization of surface mesh (B-spline to control points) with respect to centerline curve. [56] proposes a parametric deformable model evolved from [55] with defining a changing radius for centerline curve. [57] proposes a method to detect, track and match deformable contours using non-iterative dynamic programming and guarantees the finding of global minimum. Brieva *et al.* [58] use deformable model as B-spline snakes which are special forms of snakes. They are located at vessel boundaries by minimizing the energy produced from S-Gabor filters. [59] introduces a vessel dedicated two level-set scheme (implicit active contour) which iteratively minimizes energy criterion. This criterion is based both on intensity values in the image and on local smoothness properties of the vessel's boundary [59].

Using the statistical distribution of input images, it is possible to develop unsupervised learning-based method in vessel segmentation [9]. Hassouna *et al.* [60] present an automated method for the segmentation of 3D blood vessels from MRA. The method classifies vessel pixels using one Gaussian and background pixels using mixture of one Rayleigh and two normal distributions. The parameters of the distributions are predicted automatically using the expectation-maximization algorithm [61]. K-means method is employed in [62] for the segmentation of liver vessels roughly from MRI. In [63] and [64], lung and liver vessels from CT images are segmented using metaheuristic optimization algorithm, called Ant Colony Optimization. A cost path is defined for all possible paths that connects bifurcation points to obtain optimal vessel tree from physiological model and it is optimized using Ant Colony algorithm. A similar method is followed in [65] for retinal blood vessel segmentation. In [66], retinal blood vessels are segmented using Fuzzy C-means. In [67], the parameters of the

matched filter for the segmentation of retinal vessels are predicted iteratively using Particle Swarm Optimization.

3.1.2. Supervised Methods

Labeled data, called ground-truth (GT), is exploited in supervised methods. A model is created, or the parameters/weights of the model are estimated with the training process using these GTs. A set of rules is learned through this process to create a relation between input (feature vector) and output. This complex relation can be achieved using various methods such as Support Vector Machine (SVM) [68, 69, 70], AdaBoost [71], k-Nearest Neighbor [52], Bayesian Classifier [72], Artificial Neural Network (ANN) [73, 74, 75, 76], or CNN-based [11, 75-87] methods. Here, we present the most popular and best-performing approaches for blood vessel segmentation literature. The methods in related literature mainly use retinal blood vessels because there are different fundus databases with GTs publicly available for training models.

Retinal blood vessels are considered into two parts: large and thin vessels in [68]. Large vessels are detected by adaptive local thresholding in normalized images and thin vessels are classified and segmented with SVM. Wavelet and curvelet multiscale transforms are used as features in SVM. Ricci and Perfetti [69] present two methods for segmentation in fundus images. The first method is based on line-detector at 12 different orientations to obtain unsupervised pixel classification and the second method is based on SVM as supervised vessel classification. The pixel values of two orthogonal line detectors are supplied to SVM for the construction of a feature vector. The work of [70] utilizes SVM and learns configurations of the method proposed in [77] in a supervised way. The method adapted fully-connected Conditional Random Fields (CRFs) for the segmentation of blood vessels.

ANN is a popular method used in many studies for segmentation in blood vessels [73, 74, 75, 76]. ANN scheme is proposed in [73] for pixel classification. The method used in [73] computes 7-D vector composed of gray-level and moment invariants-based features. In [74], the network is trained using backpropagation learning algorithm with 200 features. The features include first principal component values in a 10x10 window

centered on the pixel being classified and the corresponding edge strength values in the same window. In [75], a neural-network classifier for detecting vascular structures in angiograms is trained backpropagation algorithm with the momentum term. It classifies the center pixel using gray-scale information within a window. The work of [76] presents an ANN architecture to find a mapping function to obtain retinal vessel map of a patch of fundus images. The proposed architecture has 5 layers to obtain the complex relationship between input and output.

The improvements in technology and development of powerful hardware enable CNN architectures to evolve faster. In [78], Wang *et al.* propose an algorithm which combines CNN and Random Forest (RF) classifiers for retinal blood vessel segmentation. CNN is used as a hierarchical feature extractor, and then RF exploits these learned features. Finally, ensemble learning is achieved for segmentation from the output of multiple RFs. Khalaf *et al.* [79] propose a modification of the output of CNN as a 3-class problem to reduce intra-class variance. The classes defined in the study are background, small vessels, and large vessels. Validation of this network is accomplished on only DRIVE database. The CNN architecture in [80] is derived from holistically-nested edge detection (HED) CNN structure [81] and has 4 layers instead of 5 as in the original work. The HED architecture composes the output from intermediate layers with conditional random field (CRF) algorithm. A similar approach is followed in multi-scale CNN architectures proposed in [82] and [83] at different scales. In recent years, researchers create deeper architectures by stacking multiple CNN layers in various ways to obtain more and more complex relations. Such configurations of CNNs are called deep learning (DL) architectures in literature. Although they are initially utilized in object recognition tasks [84], they can be adapted successfully in medical image processing field and achieve the-state-of-the-art performance. Guo *et al.* [85] propose a shallow DL architecture consisting of 3 convolutional layers and cascade this network multiple times (n=5) to classify pixels in a patch size of 64x64 for retinal vessel segmentation. Liskowski *et al.* in [86] propose a DL architecture for the segmentation of blood vessels in fundus images. The architecture classifies the center pixel of a patch. They further improve the segmentation process with structured-prediction (SP) definition. In this definition, the

architecture produces predictions for all pixels within a window centered in the patch with the size *s* instead of predicting only center pixel at a time. Similarly, Dasgupta and Singh [87] utilize fully convolutional architecture for SP to segment retinal blood vessels on DRIVE database. The method in [88] balances the number of negative and positive classes using local entropy sampling for training 6 layer network inspired by [89]. CNN architectures are often called fully convolutional if fully connected layers are removed at the end as in [87, 88, 11, 90]. However, they use max-pooling and up-sampling layers inside their DL architectures, which are not convolutional layers. We utilize strided and transpose convolution instead of max-pooling and up-sampling layers, respectively to make fully convolutional network. In this way, networks continue to learn while down and up-sampling operations.

Some of the networks in the literature also carry hierarchical features between layers to enrich feature representations. For instance, Huang *et al.* [91] introduce densely connected CNN architecture which carries features from corresponding layer to other layers in a feed-forward fashion. Feng *et al.* [92] propose a cross-connected convolutional neural network (CcNet) architecture that merges some of the layer features in intermediate locations. U-Net [12] and its derived architectures proposed in [93] and [94] follow a similar approach that combines prior layer features with next layers to boost the learning ability of the network.

It is also possible to utilize DL architectures proposed for a specific application in another problem if they are appropriately modified. Furthermore, the weights from pre-trained network can be transferred to the new architecture, and fine-tuning process can be applied to adapt the network to the new task. This is called transfer learning in the literature. The work by Jiang *et al* [11] uses transfer learning method to employ already trained network weights from fully convolutional AlexNet architecture in [90]. They resize regional patches to 500x500 from 50x50 for enlarging details in a fundus image. In this way, the architecture can catch thin vessels. The work of [95] utilizes four different networks working on retinal image patches and uses the knowledge of these networks to train the final network at the image-level.

Learned features with supervised methods reach more discriminative characteristics, which are difficult to describe analytically [41], and tend to perform better segmentation performance. Among supervised methods, deep network architectures are typically competing among themselves and achieve the-state-of-the-art performances [96]. This thesis presents fully-convolutional deep learning architecture for vessel segmentation with competitive results as its first part. Details are found in Section 4.1.

3.2 Literature Review for 3D Reconstruction

There are various 3D coronary reconstruction methods applied on data from various types of coronary imaging systems. The studies presented in [97, 98, 99, 100, 101, 102, 103, 104] utilize traditional (single-plane) XRA and [105, 106, 107, 108, 109] utilize bi-plane XRA. We focus on these XRA methods and examine them into 2 main categories as the way the technique they follow: (1) Calibrated and (2) non-calibrated (calibration-free) reconstruction. Then, we continue to present learning-based methods, but only 3D object reconstruction since there is no machine learning-based method for 3D reconstruction of coronary arteries.

In calibrated methods, camera calibration matrix is formed to map points in 3D to their corresponding 2D projection. Calibration parameters are estimated by optimizers [97, 100, 101, 102] or using the phantom object on which metal balls are attached [110]. Centerlines [100, 111, 112, 105, 98, 99] or corresponding 2D points [97, 102] are used in determining calibration parameters and estimating 3D model of coronary artery tree. [97] and [100] follow the same approach in finding external parameters of the camera (R: rotation, t: translation) using constrained nonlinear optimization algorithm given five or more correspondence points. The works [107, 101, 100, 102] use epipolar matching and triangulation methods during 3D reconstruction of coronaries after calibration. Apart from these studies, [109] utilizes bundle adjustment method and [104, 113] utilize graph-cut based sparse stereo method for 3D reconstruction. The works of [98, 99] use deformable model and update contours using active contour algorithms with back-projection external energy. Similarly, [97] proposes active contour model and [105] uses Fourier deformable model with projective external

energy. Although most of the studies assume heart as rigid during 3D reconstruction, [108, 114, 104, 97, 105] include motion caused by respiration and heartbeat into their method and obtain 4D model of coronary artery tree. In [115, 106, 101, 102, 104, 99], X-ray images are synchronized using ECG recording to ease 3D reconstruction process.

Calibration-free solution of [109] utilizes Random Sample Consensus (RANSAC) [116] method and eight-point algorithm to estimate the fundamental matrix for 3D reconstruction. In this way, the method internally achieves self-calibration in XRA image sequences. RANSAC method produces reliable correspondences during matching centerline points. The work of [103] does not require calibration and uses a non-linear optimization algorithm to estimate projection matrix directly by making some assumptions on the rotation of C-arm. The method tries to minimize the error between centerline points and back-projection of 3D reconstructed points.

As it is clearly seen that 3D reconstruction methods depend on mathematical derivation between 2D and 3D geometry by making some assumptions (pin-hole camera model). This process requires camera parameters to be known or estimated. For the ill-posed situations where camera parameters are not known, feature correspondences are utilized to achieve self-calibration [109]. However, objects can be reconstructed without a calibration step by the utilization of deep learning methods. A tremendous amount of progress has been drawn in the last 5 years in learning-based 3D reconstruction. The problem remains challenging because there is not enough 3D data for training and there are many ways to represent 3D models such as point clouds, meshes, volumetric grids (voxels), etc. The algorithm changes according to representations of 3D model. Thus, we divide 3D reconstruction literature into 4 main categories based on their 3D representations of objects as following: (1) point-cloud [117, 118, 119, 120, 121, 122], (2) volumetric grids [37, 123, 124, 125, 126, 127, 128, 129], (3) polygonal mesh [130, 131], (4) other specific data structures [132, 121, 133].

Point cloud representation is one of the most popular object representation techniques in computer graphics and computer vision. The data is simple to read, write and interpret in this format. However, the number of points should be limited if this format

is used in deep learning architectures due to pre-determined output size. CAPNet is introduced in [117] and [120] and the method uses an encoder-decoder architecture to obtain point cloud reconstructions from 2D image(s). Arbitrary views of reconstructed point clouds are projected onto 2D plane to refine point clouds with known camera parameters. Fan *et al.* [118] proposed a network, PointOutNet, for generating 3D object from a single image by obtaining its conditional shape sampler. The output shape is 1024x3 in PointOutNet. Zamorski *et al.* [119] proposed adversarial autoencoders to learn 3D latent space of compact representation of 3D point clouds and generate 3D shape out of it. In [121], a deep learning method is proposed for reconstructing 3D shapes from 2D sketches in the form of line drawings. The network produces depth and normal map of input sketches and these outputs are transformed into a dense point cloud representing a 3D reconstruction of the input sketch. In [122], neural networks are trained to produce corresponding depth maps and silhouettes from input images. From these silhouettes or depth maps 3D point cloud is generated and refined.

Volumetric grid representation is generally preferred in the related literature because the output of deep learning architectures is definite in terms of size. In this method, objects are placed into a 3D grid and grid size is constant. In [37], recurrent network-based solution is proposed for 3D object reconstruction from a single or multi-view image. The architecture, called 3D-R2N2, takes 127x127 input image and generates features from that image using auto-encoder network. The output of auto-encoder network is fed into GRU based recurrent network to hold necessary features in its internal memory. Finally, 3D reconstruction of arbitrary viewpoints of images is achieved in 3D occupancy grid (32x32x32 voxel grid) with a decoder network. Training and testing are achieved using ShapeNet [134], PASCAL VOC 2012 [135] and online 3D databases. In [127], an enhanced version of 3D-R2N2 is proposed by changing regular voxel values into weighted representation. In weighted representation, voxels are defined as integers in the range of (-53, +53) instead of 0 and 1. Neighbor voxels are taken into account with 3x3x3 kernel filter in their representation. It is claimed that training time is also improved in this way [127]. [123] and [124] proposed deep learning architecture using Octree [136] structure for

obtaining a higher resolution of a 3D model in grid voxel representation. By using Octree, 3D models can be generated efficiently in terms of memory and calculation because the architecture does not have cubic complexity for 3D objects (for instance: 512*512*512 or 256*256*256). The neural network learns Octree structure in [124]. Wang *et al*. [126] designed O-CNN, a novel octree data structure, to efficiently store the octant information occupied by 3D surfaces. The surface of objects is predicted at different scales of voxel blocks with octrees in [128] by using an encoder-decoder network. In order to incorporate the physics formations such as perspective geometry and occlusion, Markov Random Fields with ray-potentials explicitly modeled with CNN in [129]. Yan *et al.* propose an encoder-decoder network for 3D object reconstruction (output size 32*32*32) without 3D supervision by including perspective projection in the loss function [125]. In this study, the intrinsic and extrinsic parameters are known for the camera.

Polygonal mesh representation another format used mostly in computer graphics, games and animations. Wang *et al.* [130] propose to use meshes generated by deforming an ellipsoid into the desired shape from a single-color image using CNN. Distance between camera and objects are assumed to be fixed. Pontes *et al.* [131] deform mesh model by selecting a similar object from a database from a single image. Neural network is used to create latent space from this single image and output parameters are utilized in deformation to obtain 3D representation of the object in the image. A similar shape is retrieved by the classification index from 3D database, and it is deformed by a graph which is embedded to represent the final 3D object in [131].

We include other 3D representations into problem-specific part because these are mostly data-specific formats. For instance, [132] utilizes encoder decoder architecture of CNN for constructing RGB+D of a given single image view of a car or a chair. Point cloud and mesh are generated from multi-view RGB+D output of the network. They train their network with ShapeNet [134]. Lun *et al.* [121] propose a deep learning method for automatically translating hand-drawn sketches into 2D images representing surface depth and normal across several output views. Encoder-decoder network is utilized for the generation of 12 normal maps and corresponding depths. 3D point

cloud and polygonal mesh of the object are obtained fusing these depth and normal predictions later. A relatively new study presented in [133], introduces 3D representation of an object with defining 64-D vector. The vector allows objects to be predictable directly from 2D images.

3.3 The Position of Proposed Methods in the Related Literature

The overall pipeline proposed in this study contains two major steps: (1) segmentation of blood vessels and (2) reconstruction of the vessel tree from segmented images. For the first phase, we propose fully automated supervised method for blood vessel segmentation. The method utilizes a novel fully convolutional deep network architecture trained on retinal images. We follow an untraditional way while construction of the network. We first apply up-sampling and then down-sampling operations. Besides that, the network we propose has some advantages. The first advantage is that it does not include fully connected layers at the end for vessel classification. This property improves training and response time of prediction because an intensive multiplication generally exists in fully connected layers. The second advantage is that the local connectivity substantially reduced due to a larger patch size of 448x448 as input. This improves prediction time for images in contrast to smaller patch sizes. The works proposed in the literature mainly utilize patch-wise approach to train networks. Typically, smaller patches are utilized such as 27x27, 32x32 or 50x50. Our method uses a larger patch size that fit DRIVE, STARE, and CHASE_DB1 images. The third advantage is that our network continues to learn features during down-sampling and up-sampling operations in progress due to convolutions. In this way, the learning ability of the network is increased.

Most of the solutions proposed in the literature for 3D reconstruction pipeline include camera calibration process. With the development of learning-based methods, this step can be eliminated. In literature review provided in Section 3.2, there is no learning-based method for 3D reconstruction of coronary arteries. There are several ways to include 3D reconstruction process in deep learning architectures due to the variation of 3D representation of objects. The methods mostly used in the literature are based on point cloud, voxel grid solutions and mesh generation with deformation parameters.

Since our data relatively have a definite shape, we define our structure to represent 3D coronary vessels. We use consecutive points to represent a continuous segment of cylindrical shapes and construct multiple segments to form vessels. Therefore, our method falls into problem-specific 3D reconstruction of objects.

CHAPTER 4

MATERIALS AND METHODS

We first segment blood vessels in angiogram images. However, public data-set for XRA is not available. Thus, we adapted retinal images for training of our network. Next, we use the segmented images in a 3D reconstruction network to produce 3D model of the coronary tree. In this section, we present proposed methods for blood vessel segmentation and 3D reconstruction, respectively.

4.1 Blood Vessel Segmentation

Deep learning architectures proposed in literature has demonstrated their superiority in automatically learning of rich hierarchical representations. The main disadvantage of utilizing deep learning architecture is that a large number of labeled training data and computational power is needed to achieve the-state-of-the-art performance. For our case, we do not have any labeled data for X-Ray angiogram images. Instead of creating labeled data from scratch, we adapted publicly available datasets containing other forms of vessels, namely retinal blood vessels. We propose a novel fully CNN architecture achieving the-state-of-the-art segmentation performance and introduce publicly available retina databases, the core components of the method, preprocessing and data augmentation techniques, loss function, training procedure, and finally performance comparison measures in this section.

4.1.1. Retinal Blood Vessel Databases

To the best of our knowledge 9 publicly available ophthalmological databases exist. However; only 4 of them provide ground-truth images which are annotated by experts, namely DRIVE (Digital Retinal Images for Vessel Extraction) [52], STARE (STructured Analysis of the Retina) [137], CHASE_DB1 (Child Heart and Health Study) [138] and HRF (High-Resolution Fundus) [139] as listed in Table 4.1. DRIVE, STARE and CHASE_DB1 databases are commonly preferred in the literature because they are all decent in quality. HRF database is mostly ignored in vessel segmentation

literature because it is relatively new, and the resolution of images is approximately 4 times higher than DRIVE and STARE databases.

Table 4.1 Summary of publicly available retina databases.

Database Name	# of Labeled Images	Additional Information
DRIVE	40 (20 Training + 20 Test)	Digital Retinal Images for Vessel Extraction. Consists of 40 color retinal images. Test-set images are labeled by 2 experts.
STARE	20	Structured Analysis of the Retina. Consists of 400 retinal color images, 20 images have ground-truth labels from 2 experts. 9 images belong to healthy, 11 images show diseases.
CHASE_DB1	28	Child Heart and Health Study in England. 14 children (left and right eye) total images of 28. Good quality and good contrast.
HRF	45	High-Resolution Fundus. Total number of images: 45 (15 healthy subjects, 15 images of diabetic retinopathy and 15 images of glaucomatous.).
DIARETDB1	-	Standard Diabetic Retinopathy Database. Total number of images: 89.
REVIEW	-	Retinal Vessel Image set for Estimation of Widths, Total number of images: 16
ROC	-	Retinopathy Online Challenge. Total number of images: 100.
VICAVR	-	The VICAVR database. Total number of images: 58
MESSIDOR	-	The Messidor DB. Total number of images:1200.

The **DRIVE database** [52] is obtained from a diabetic retinopathy screening program in Netherlands and contains 40 labeled retinal images of size 565x584. 33 images belong to healthy subjects while 7 images have pathology (early mild disease). The labeled images are randomly selected from 400 diabetic subjects between 25 and 90 years of age. The data-set is already divided into test and training set consisting of 20 images each. All images are colorful and 8-bit resolution per channel. Test set is annotated by 2 experts. An image randomly selected from DRIVE test set is presented in Figure 4.1.

| (a) | (b) | (c) |
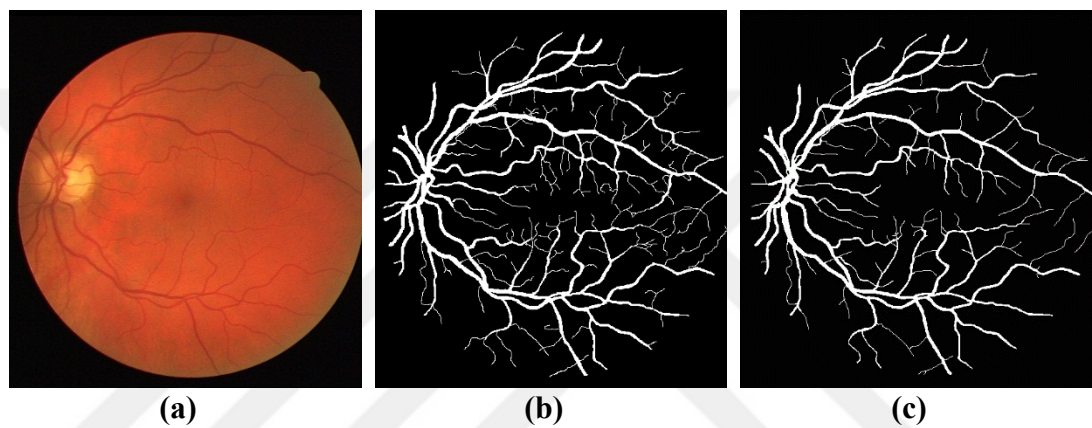
Figure 4.1 (a) An image from DRIVE database (05_test.tif), (b) shows annotation from expert #1 and (c) shows annotation from expert #2.

The **STARE database** [137] is the part of the project named STARE (STructured Analysis of the Retina) at the University of California funded by the U.S. National Institutes of Health. STARE data-set contains 400 images with size of 700x605 and only 20 images have been manually labeled by 2 experts. 11 images show pathology, 9 images belong to healthy subjects. The labels marked by the second expert show thinner vessels than the first experts. STARE database is the most challenging database among all the others because some of those pathological images also suffer from decreased sharpness and deterioration due to eye illnesses (see Figure 4.2).
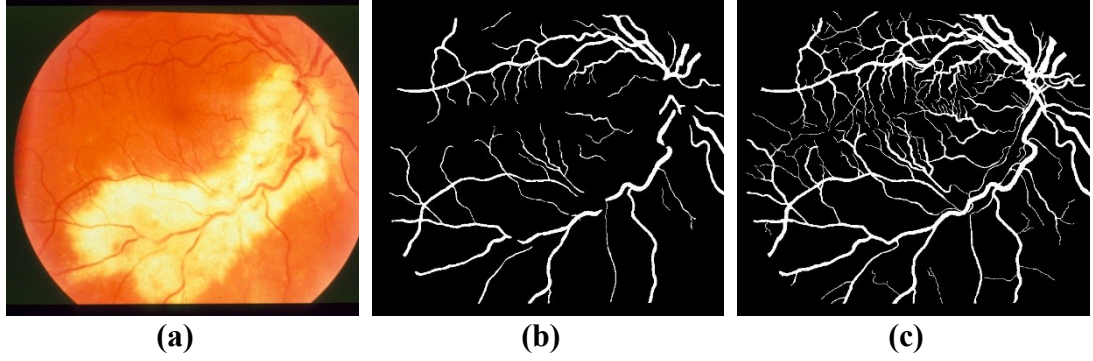
Figure 4.2 (a) An image from STARE database (im#0044) containing pathology. (b) and (c) show corresponding labeled images of (a) from experts "ah" and "vk", respectively.

The **CHASE_DB1 database** [138] is part of CHASE (Child Heart and Health Study in England) project and contains 28 images which are collected from left and right eyes of 14 international children. The images are of size 999x960 and all in good quality and contrast with no symptoms. All the images are captured using a fixation target and focused on optic disc to be centered. Manual segmentation for each image is achieved by 2 experts. This database is important because it contains retinal images of multi-ethnic children. An example image from CHASE_DB1 is demonstrated in Figure 4.3.
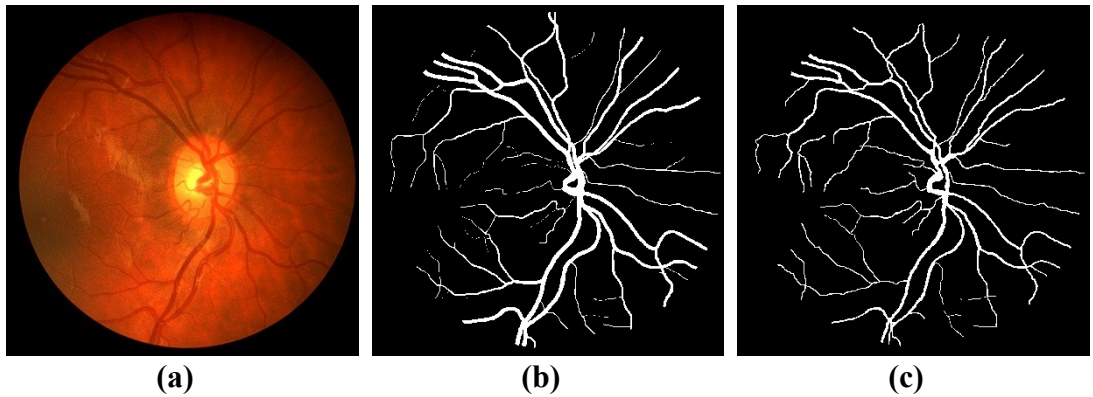
Figure 4.3 (a) delineates retinal image from CHASE_DB1 (Image_03R). (b) and (c) are labeled images of expert #1 and expert #2, respectively.

The statistical details of the retina databases used in this thesis are summarized in Table 4.2. We train our network using DRIVE, STARE and CHASE_DB1 individually. To

generalize blood vessel segmentation further, we also train the model with a combination of all, called Hybrid-DB. In all databases, we utilize labeling from expert #1 as the gold standard (ground-truth).

Table 4.2 The statistical properties of DRIVE, STARE and CHASE_DB1 databases. There is no field of view (FOV) distinction, all pixels in images are considered in calculations.

Properties of databases	DRIVE		STARE	CHASE_DB1
	Train	Test		
Number of images	20	20	20	28
Image dimensions	565x584x3	565x584x3	700x600x3	999x960x3
Number of positive pixels	569615	577945	644053	1861974
Number of negative pixels	6029585	6021255	7825947	24991146
Positive pixel in percentage ≈	0.0863	0.0876	0.0760	0.0693
Negative pixel in percentage ≈	0.9137	0.9124	0.9240	0.9307

4.1.2. Data Augmentation Technique

The DRIVE database is already divided into training and test sets which include 20 images each as seen in Table 4.2. The STARE database has 20 labeled images and we divide the first 70% and last 30% of images for training and test sets, respectively. Finally, CHASE_DB1 has 28 labeled images and we follow the same division procedure as in the STARE database.

Deep learning architectures need plenty of data during training process for a better generalization, but available public retina databases do not have sufficient training images, therefore the network over-fits training data-sets in such cases. In order to overcome over-fitting, we apply rotation, translation and mirroring operations to

images for data augmentation. Rotation is applied according to the center of an image in the range of $-45° < \theta < 45°$. Then translation is applied randomly in the interval defined in (4.1) and (4.2).

$$t_x < w - 448, \tag{4.1}$$

$$t_y < h - 448 \tag{4.2}$$

where t_x and t_y represent translation in x and y directions, respectively. w and h are the image width and height. 448 is the input size of the proposed network in this study. After one random image is generated using this procedure, we augment it to 8 with definite rotations (-90°, 90°) with respect to image center and flipping operations as demonstrated in Figure 4.4. We randomly generate 10000 images using train-set for each database using this technique. For databases which do not include training-set, we apply the division procedure mentioned at the beginning of this section.

In data augmentation technique shown in Figure 4.4, we contribute to network generalization in blood vessel segmentation with random rotations and flipping operations. In this technique, we increase the possibility of seeing different vessel forms that may be encountered in other imaging modalities as well. Namely, the network can be trained for every direction of vessels and increases the capability to distinguish vessel-like structures for different orientations. This approach makes the network to generalize on blood vessel segmentation.
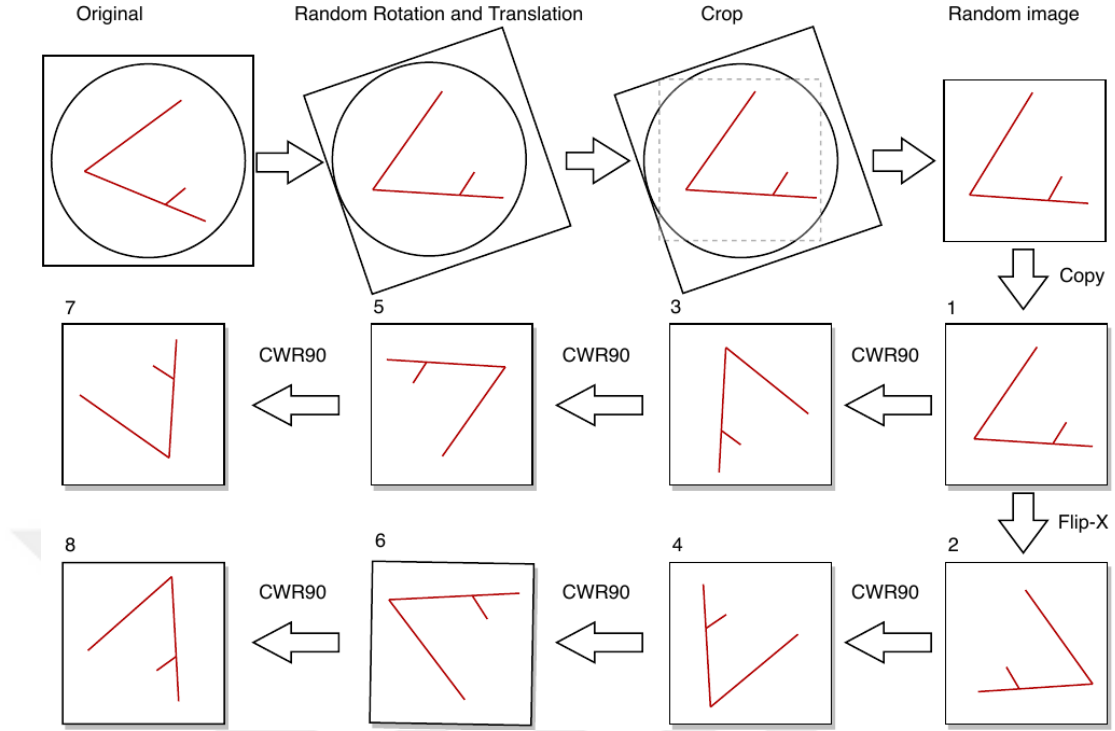
Figure 4.4 The illustration of data augmentation algorithm. By using this method, 8 images are generated from a single random image. The first row delineates how original image is turned into a random image using random rotation and translation within the ranges defined in section 4.1.2. Last two rows show how the random image is augmented using clock-wise-rotation (CWR) and flipping in X-axis technique to generate different shapes of vessels.

4.1.3. Preprocessing of Images

Although deep learning architectures perform well on raw input data, appropriate preprocessing operations may improve segmentation performance [86]. Thus, we conduct tests with and without preprocessing to clarify the effect of preprocessing. Since some images have illumination problems such as darker or lighter background, we exploit contrast limited adaptive histogram equalization (CLAHE) algorithm [140] to improve contrast in images. We also employ multi-scale top hat transform (MTHT) [141] to enhance image contrast and details further. The preprocessing effect is visualized for one of DRIVE database images in Figure 4.5. The disk size range for MTHT is (3, 19), and CLAHE clip limit is 2.0.
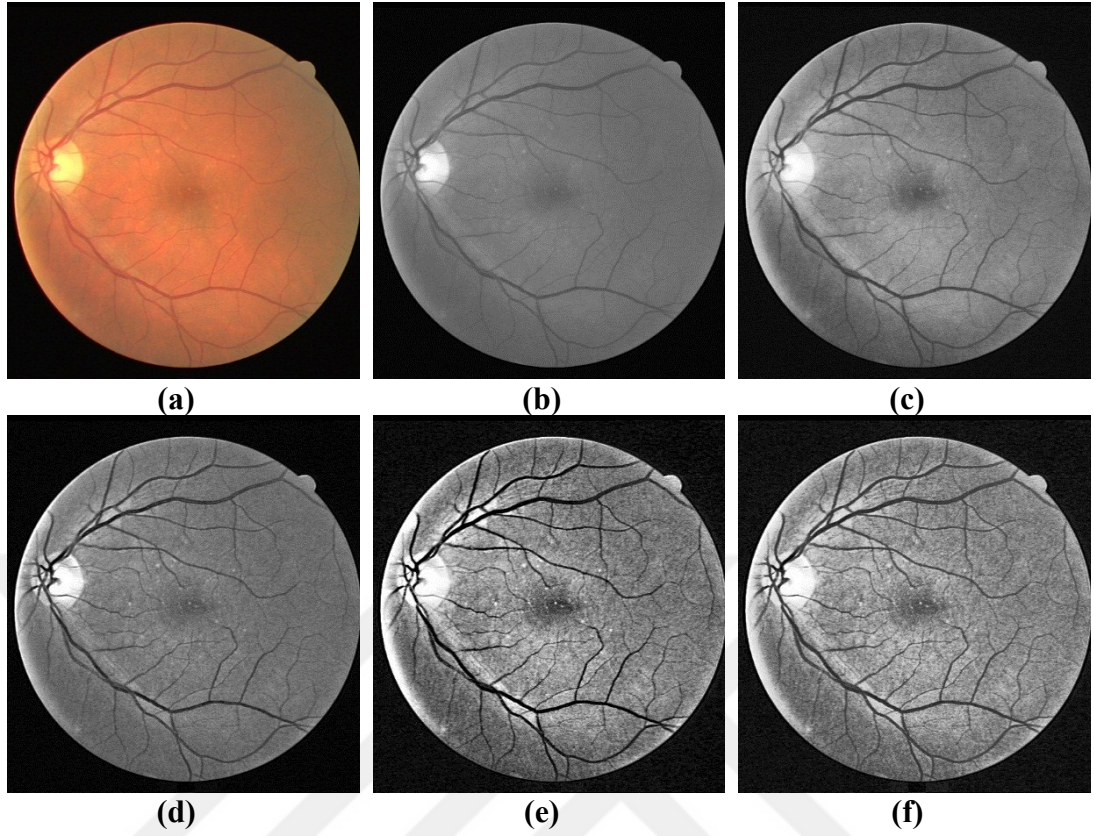
Figure 4.5 The effect of pre-processing on DRIVE image #21. (a) shows original image, (b) shows grayscale of (a), and (c) and (d) show CLAHE and MTHT pre-processed images of (b), respectively. (e) is MTHT followed by CLAHE, (f) is CLAHE followed by MTHT. We used (e) due to high contrast in the training of the network. CLAHE clip-limit is set to 2.0, and the range for MTHT disk size is (3, 19).

The order of preprocessing is determined by looking at multiple images in a window like Figure 4.5. We try to observe the enhancements visually in all images and conclude that the preprocessing operation of MTHT followed by CLAHE makes vessels more distinguishable from its background. We follow this order in our experiments for training with pre-processed.

We visualize the effect of pre-processing on angiogram images as well in Figure 4.6. We use XRA images from [142]. The same parameters are utilized in pre-processing as in Figure 4.5.

(a)   **(b)**   **(c)**   **(d)**
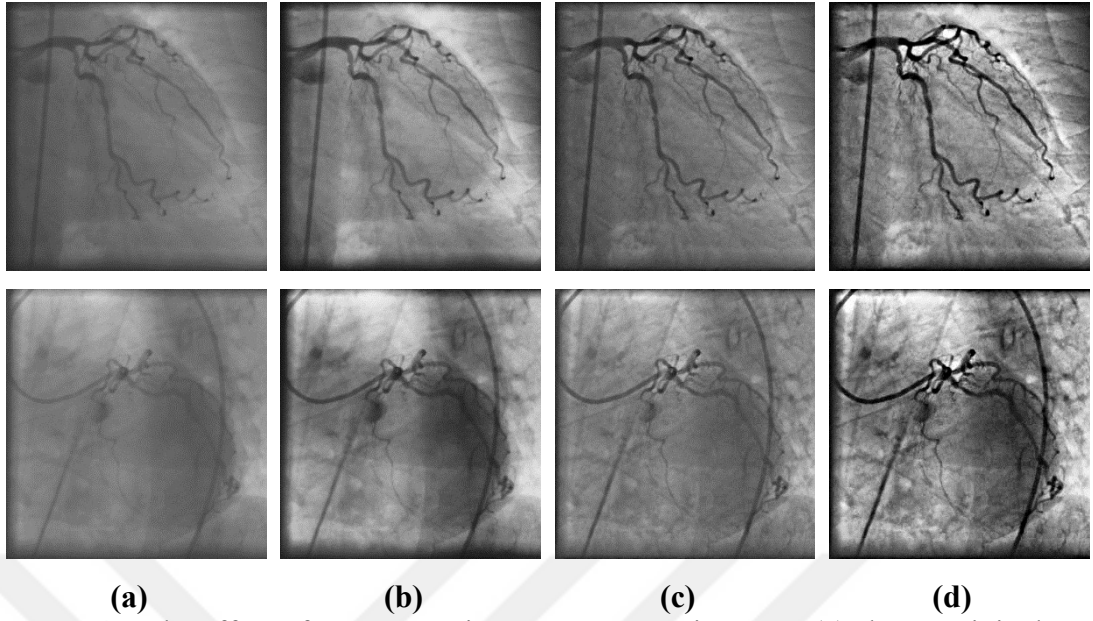
Figure 4.6 The effect of pre-processing on X-Ray angiograms. (a) shows original X-Ray image, (b) and (c) show CLAHE and MTHT pre-processed images of (a). (d) is MTHT followed by CLAHE.

Retinal images are colored, but XRA images are in grayscale. The main goal of the first part of this thesis is to segment coronary arteries in XRA images to feed them into 3D reconstruction network to obtain corresponding 3D model. Therefore, we convert retinal images into grayscale. Using grayscale image at the input provides 2 advantages. The first advantage is that our network can generalize blood vessel segmentation so that it can alternatively be applied to other medical image modalities which are in grayscale. The other advantage is that the complexity of the network is decreased by using a single channel at the input. Correspondingly, less computation is achieved for each segmentation.

Normalizing input values minimizes bias within neural network and speed-up neural network training [143]. It becomes useful, especially when data is of different scales [143]. Therefore, we normalize input images using (4.3) and scale between 0 and 1 just before feeding them to the proposed architecture in both with and without preprocessing tests.

$$n_i = \frac{x - \mu_i}{\sigma_i} \tag{4.3}$$

where n_i is normalized image at i, μ_i is mean and σ_i is standard deviation of image at i in the training set.

4.1.4. Deep Learning Architecture

We propose a novel FCNN architecture and name the architecture as Sine-Net because in an abstract view from the top, it resembles a sine-wave as illustrated in Figure 4.7. We prefer the architecture to be fully convolutional because this structure shows superior performance for natural image segmentation [90]. We also want the learning process to be continued during up and down-sampling operations are in progress and prefer to place layers involving convolution operation instead of max-pooling and up-convolution layers. The architecture is novel since we first apply up-sampling and then down-sampling layers for catching thin and thick vessel features, respectively. However; the trend for creating a CNN model is just the opposite, i.e. down-sampling operation comes first. We also include residuals to pass mores features to the deeper level of the architecture from lower levels. In this way, the network can recover features which may be lost in up-sampling or down-sampling process.

The Sine-Net depicted in Figure 4.7 contains 15 convolutional layers with filter size of 3x3 for each. Rectified Linear Unit (*ReLU*) is used as an activation function after each convolution except the final layer. Final layer activation is *tanh* because predictions should be in the range between 0 and 1 (non-linear). Since the previous layer of final activation is ReLU, the input to *tanh* is either zero or some positive value which makes the output in range 0 and 1. Black arrows in Figure 4.7 indicate transferring feature maps from intermediate layers to the deeper layers of the architecture. Green arrows show up-sampling operation with transpose convolution and red arrows show down-sampling with convolution of a stride of 2. In this way, the network continues to learn hierarchical representations by updating weights which do

not exist in max-pooling or up-sampling layers. The number of channels is doubled and halved after each down-sampling and up-sampling operation, respectively.

Sine-Net provides some advantages due to its construction style. The first advantage is that the architecture produces the final prediction using only convolutional layers. It does not contain a dense layer (fully connected layer) as many classification architectures typically have. This construction style decreases the size of the network, improves training and prediction time because an intensive computation generally exists in dense layers. The second advantage is that Sine-Net accepts larger input size in contrast to many architectures proposed in the literature. They use smaller patches for increasing the number of training data. However, the selection of larger patch sizes (448x448) reduces local connectivity substantially and speeds up training and prediction. Increasing patch size continuously is not feasible because networks tend to decrease its segmentation performance and need more memory. In [11], the patch size of 500x500 has performed better results than 400x400 and 600x600 in the training phase. The third advantage is that our network uses convolutional operations in down-sampling and up-sampling. Normally, down-sampling is achieved using max-pooling and up-sampling is achieved using copy of the same data. We prefer strided convolution for down-sampling and transpose-convolution for up-sampling operations. In this way, the learning ability of the network is boosted because the network continues to learn features (updates weights) during these operations.
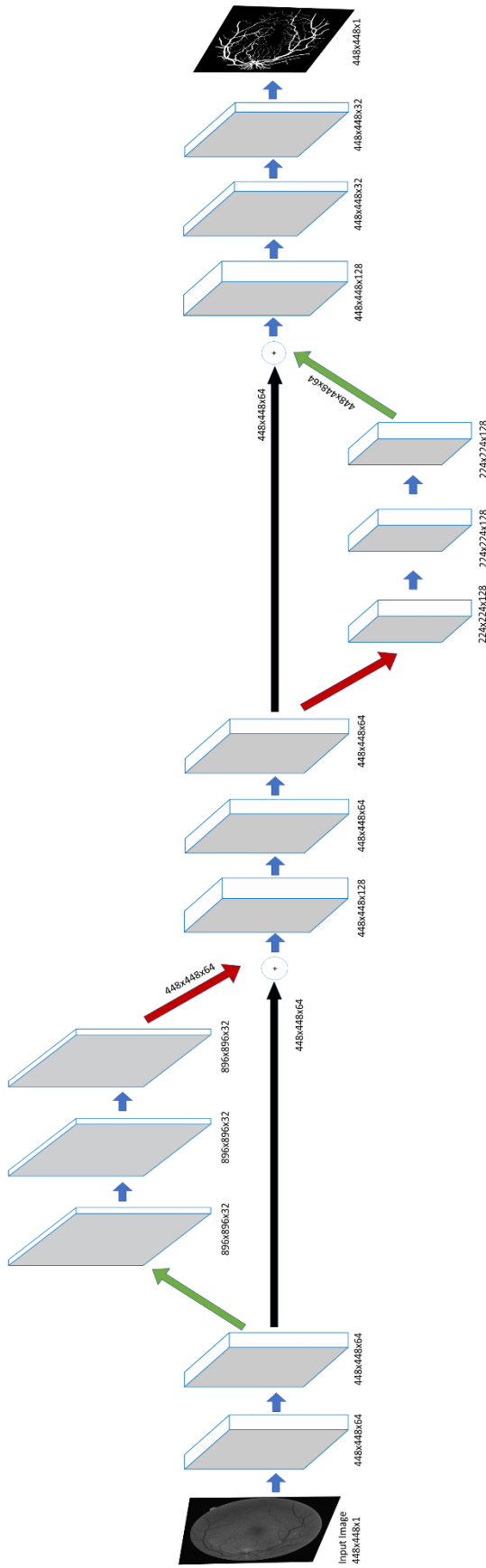
Figure 4.7 Proposed Sine-Net architecture for blood vessel segmentation. Blue arrows indicate convolution operation, green arrows indicate up-sampling with transpose-convolution and red arrows indicate down-sampling with strided (step=2) convolution operation. Convolution filter size is 3x3, and activation functions are ReLU for each layer except the last. It is tanh. Finally, black arrows represent residuals which transfer lower level feature maps to the deeper level of the network.

4.1.5. Loss Function

Loss function represents the cost for inaccurate predictions made in classification/regression problems. There are many loss functions defined in machine learning to train networks such as mean squared error (MSE), categorical cross-entropy, binary cross-entropy, cosine proximity, log-cosh, etc. The output of the proposed network is a binary classifier, indicating 0 as background pixel and 1 as the vessel. For training of our network, we use binary cross-entropy defined as:

$$L_{CE}(\hat{y}, y) = -\sum_k [y_k \, log \, \hat{y}_k + (1 - y_k) \, log(1 - \hat{y}_k)] \qquad (4.4)$$

where $\hat{y}$ and y corresponds to predicted and ground-truth values respectively. k represents k^{th} sample in the training set.

4.1.6. Training

Several trainings with different parameters and datasets are performed on the proposed network and we have created a table in order to clearly show and easily understand the training details. We utilized Keras [144] library running on top of TensorFlow for training each data-set given in Table 4.3.

There are eight training processes in total for the first part. Each training contains 50 epochs with Stochastic Gradient Descent (SGD) followed by 20 epochs with Adam [145] optimizer. This kind of approach improves generalization performance in training process as presented in [146]. In each epoch, training set is shuffled and 10% is used for validation. The learning rate for SGD optimizer is 10^{-4}. The batch size is set to 10. All training processes are conducted on a Linux server with Xeon(R) CPU E5-2667v4 (3.20GHz) CPU, 128 GB of RAM and NVIDIA Tesla P100 Graphics Processing Unit (GPU). The training time for each data-set is noted in Table 4.3.

Table 4.3 The details of the training for data-sets used in this study. Training time corresponds to overall training time for 70 epochs (20 Adam + 50 SGD).

Training name	Dataset included	Number of images	Preprocessing (CLAHE+MTHT)	Training Time
DRIVE	DRIVE	10000	NO	13 hours
CHASE	CHASE_DB1	10000	NO	13 hours
STARE	STARE	10000	NO	13 hours
HYBRID	DRIVE+STARE+CHASE_DB1	30000	NO	41 hours
DRIVEPP	DRIVE	10000	YES	13 hours
CHASEPP	CHASE_DB1	10000	YES	13 hours
STAREPP	STARE	10000	YES	13 hours
HYBRIDPP	DRIVE+STARE+CHASE_DB1	30000	YES	41 hours

4.1.7. Performance Measure for Segmentation

Accuracy is defined as the ratio of correctly classified pixels to all pixels' predictions. It is alone not enough for fair performance comparison due to unbalanced classes. Hence, studies include sensitivity and specificity measures. Sensitivity and specificity are statistical measures of how a model is successful in detecting true and false cases. Specifically, sensitivity is a measure of correctly classified vessel pixels, and specificity is a measure of correctly classified non-vessel pixels for our case. In addition to these metrics, AUC is a powerful measure in machine learning to test binary classifiers by indicating how much a model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0's and 1's correctly. Accuracy, sensitivity and specificity are calculated using (4.5-4.7), and AUC is the area under Receiver Operating Characteristic (ROC) curve.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \qquad (4.5)$$

$$Sens. = \frac{TP}{TP + FN} \qquad (4.6)$$

$$Spec. = \frac{TN}{TN + FP} \qquad (4.7)$$

The performance comparison of each training is accomplished with four metrics as described here; namely, sensitivity, accuracy, specificity and AUC. The metrics are evaluated for every single image in each database's test-set. Instead of giving individual results, we prefer to share all results in terms of mean ($\bar{x}$) and standard deviation (σ) for interpreting them clearly and effectively. Section 5.1 presents all results for our network and compares them with the-state-of-art methods proposed in the related literature.

4.2 3D Reconstruction

Three-dimensional reconstruction itself is a challenging problem due to various requirements and constraints. The algorithms proposed in the literature follow similar analytical methods like finding correspondences between images, estimating camera parameters, using epipolar constraints, etc. There are so many parameters if we dive into the problem. Deep learning architectures with appropriate data structures are now emerged and becomes trending. As the availability of 3D model of objects increases, DL architectures can reconstruct the object even a single image is introduced. It eliminates parameters that analytical methods use. We have provided details of methods covered in the related literature in section 3.2. In this section, we present our 3D reconstruction methods along with the proposed data structure for tubular shapes. We also introduce the generation of our synthetic coronary artery tree database and corresponding multi-view images here.

4.2.1. Tubular Shape Representation

There are various ways to represent objects in 3D and 3D representation matters in 3D machine learning. A data structure compatible with machine learning can improve

reconstruction performance of networks. Thus, we define a new data structure for holding tubular shapes specifically. The center of circles and its radiuses are used to represent tubular shapes. In this structure, center points must come sequentially to obtain the path for a continuous tubular shape. In this way, we can find the direction of each consecutive circle using (4.8).

$$\vec{n} = \frac{\overrightarrow{C_i C_{i+1}}}{\left\| \overrightarrow{C_i C_{i+1}} \right\|} = \frac{(C_{i+1} - C_i)}{\left\| (C_{i+1} - C_i) \right\|} \tag{4.8}$$

where C_i is the ith center point, $\vec{n}$ is unit normal showing the direction C_i to C_{i+1}. The illustration given in Figure 4.9 delineates how we define data structure. Tubular shaped objects can have many continuous segments. Thus, we ended each segment with a negative radius value to indicate this point is the last center point for the corresponding tubular segment. Bifurcation points for branching can start anywhere from one of the continuous segments. Each segment forms a continuous vessel on its own. An example representation of one segment is as follows:
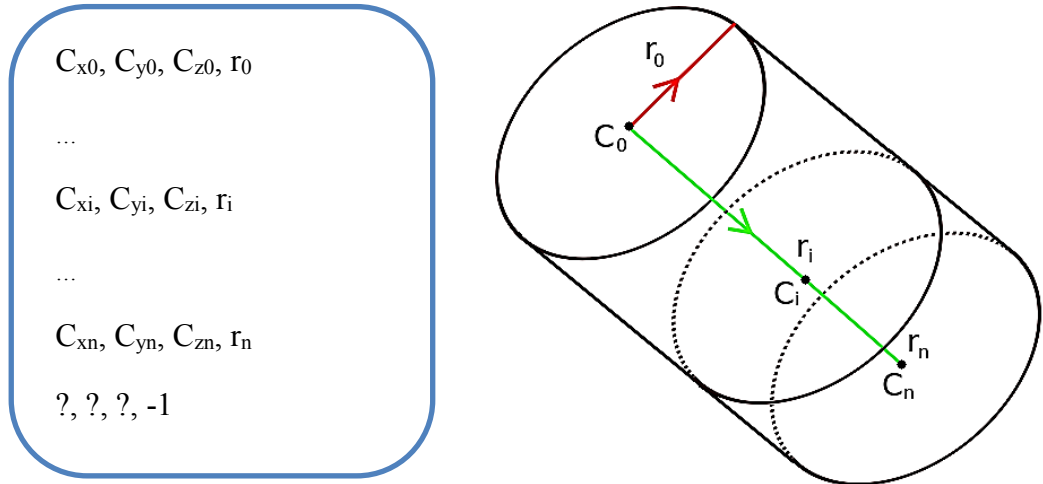
Figure 4.8 Tubular shape representation for one segment. C_x, C_y, C_z are the coordinates of center points in 3D space and r is the radius of a tubular object at that point.

The current segment shown in Figure 4.8 starts with its center point (C_0) in 3D space and ends with the corresponding radius of the circle centered at C_0. We assume that this segment is continuous until the end of the segment sign (?, ?, ?, -1) appears. We represent the end of the segment with ?, ?, ? for indicating any value for center point and a negative value (it is -1 in the example) for radius. The direction of tubular segments can easily be derived from the continuity assumption in which center points must come in order. Figure 4.9 illustrates how direction (unit normal vector) is derived from consecutive center points.
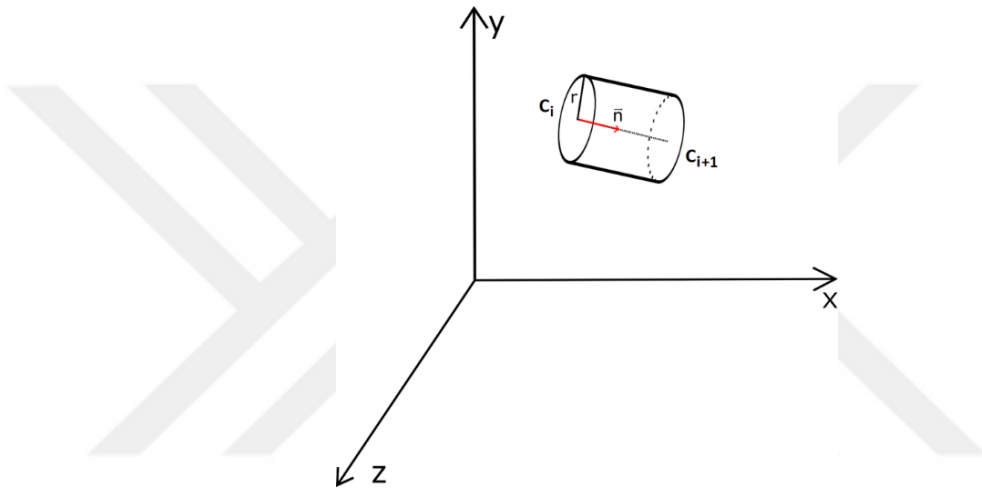
Figure 4.9 Tubular shapes are represented with circles coming in order. The center point (X, Y, Z) of each circle and its radius is written sequentially. The unit normal vector to the circle surface can be obtained using (4.8).

In order to recover tubular shape, we need to sample points around centers with a given radius as in Figure 4.10. $\vec{u}$ and $\vec{v}$ can be derived with a little effort using (4.9), (4.10), (4.11) and (4.12) together.

$$\vec{n} = \langle x, y, z \rangle \tag{4.9}$$

$$\vec{t} = \langle n_y, -n_x, 0 \rangle \tag{4.10}$$

$$\vec{u} = \vec{n} \times \vec{t} \tag{4.11}$$

$$\vec{v} = \vec{n} \times \vec{u} \tag{4.12}$$

where n_x and n_y are unit normal values in X and Y coordinates, respectively. The last step for sampling is to parametrize the circle using vectors around each center point C_i. In our test, we sampled each circle with 20 points. If desired, this can be increased for a better resolution of tubular shapes. Parametrization over angle θ can be achieved as:

$$p_{s_i} = (r * \cos\theta)\,\vec{u} + (r * \sin\theta)\vec{v} + C_i \tag{4.13}$$

where p_{s_i} is an array of sampled points for $0 \leq \theta \leq 2\pi$ with a step of $2\pi/20$ and C_i is i^{th} center point.
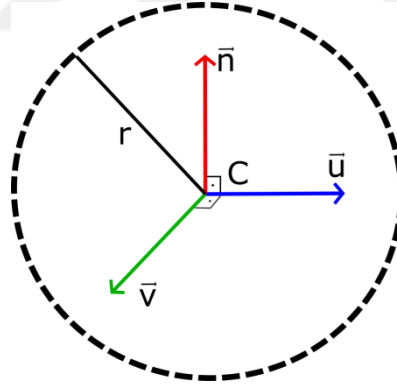
Figure 4.10 Sampling points around centers with a given radius of r. *n* is the unit normal to the surface, *u* and *v* are perpendicular unit vectors lie on the surface of the circle.

4.2.2. 3D Data Augmentation

Deep network architectures need plenty of labeled data for training in order to create input and output relations. If the number of elements in the training set is lower, the

network memorizes input-output relation, called overfitting. In this case, the network performance (the quality of output) degrades in tests. To generalize network such that it produces better results, training should be done using a relatively large data-set. Thus, we generate 10000 3D models randomly from real coronary data from [147]. Only centerlines and corresponding radiuses are shared with us. Two of them are visualized using ParaView[1] tool in Figure 4.11.

(a)

(b)

(c)

(d)
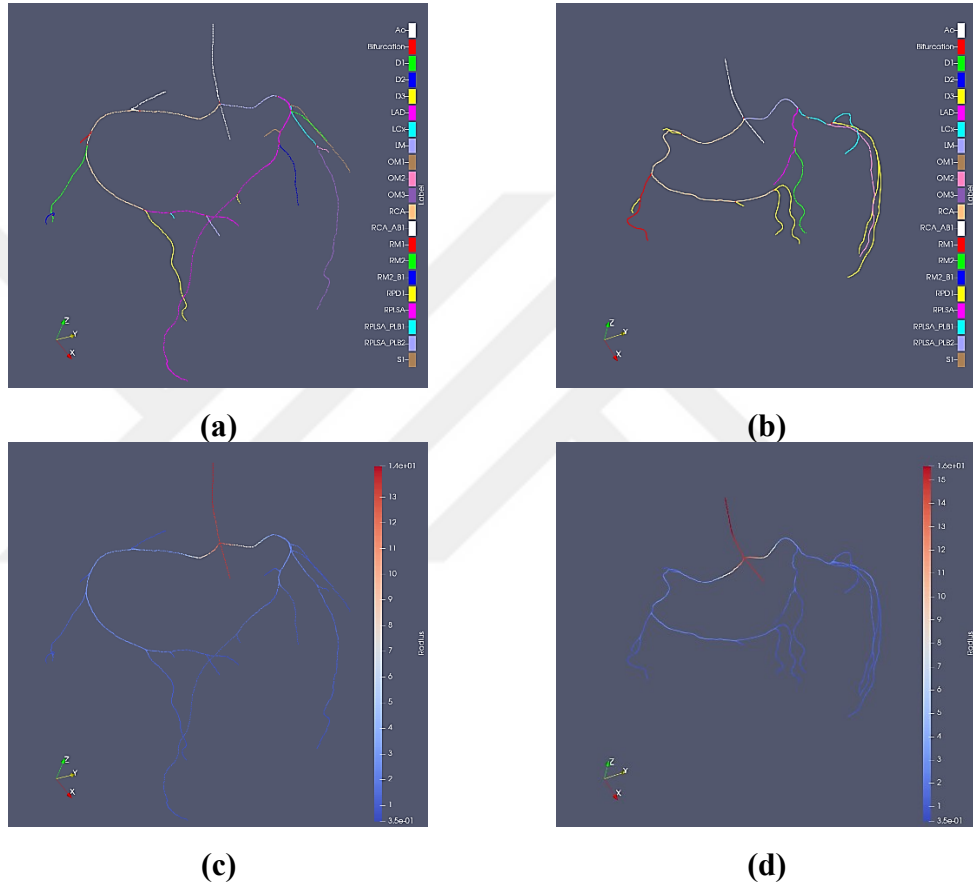
Figure 4.11 Coronary tree visualization of two subjects in [147]. (a) and (b) show centerlines and their labels including branches, (c) and (d) are the radius of each point of (a) and (b), respectively. The units are in millimeters.

RCA and LCA are given and connected to aorta in each patient data as seen in Figure 4.11. X-Ray angiogram is applied mostly one side either RCA or LCA which means there is one tree in X-Ray angiograms. Thus, we separate RCA and LCA in given 3D

---

https://www.paraview.org/[1]

models. In this way, our network can specifically learn branches of RCA and LCA separately. The other advantage of separating these main branches is that the output size of the network will be smaller. This makes the network simpler and leads to less calculation for each prediction. Later the combined version can also be developed if needed.

We do not utilize the actual 3D vessel data from [147]. We create random vessels by selecting one of the corresponding branches from 9 subjects. Additionally, some of the branches follow a similar path and we add them to the same group to obtain a random pool. We determine similarities by looking at visually multiple samples of branches. Vessels with similar patterns are gathered into the same group. Example branching pattern for D1, D2, RPD1 and RPD2 from 9 subjects can be examined in Figure 4.12.

D1 branches on LCA   RPD1 branches on RCA

D2 branches on LCA   RPD2 branches on RCA
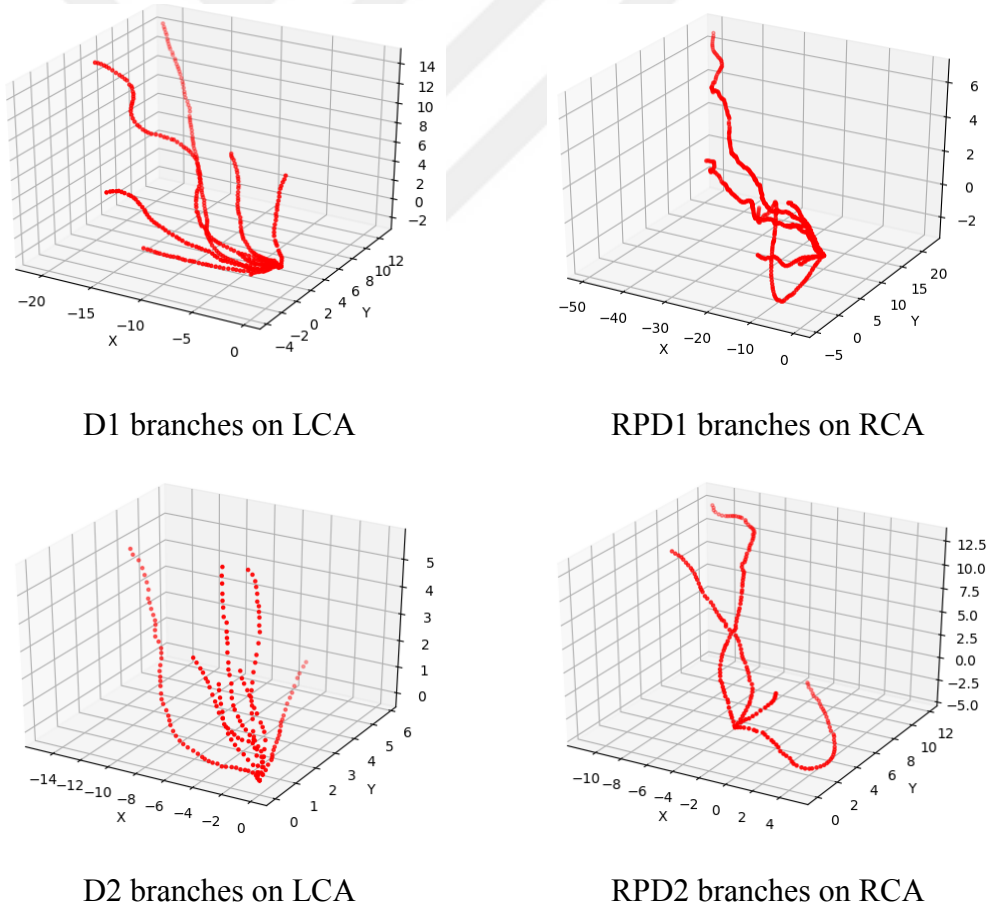
Figure 4.12 Branching pattern of similar vessel parts in a coronary artery tree.

A similarity table is obtained for the construction of different coronary arterial trees. While generating RCA or LCA, we randomly determine the branching pattern from the pool we created. Table 4.4 summarizes possible alternatives to coronary branches. We obtain this table by creating multiple visualizations of vessel patterns as in Figure 4.12 and group similar patterns to create a random pool for their alternatives.

Table 4.4 Possible branches of coronary arteries that can be replaced with the actual branching pattern. This table is obtained experimentally by visualizing vessel patterns shown in Figure 4.12.

ACTUAL LABEL	POSSIBLE LABELS
RCA	Self
CONUSA1	Self, RCA_AB1[shorter 1/3], -RPD2 [shorter 1/5]
CONUSA1_B1	Self, RPD2_B1
RCA_AB1	Self, ConusA1, -RPD2[shorter 1/5]
RM1	Self, RM2, RM3
RM1_B1	Self, D1_B1 (x, -y, -z), S3(-x, y, z), D1(x, y, -z) [shorter1/3], D2(x, y, -z)
RM2	Self, RM1, RM3
RM2_B1	Self, RM1_B1, RM_B2
RM3	Self, RM1, RM2
RM3_B1	Self, RM1_B1 and its following
RPD1	Self, RPD2, RPD3(x, -y, z)
RPD2	Self, RPD1, RPD3(x, -y, z)
RPDN_B1	Self, RPLSA [shorter1/10]
RPD3	Self, RPD1 (x, -y, z), RPD2(x, -y, z)
RPLSA	Self, RPD2_B1
RPLSA_PLB1	Self
RPLSA_PLB2	Self
R_AVNA	Self

We add more randomization to the vessel generating software. There is a possibility to deviate 3D coordinates of centerline points and their corresponding radiuses. Some branches are also determined to be added in the coronary tree with an 80% possibility. We also add the possibility to change the distance between two centerline points. In this way, we can generate various coronary vessel forms that may resemble real coronaries. It is a very low possibility to generate the same vessel tree in this configuration. Some example outputs of vessel generating software are illustrated in Figure 4.13.

(a)

(b)

(c)

(d)
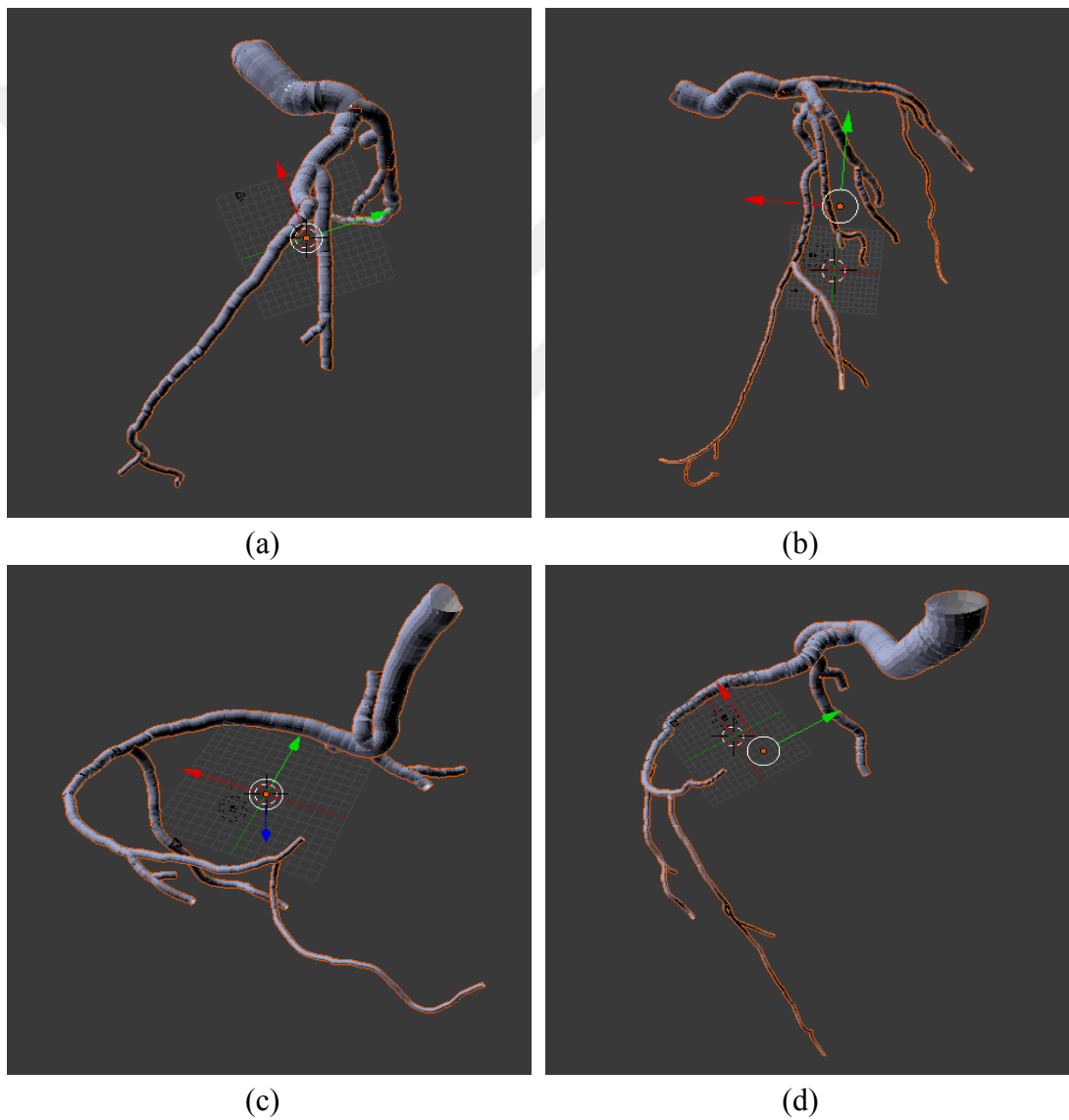
Figure 4.13 Random coronary vessel generations. The top row shows LCA, the bottom row shows RCA.

4.2.3. Multi-view Image Generation

The main task of this study is to reconstruct 3D coronaries from 2D X-Ray angiograms. We have generated 3D coronaries synthetically with the method mentioned in Section 4.2.2. The next step is to produce multi-view images of coronaries. We have created a tool for taking multiple images from different views of angle like in C-arm. We simulate 2 axis rotations as in Figure 4.14. The coordinates of the camera after rotations are calculated using spherical coordinates $p(x, y, z) = (r, \theta, \varphi)$.
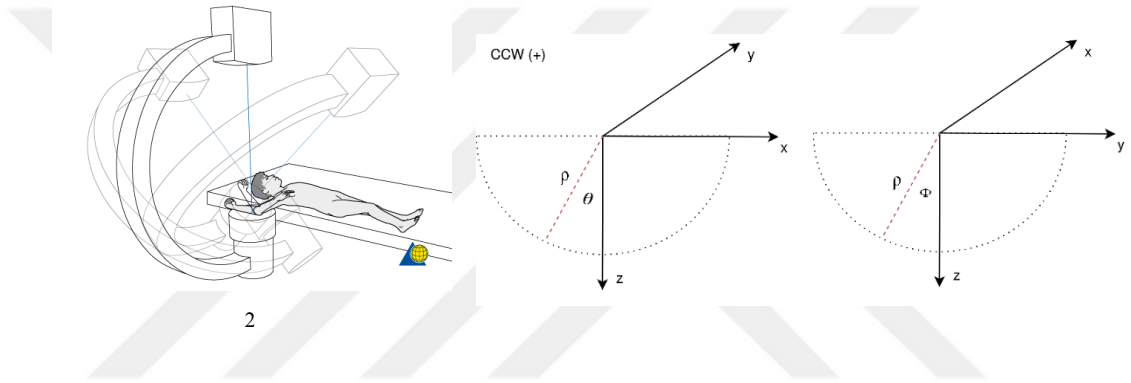
Figure 4.14 C-Arm rotation axis simulation. Multi-view coronary artery images are generated using this principle.

If camera is rotated around y-axis (θ rotation), x and z coordinates will be updated. If camera is rotated around x-axis (φ rotation), y and z coordinates will be changed. The new coordinates after rotation become:

$$C_x = \rho sin(\theta) \tag{4.14}$$

$$C_y = \rho sin(\theta) \tag{4.15}$$

$$C_z = \rho \sin(\theta) * cos(\varphi) \tag{4.16}$$

where ρ is the distance between the camera and the geometric center of objects (coronaries). Figure 4.15 demonstrates multi-view images produced by the tool we developed using Blender software[3].

Theta = -30, phi = 30, r=10   Theta = -70, phi = 30, r=10

Theta = 30, phi = -10, r=10   Theta = -10, phi = -30, r=10
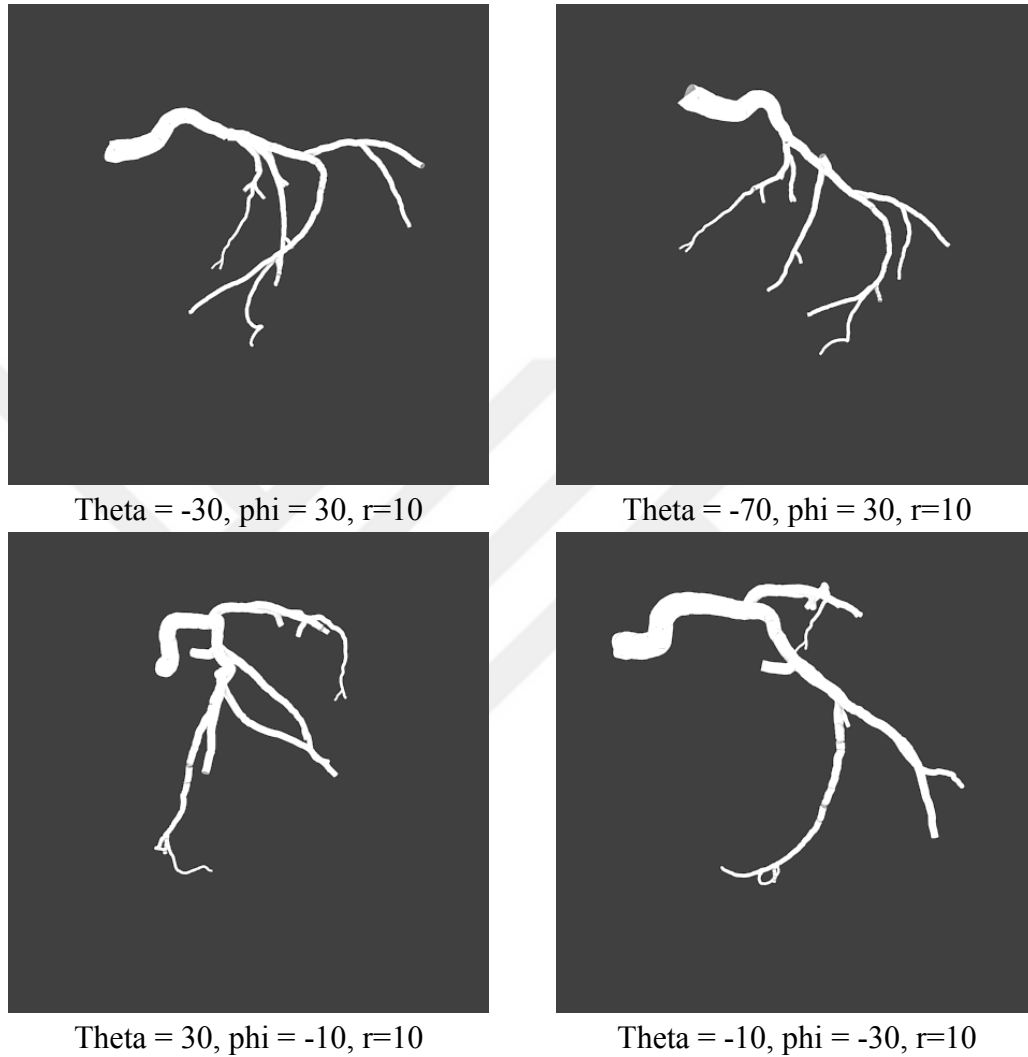
Figure 4.15 Multi-view image generation by imitating C-arm for some of coronary arteries in the data-set. Angles are in degrees. r is the distance between the camera and the object center.

---

[3] www.blender.org

4.2.4. 3D Reconstruction Architectures

We propose 4 deep learning-based architectures for 3D reconstruction: (1) Multi-view fully CNN, (2) Time distributed auto-encoder CNN and (3) Auto-encoder followed by LSTM and (4) GRU. The input to the networks is multi-view segmented coronary vessel tree (binary images) and the output is the sequential data structure defined in section 4.2.1. The overall system for 3D reconstruction pipeline is illustrated in Figure 4.16.
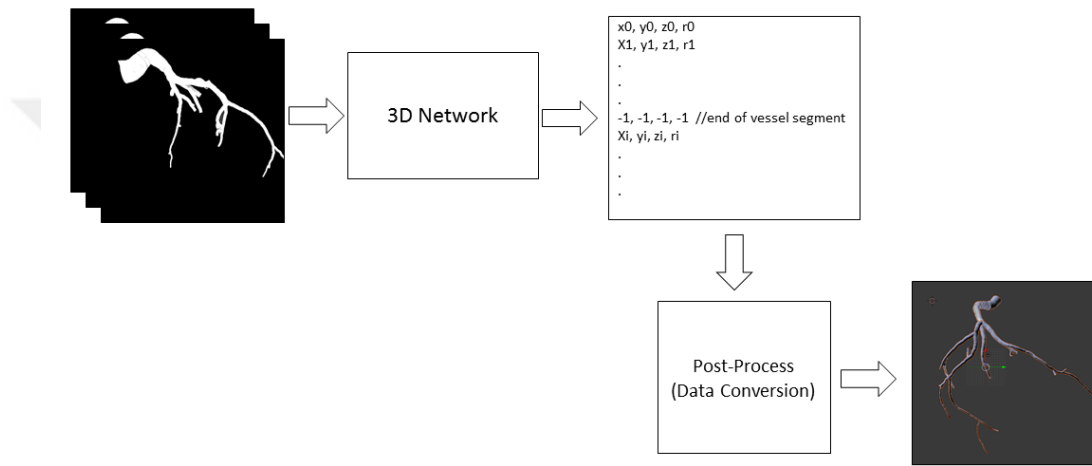
Figure 4.16 3D reconstruction pipeline for all networks proposed in this study.

4.2.4.1. Multi-view Fully CNN

We propose fully CNN network as shown in Figure 4.17 for the construction of 3D synthetic coronaries. In this architecture, multi-view images are provided to the network as channels. Thus, we name it as Multi-view Fully CNN (MvFCNN). We add residual connections to carry features from previous layers to deeper layers. This approach generally prevents loss of information and performs better performance on the given task.
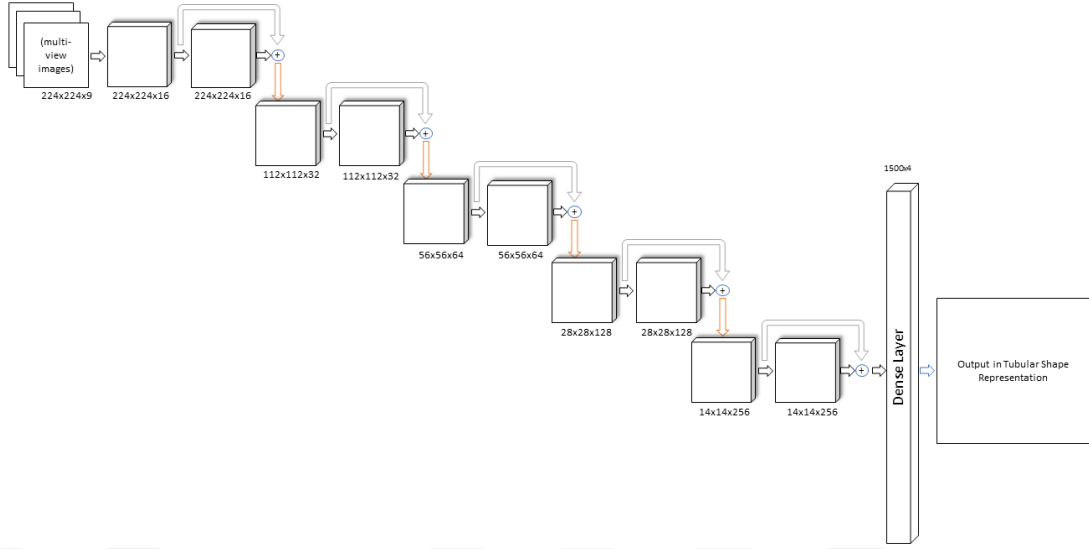
Figure 4.17 Multi-view fully CNN 3D reconstruction network. Activation function followed by each 3x3 convolution layer is Leaky-ReLU. Gray and orange arrows show residual connections and down-sampling with strided convolution (s=2), respectively.

The convolution filter size for each layer for MvFCNN architecture is 3x3. Down-sampling is achieved using convolution operation with a stride of 2 as in Sine-Net presented in Section 4.1.4 for vessel segmentation. In this way, the architecture continues to update weights while shrinking the size of the input (or embedding information from inputs) in intermediate layers. We utilize residual connections to transfer hierarchical representations to deeper layers of the network. This kind of architecture has superior performance as demonstrated in many fields and in this study for vessel segmentation shown in Section 5.1. In the end, we utilize fully connected layer (dense layer) to transform embedded information into the data structure we defined for the tubular shapes. Finally, we convert tubular shape representation into 3D objects using some math described in Section 4.2.1.

4.2.4.2. Time-distributed Auto-encoder CNN

We use the same architecture called MvFCNN in this part, but we modify input and output layers to build a larger network as shown in Figure 4.18. The differences are that (1) input images are separately supplied to the network instead of giving them as

channels and (2) we remove the fully-connected layer at the end. In this way, each image is encoded into a feature space with the same auto-encoder. The features coming from different images are combined and used to produce our structured output with fully-connected layer. Figure 4.18 visualizes this process clearly. In this figure, binary images of the same coronary tree are denoted as X_1, X_2, ..., X_n. These images are supplied to the same network to produce corresponding embedded features E_1, E_2, ..., E_n. Finally, embedded features are fed into a dense layer to produce our structured output.
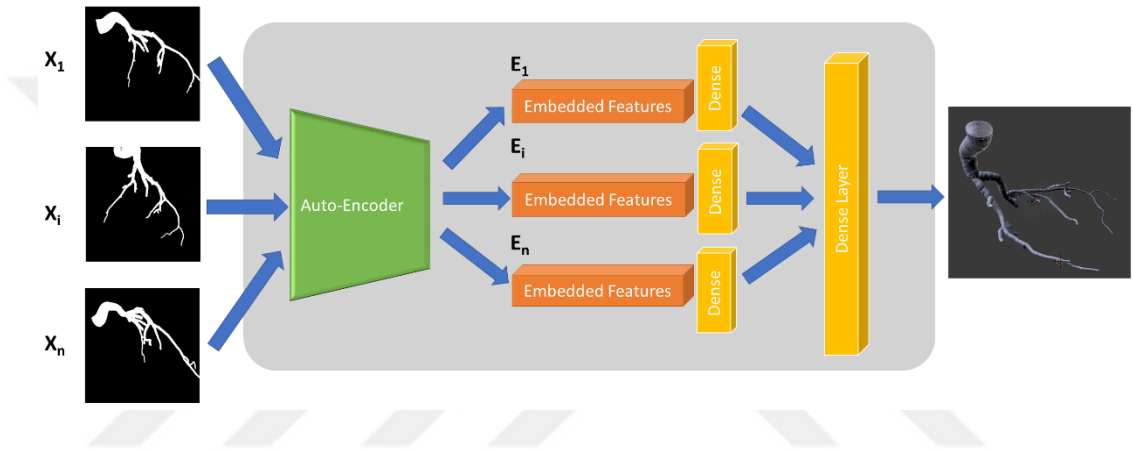
Figure 4.18 Time-distributed auto-encoder network for 3D reconstruction of the coronary tree.

For simplicity, we abbreviate time-distributed auto-encoder network shown in Figure 4.18 as "TDAEn-Dense". In this naming convention, TD stands for time-distributed, AEn stands for auto-encoder and Dense is the final layer.

4.2.4.3. Auto-encoder Followed by LSTM

We use auto-encoder network to embed features into a latent-space, and pass these features into memory cells of LSTM to hold necessary information that is coming from the current input. We remove the fully connected part of the network in Figure 4.17 and use it as auto-encoder. We place a decoder network at the end of LSTM to extract embedded information through auto-encoder and LSTM. In this configuration, LSTM learns to hold important features in its internal memory and updates its hidden state according to incoming inputs. The overall network diagram for auto-encoder-decoder

architecture is illustrated in Figure 4.19 and abbreviated as En-LSTM-Dec in this study. The LSTM cell size is 1500.
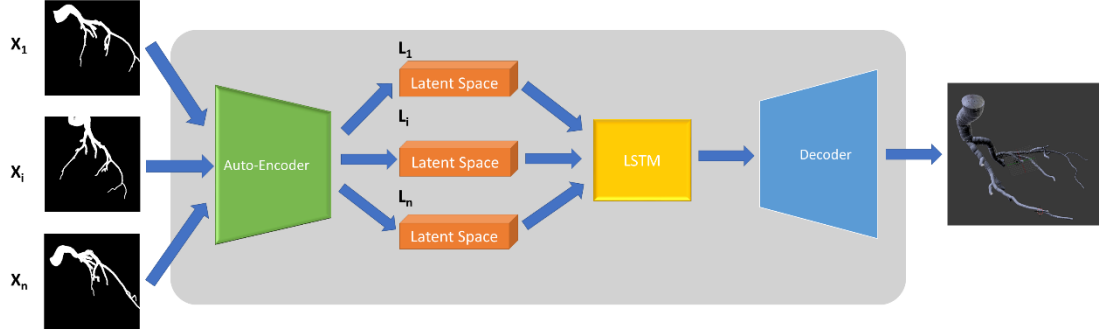
Figure 4.19 3D reconstruction network with LSTM and decoder.

The decoder part plays an important role in the architecture because the entire input sequence is now represented in LSTMs memory cells as a fixed-length vector. This encoded vector should be transformed into desired output sequences. The intuition here is that the decoder learns this complex representation and extracts the embedded information that LSTM holds. Thus, the output of LSTM is provided to the input of the decoder part in Figure 4.20 to decode the embedded information within it. The output is the data structure we defined previously in Section 4.2.1.
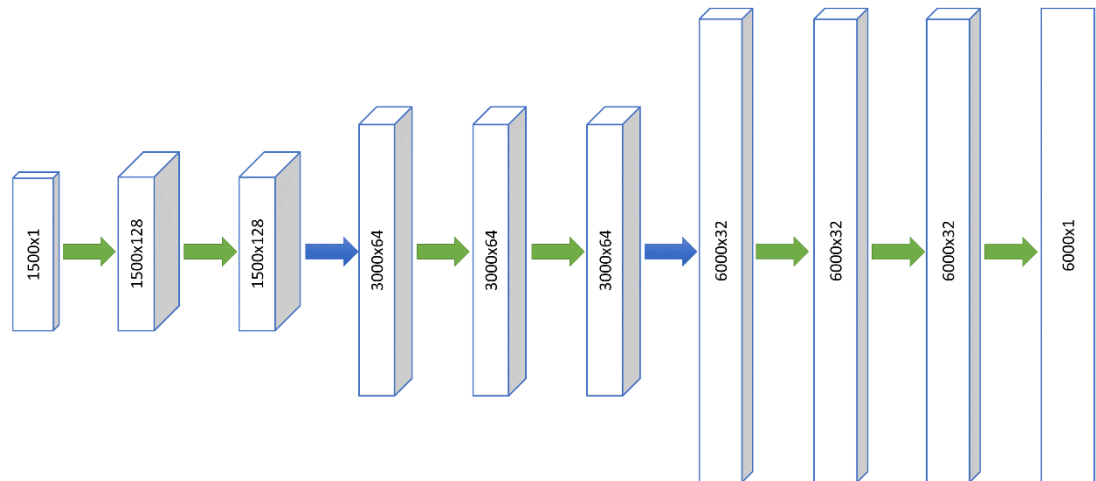
Figure 4.20 The decoder part for 3D reconstruction network.

4.2.4.4. Auto-encoder Followed by GRU

GRUs can work as efficiently as LSTMs [35]. Thus, we also construct the same architecture with GRU instead of LSTM shown in Figure 4.19 and name it as "En-GRU-Dec". GRU cell size is set to 1500 for a fair comparison with LSTM architecture. The reconstruction results shared in section 5.2.
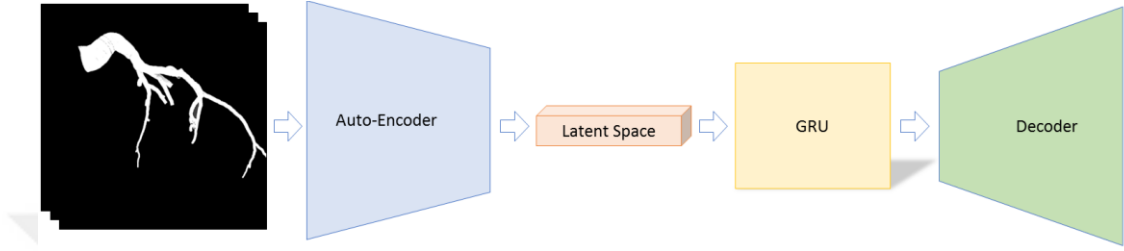
Figure 4.21 3D reconstruction network with GRU.

4.2.5. Loss Function

Center points and corresponding radiuses are estimated via 3D networks presented in Section 4.2.4. The output values are floating-point numbers rather than a binary value. Therefore, 3D reconstruction process in this study can be regarded as a regression problem. To train regression networks, we need an objective function that minimizes the error between target and prediction. Mean squared error (MSE, L2-Loss), mean absolute error (MAE, L1-Loss), Huber loss, Log cosh and Quantile loss are the top five commonly used loss functions in machine learning for regression problems. There are also variations of robust loss functions in the literature that consider the impact of outliers [148]. In this study, we prefer MSE because our synthetic data include negative values for radiuses that might be treated as outliers in such loss functions. We use MSE for estimating center points and radiuses of tubular shapes as follows:

$$L(P, \tilde{P}, r, \tilde{r}) = \frac{1}{N} \sum_i (P_i - \tilde{P}_i)^2 + \frac{1}{N} \sum_i (r_i - \tilde{r}_i)^2 \qquad (4.17)$$

where P_i and $\tilde{P}_i$ are ground-truth and predicted 3D central point coordinates of circles for our tubular data representation, respectively. r_i is reference radius value and $\tilde{r}_i$ is the predicted radius value for the center point at i.

4.2.6. Training

Keras [144] library is used for network construction and its training. There are 4 networks for 3D reconstruction of coronary arteries. We utilized Adam optimizer [145] during the training of each network. The number of samples is 10000 coronary models produced synthetically as mentioned previously in 4.2.2. We divide the last 70% of data-set for training of networks and 30% for test-set. 10% of training-set is used for validation. The number of epochs is set to 500 and the best weights are saved through all training processes. The batch size is set to 10. We use Nvidia Tesla P100 graphic card for training. Training time is saved in Table 4.5 for each 3D network architectures we proposed.

Table 4.5 Training time for 3D networks for 500 epochs.

Network	# of samples in set	Training time (hours)
MvFCNN	7000	5.952
TDAEn-Dense	7000	18.056
En-LSTM-Dec	7000	34.861
En-GRU-Dec	7000	30.254

4.2.7. 3D Reconstruction Performance Measures

We evaluate 3D reconstruction predictions for 3D deep network architectures mentioned in Section 4.2.4 using Chamfer Distance (CD). However, this measure works with point cloud representation. Therefore, we use center points in our structured data definition of 3D tubular objects for quantitative comparisons. Chamfer Distance is calculated using (4.18).

$$D(S1, S2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2$$

$$+ \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

(4.18)

where S_1 and S_2 are two point sets. As it can be inferred from the formula, CD finds the nearest point in the other point set and sums the square of distance up for each point in each point cloud set. It is utilized in ShapeNet's [134] shape reconstruction challenge. Smaller values of CD indicate closer predictions to the corresponding ground-truths.

CD is the evaluation criterion for estimating 3D points in our data representation. However; our networks also predict the radius of each point. To compare network performance quantitatively, we use mean squared error to evaluate average error on radiuses as follows:

$$E_R(r, \tilde{r}) = \frac{1}{M} \frac{1}{N} \sum_{s} \sum_{i} (r_{si} - \tilde{r}_{si})^2$$

(4.19)

where M is the number of test samples, N is the number of points in sample s, r_{si} and $\tilde{r}_{si}$ are the reference (ground-truth) and predicted radius values in test sample s for the point at i.

CHAPTER 5

EXPERIMENTAL RESULTS

This thesis proposes 2 sets of algorithms: (1) Blood vessel segmentation and (2) 3D reconstruction. Firstly, we demonstrate the segmentation performance on blood vessels for Sine-Net architecture. Secondly, we share 3D reconstruction results for synthetically produced coronary arteries.

We proposed another network derived from U-Net and the results for this network can be examined in [13]. We include only Sine-Net results in this thesis due to better segmentation performance for blood vessels.

5.1 Experimental Results for Blood Vessel Segmentation

The segmentation performance for Sine-Net architecture is presented under this section. We have performed 8 different training as given in Table 4.3. Besides single and hybrid database tests, we also investigate the effect of preprocessing. Moreover, Sine-Net is compared with human segmentation performance.

The image size is different in each data-set, so it should be noted that rotation and translation operations in the data augmentation technique may cause the vessel pixels to exceed the border of images. Thus, we determine the input size of the network as 448x448 which is compatible with our data augmentation method presented in Section 4.1.2. Since test images are bigger than the network input size, we divide images into a minimum number of patches to fit each patch into the input. For the overlapping regions in the patches, we take a weighted average of predictions while producing the final output.

5.1.1 Human Performance

DRIVE (for only the test-set), STARE and CHASE_DB1 databases include annotated images from two different experts. In the related literature, the general trend is to select one expert's labeling as the ground-truth for training a model and another is used for

comparing the model with an observer. We employ the first expert's labeling as the gold standard and the second expert's labeling as the human performance in this study. STARE database expert names are labeled as "ah" and "vk". We follow alphabetical order here and accept "ah" as the first expert. The performances of the second observer on blood vessel segmentation are presented in Table 5.1 and compared with networks in Table 5.5.

Table 5.1 The human performance on manual segmentation of retinal vessels for databases. The first expert labeling is accepted as ground-truth and the second expert is accepted as an observer. The average term shows the weighted average performance of the second expert for all databases.

Database	Accuracy		Sensitivity		Specificity	
	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
DRIVE	0.9637	0.0033	0.7757	0.0596	0.9819	0.0055
STARE	0.9568	0.0145	0.9182	0.0355	0.9589	0.0165
CHASE_DB1	0.9635	0.0054	0.8339	0.0265	0.9792	0.0038
Average	0.9635	0.0076	0.8145	0.0731	09772	0.0117

5.1.2   Single Database Results

DRIVE database is already divided into training and test set. We do not modify the sets and use them as they are. STARE and CHASE_DB1 databases are not initially configured for test and training sets. Thus, we apply the same division procedure mentioned in section 4.1.2.

Table 5.2 presents the performance of the test set of single database training for Sine-Net architecture. STARE database training with preprocessing has the highest accuracy and specificity but the lowest sensitivity. It means the network can identify non-vessel pixels better than vessel pixels. As mentioned earlier in Section 4.1.1, STARE database is the most challenging database among others due to poor contrast, and some images are complicated by a disease that causes vessel-like structures to be deformed. The standard deviation values also support this because σ values for

accuracy and AUC metrics vary 2 or nearly 3 times higher than others. It is also clearly seen that preprocessing affects sensitivity to be increased except CHASE_DB1 database. It is an expected result because preprocessing operation enhances the contrast in images, and vessels become more distinguishable. The best sensitivity and AUC values are obtained in DRIVE database training with preprocessing.

Table 5.2 Single database results for Sine-Net. The table summarizes evaluation metrics in terms of mean ($\bar{x}$) and standard deviation (σ). The highest metrics are highlighted in bold. PP at the end of training names indicates preprocessing operation applied.

Training	Accuracy		Sensitivity		Specificity		AUC	
	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
DRIVE	0.9689	0.0030	0.7987	0.0545	0.9854	0.0041	0.9851	0.0039
DRIVEPP	0.9685	0.0024	**0.8260**	0.0537	0.9824	0.0049	**0.9852**	0.0041
STARE	0.9682	0.0127	0.6574	0.0879	0.9933	0.0050	0.9748	0.0159
STAREPP	**0.9711**	0.0108	0.6776	0.0828	**0.9946**	0.0018	0.9807	0.0122
CHASE	0.9678	0.0052	0.8011	0.0476	0.9815	0.0036	0.9833	0.0045
CHASEPP	0.9676	0.0053	0.7856	0.0519	0.9825	0.0035	0.9828	0.0050

Figure 5.1 shows prediction results from single database training for DRIVE database. The results of the predictions with and without preprocessing operations are extremely close. Figure 5.1c and Figure 5.1d demonstrate that thinner vessels are estimated better with preprocessing than without preprocessing results. Table 5.2 support this in the sensitivity metric, which is the highest of all training. For the thick vessels, Sine-Net performs well with and without preprocessing.

For STARE database training results shown in Figure 5.2, preprocessing effects on segmentation seems limited visually, but Table 5.2 demonstrates that there is 0.0202 improvement on sensitivity metric. If we closely look at Figure 5.2c and Figure 5.2d,

we can see little differences in the vessel ends. However; we cannot state that preprocessing result is better than without preprocessing because in some cases without preprocessing training show visually better segmentations or vice versa.

(a)   **(b)**

(c)   **(d)**
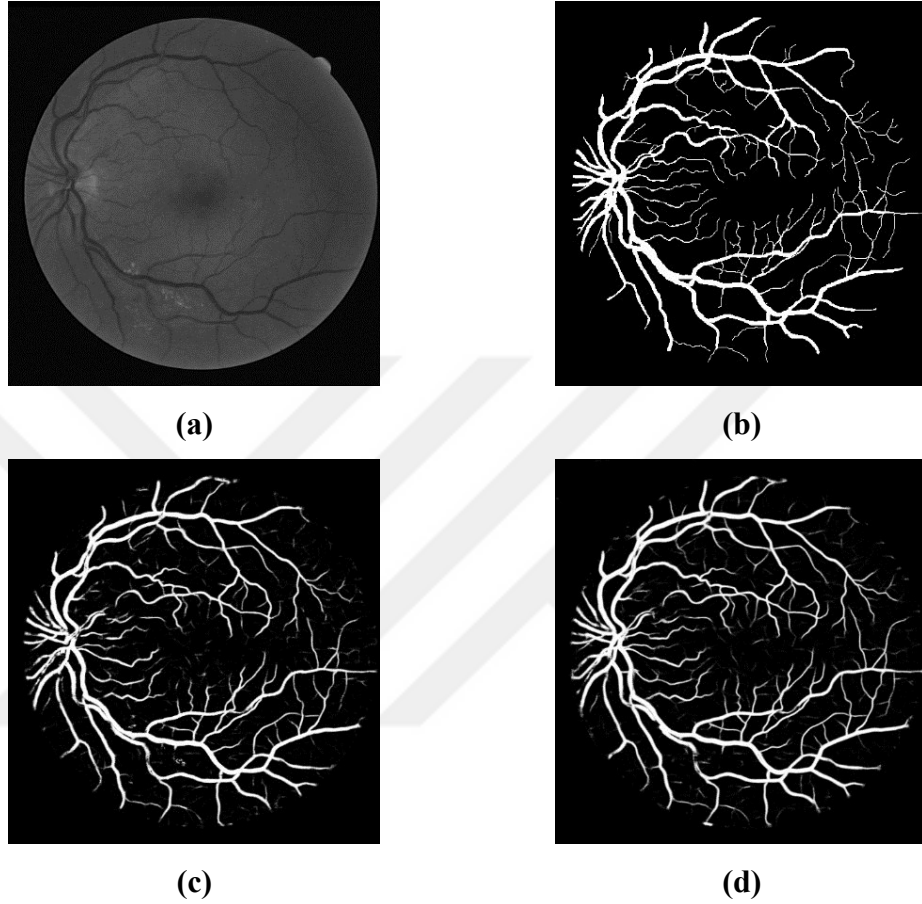
Figure 5.1 Single database training prediction of the proposed network for DRIVE test image #03. The original image in grayscale is shown in (a) and corresponding ground-truth (expert #1 labeling) image is shown in (b). The predictions of the network for with and without preprocessing are delineated in (d) and (c), respectively.

For CHASE_DB1 training results shown in Figure 5.3, thick vessels are successfully detected by the network, but it fails on some of thinner vessels as compared to corresponding ground-truth given in Figure 5.3b. If we take a closer look inside the red rectangle for the same figure, it is very hard to see thin vessels in the original image. Moreover, this part is not labeled by the second expert as shown in Figure 5.4

for the same image. If we compare the effect of preprocessing of Sine-Net on training CHASE_DB1 database, the preprocessing training produces noisier background. This is also supported by the sensitivity metric presented in Table 5.2. The sensitivity is decreased by 0.0155 for the preprocessing training.

(a)

(b)

(c)

(d)
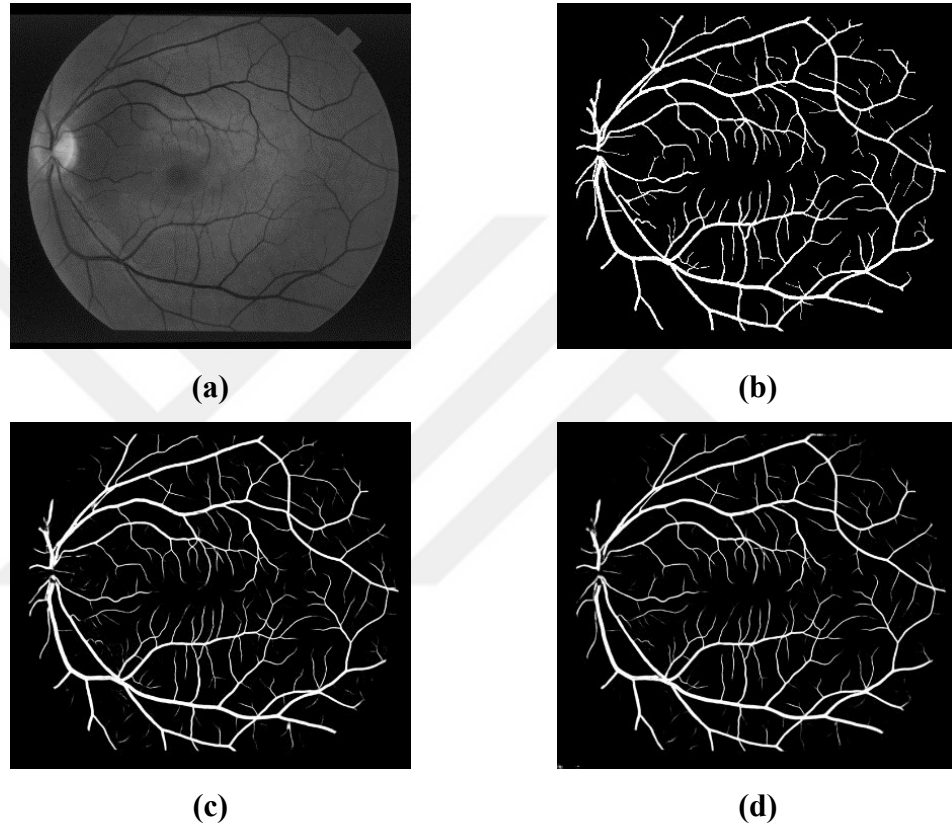
Figure 5.2 STARE database training results for Sine-Net. (a) shows the original image (#0255), (b) shows (a)'s ground-truth. (c) is the prediction of the network without preprocessing and (d) is the prediction with preprocessing of (a).
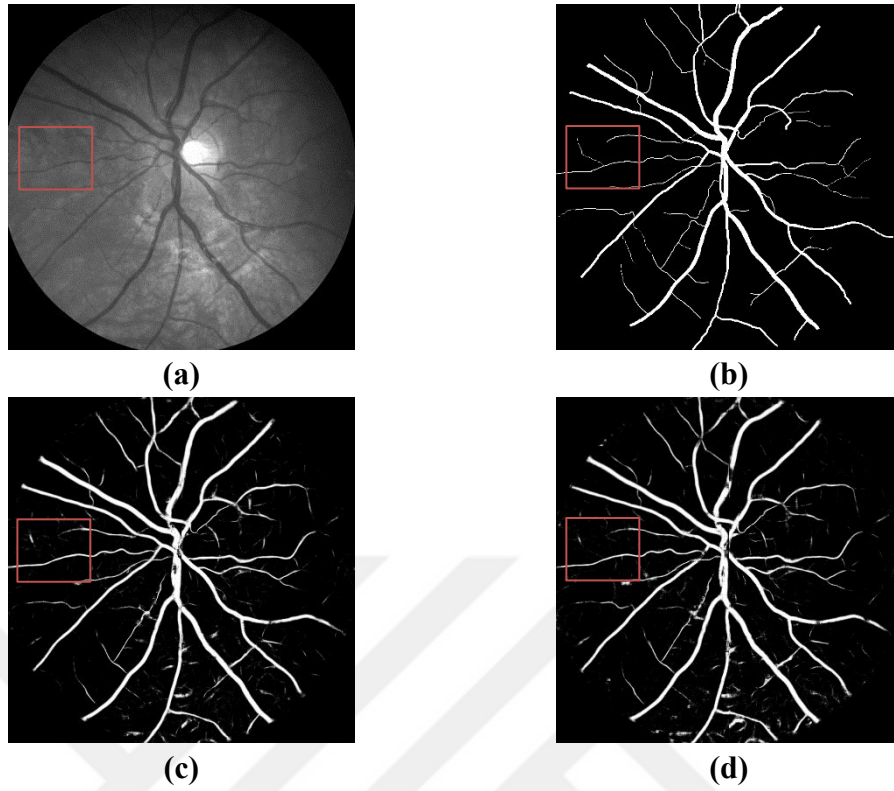
Figure 5.3 CHASE_DB1 database training results for the proposed network. (a) is the original image (#11L), (b) is ground-truth of (a), (c) is the prediction without preprocessing and finally, (d) is the prediction of the network of (a) with preprocessing.
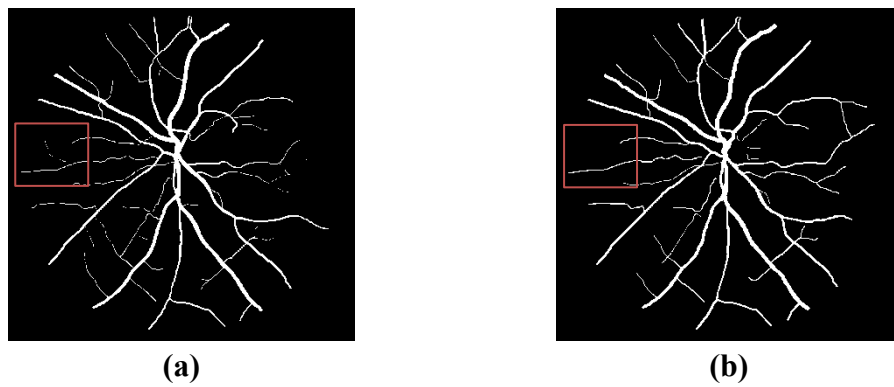
Figure 5.4 Experts labeling comparison for the image (#11L) in CHASE_DB1 database. (a) shows the reference image (expert #1) and (b) shows labeling from 2nd expert.

We produce some of the outputs from intermediate layers to show and understand what the network learned. We can see that borders are extracted in very early layers from Figure 5.5b, and the network carries features to a deeper level by residual connections as in Figure 5.5c. Up-sampling operations cause the network to produce some checkerboard noise (Figure 5.5d), but it is recovered with residual connections at down-sampling layer (Figure 5.5e). Before the final layer seen in Figure 5.5f, the network almost estimates each pixel location for vessels.

(a) **(b)** **(c)**

(d) **(e)** **(f)**
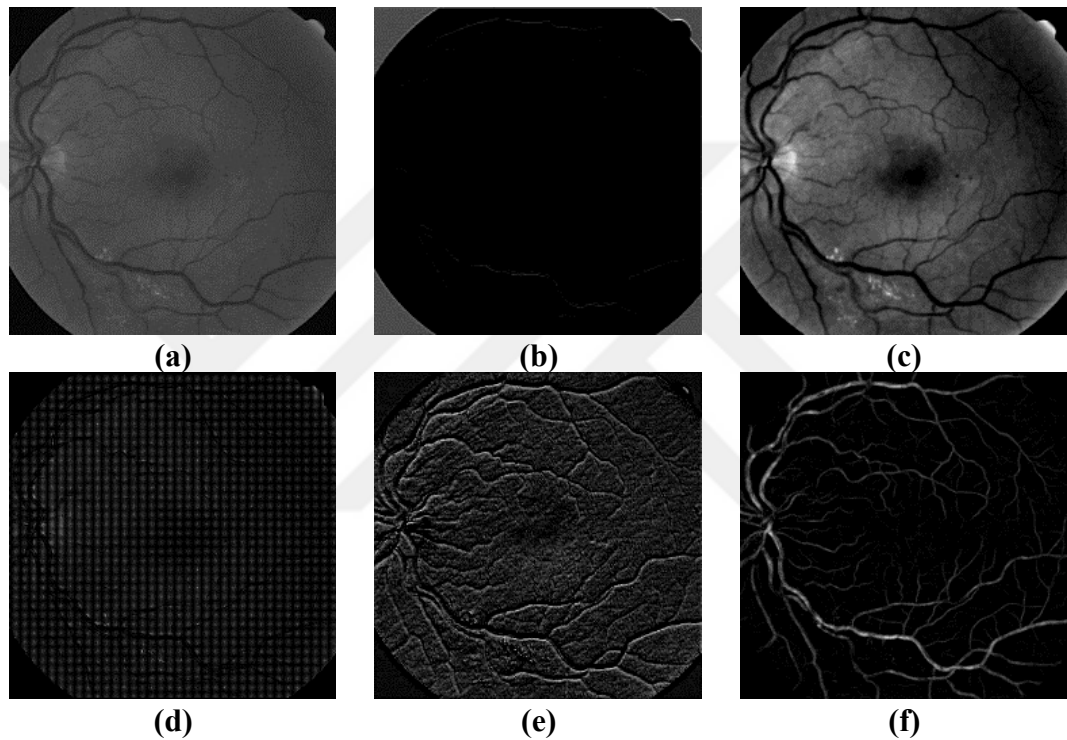
Figure 5.5 Some of the outputs from intermediate layers for DRIVE training. (a) shows the original image, (b) shows one of the channel outputs from layer 1 which network learns how to extract the border around retina image. (c) is the output from the layer before up-sampling. Due to residual connections, an image similar to input is obtained at a deeper level of the architecture. (d) and (e) are middle layer output of up-sampling part and down-sampling part. (f) is one of the outputs from the layer before the final convolution.

Segmentation of X-Ray Angiogram Images for Single Database Training

We perform tests on angiogram images as well to figure out Sine-Net performance on our actual task. We show the prediction result of each database training for one of the angiogram images from [142]. We visually interpret the segmentation performance for angiogram images in Figure 5.6 because quantitative evaluation is not possible due to lack of reference (ground-truth) images. We observe that DRIVE database training with pre-processing produces the poorest segmentation performance. Without pre-
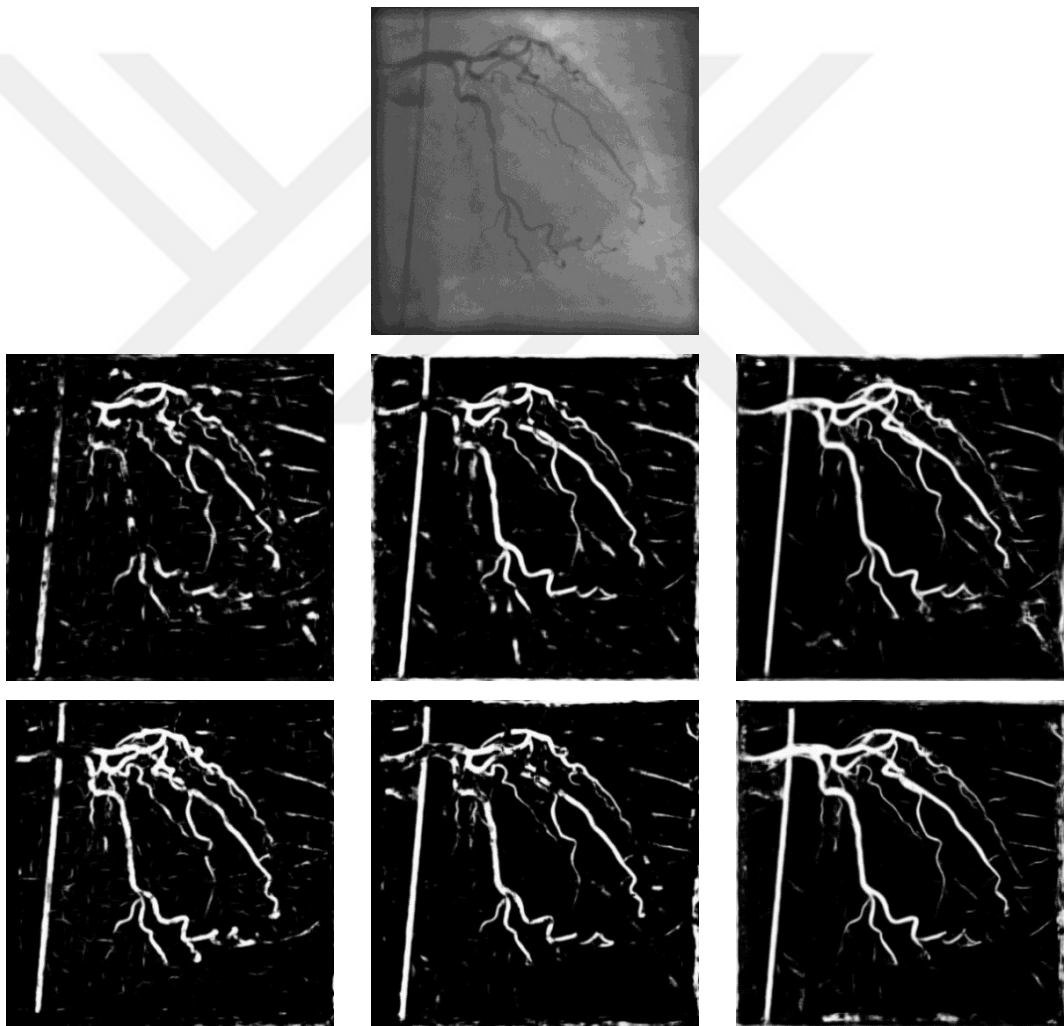
Figure 5.6 Sine-Net prediction results for angiogram images. Rows show results for with and without pre-processing the input, columns show results from DRIVE, STARE and CHASE_DB1 database training, respectively.

processing operation, the network shows better segmentation for DRIVE training. We consider the reason could be DRIVE database images are already in good contrast and quality as compared to angiogram images. Pre-processing improves the segmentation in STARE database training but has almost no effect on CHASE_DB1 training. Among single database training, CHASE_DB1 training has performed the best segmentation for both with and without pre-processing of the input.

5.1.3 Hybrid Database Results

In order to assess the generality of the proposed network, we have conducted tests on the network using images from DRIVE, STARE and CHASE_DB1 databases at the same time. Table 5.3 demonstrates the segmentation performance on blood vessels for hybrid training of each database test-set. The first half of Table 5.3 shows without preprocessing results while the latter half shows the results with preprocessing. Since the number of test images is different in each database, we summarize the overall performance of the corresponding network as a weighted average in Table 5.3.

Table 5.3 supports that STARE database is the most challenging database among others in this study due to higher σ values. There are slight increases and decreases in metrics if we compare the results with single database tests in Table 5.2. Instead of comparing each metric one by one, AUC is considered as the ultimate metric for the comparison because it includes true-positive rate (TPR) and false-positive rate (FPR) at the same time in its calculation. AUC metric is increased in hybrid training in DRIVE and STARE databases for without preprocessing operation as compared to single database training. However, it is slightly decreased in DRIVE and CHASE_DB1 database training for with preprocessing operation (see Table 5.2 and Table 5.3). If we compare with and without preprocessing for the hybrid-training results in Table 5.3, AUC measure is slightly decreased in all database training results for pre-processing. Thus, we can conclude that the network can produce successful results even there is no pre-processing.

Table 5.3 The segmentation performance from hybrid database training for Sine-Net. The table summarizes evaluation metrics in terms of mean ($\bar{x}$) and standard deviation (σ). The highest metrics are highlighted in bold for each column. PP means preprocessing enabled or disabled.

PP	Training	Accuracy		Sensitivity		Specificity		AUC	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
N	DRIVE	0.9689	0.0031	0.7817	0.0558	0.9871	0.0039	**0.9853**	0.0040
	STARE	0.9702	0.0127	0.6998	0.1031	0.9920	0.0050	0.9837	0.0079
	CHASE_DB1	0.9668	0.0047	0.7804	0.0570	0.9822	0.0035	0.9816	0.0043
	Weighted avg.	0.9686s	0.0060	0.7670	0.0713	0.9868	0.0051	0.9841	0.0050
Y	DRIVE	0.9685	0.0028	**0.7882**	0.0541	0.9859	0.0041	0.9843	0.0039
	STARE	**0.9712**	0.0124	0.6999	0.1248	**0.9928**	0.0027	0.9823	0.0104
	CHASE_DB1	0.9666	0.0053	0.7668	0.0613	0.9830	0.0037	0.9810	0.0050
	Weighted avg.	0.9685	0.0060	0.7676	0.0770	0.9865	0.0049	0.9832	0.0057

Hybrid database predictions for without and with pre-processing are given in Figure 5.7 and Figure 5.8, respectively. In order to compare predictions of single and hybrid database training, we present them sside by side. Each row in Figure 5.7 and Figure 5.8 present randomly selected test images from DRIVE, STARE and CHASE_DB1 databases. From both figures, we can see that the network produces very close predictions to the corresponding ground-truths. Hybrid database training improves vessel segmentation, but it is not visually clear. Figure 5.7 and Figure 5.8 show that the difference between single and hybrid database predictions is very limited. However, the comparison on AUC metric in Table 5.2 and Table 5.3 supports that hybrid database training has improved the vessel segmentation performance quantitatively.
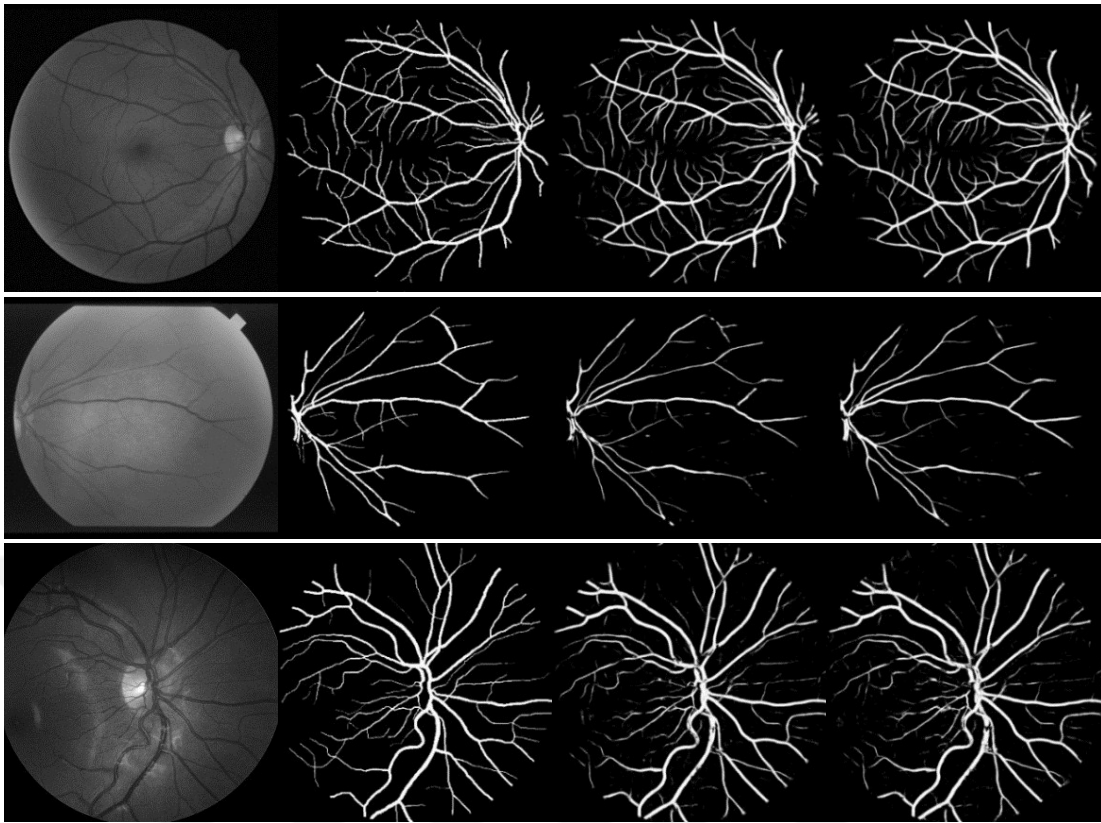
Figure 5.7 The prediction comparisons of Sine-Net for single and hybrid database training without pre-processing. Rows belong to images from DRIVE (test image #19), STARE (image #0291) and CHASE_DB1 (image #12R) databases, respectively. Columns show in order original grayscale retinal image, its ground-truth, network prediction of single database training and network prediction of hybrid database training.

If we compare the result for single and hybrid database training, the network misses some of thick vessels due to pathology images in STARE database. The network has learned some exceptional cases from these samples. For better visualization, we illustrate an example image containing pathology and its prediction by the network in Figure 5.9.
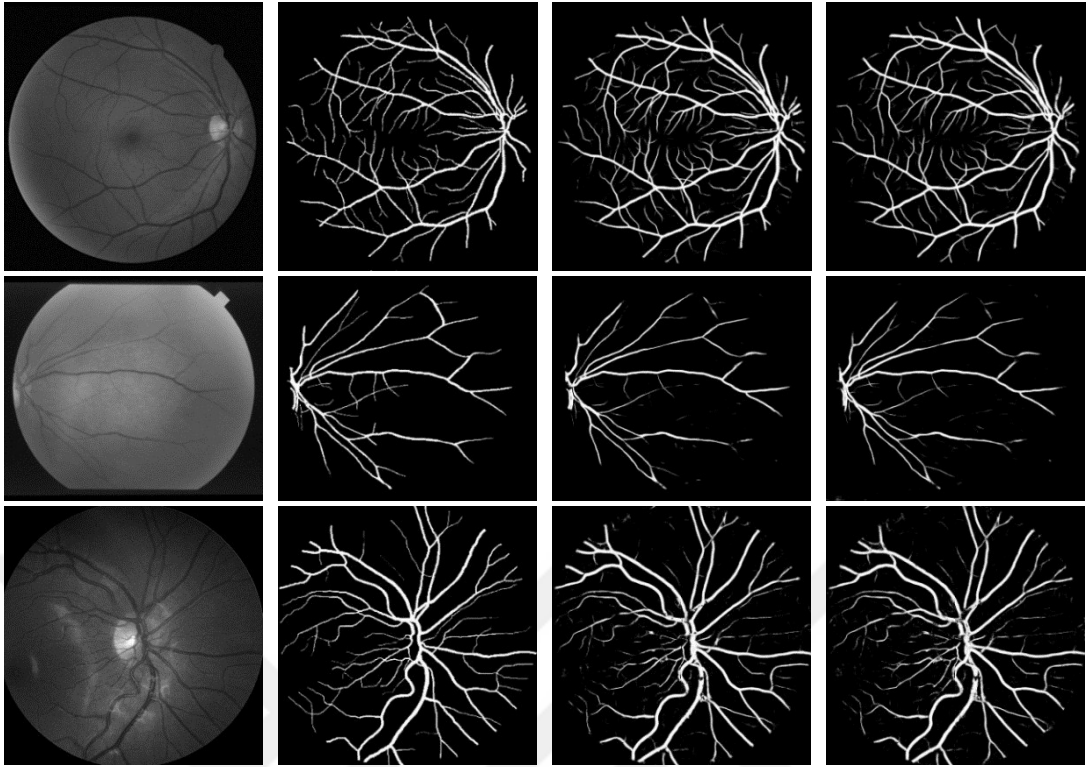
Figure 5.8 Hybrid database predictions of Sine-Net with pre-processing for all images in the same order in Figure 5.7.

(a)

(b)

(c)

(d)
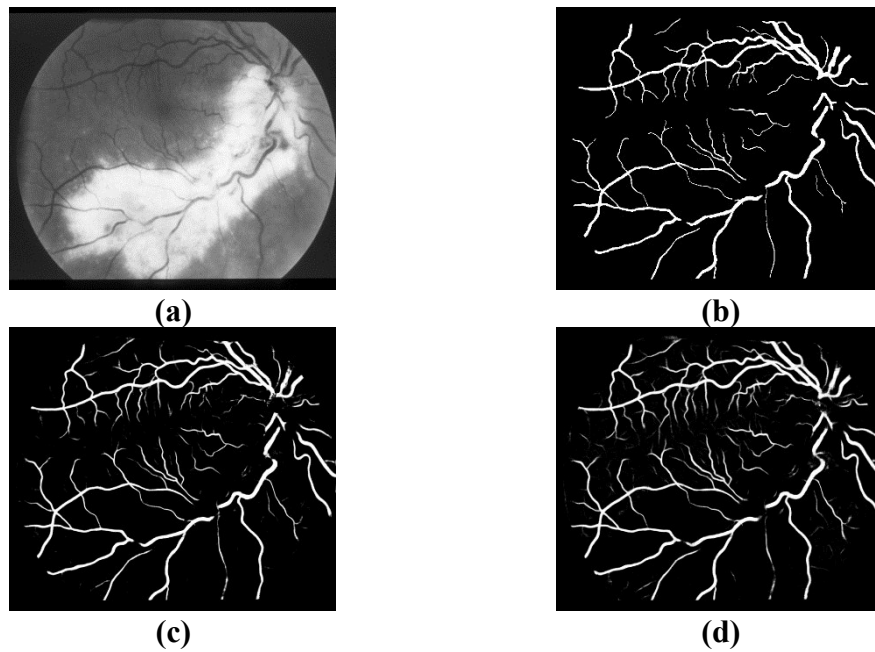
Figure 5.9 An image from STARE database (im#0044) containing pathology and its network prediction. (a) shows grayscale image, (b) shows its corresponding ground-

truth labeled by 1st expert, (c) and (d) are Sine-Net predictions without preprocessing operation for single database and hybrid database training, respectively.

The network is sensitive to vessel-like structures and catches some of thin vessels which are not labeled by experts or incorrectly predict continuous noisy background structures as vessels. Therefore, sensitivity and specificity values become lower than what we expect. We show the network capability to catch thin vessels that are not labeled by the experts in Figure 5.10.
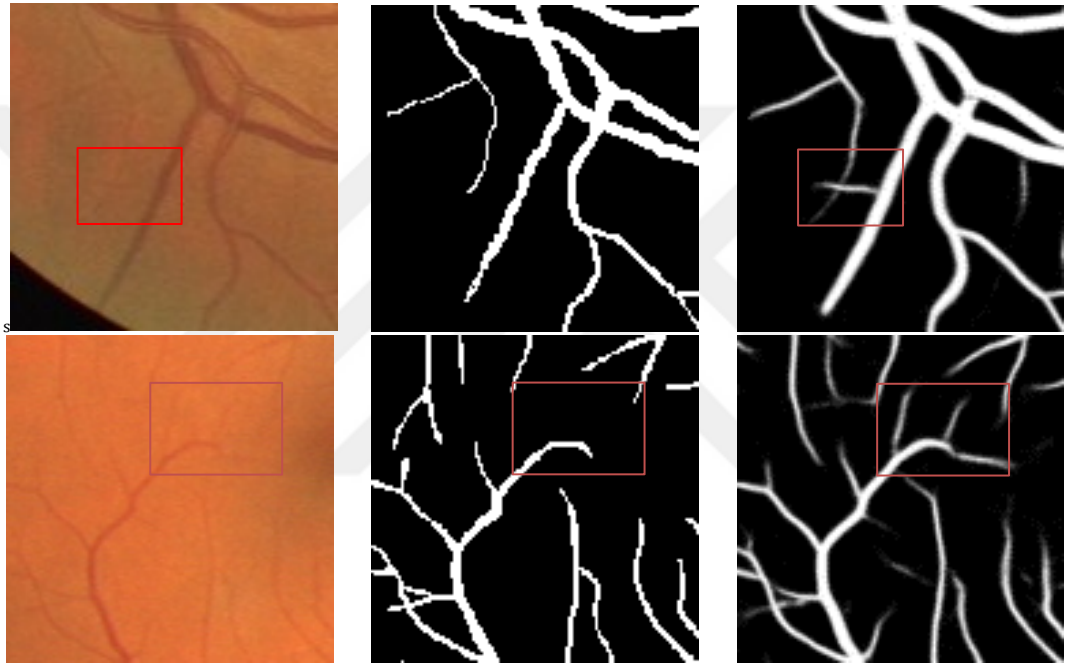
Figure 5.10 The network capability of catching thin vessels. Top row images belong to DRIVE test image #01, and below row images belong to DRIVE test image #16. Columns show the images in order that the original, ground-truth and the prediction of the network without preprocessing for DRIVE database.

Segmentation of X-Ray Angiogram Images for Hybrid Database Training

The main task of the first part of this study is to segment X-Ray angiogram images. Thus, we introduce hybrid database training with an appropriate data augmentation method for generalization of blood vessel segmentation. Figure 5.11 shows the prediction results of angiogram images from [142].

| (a) | (b) | (c) |
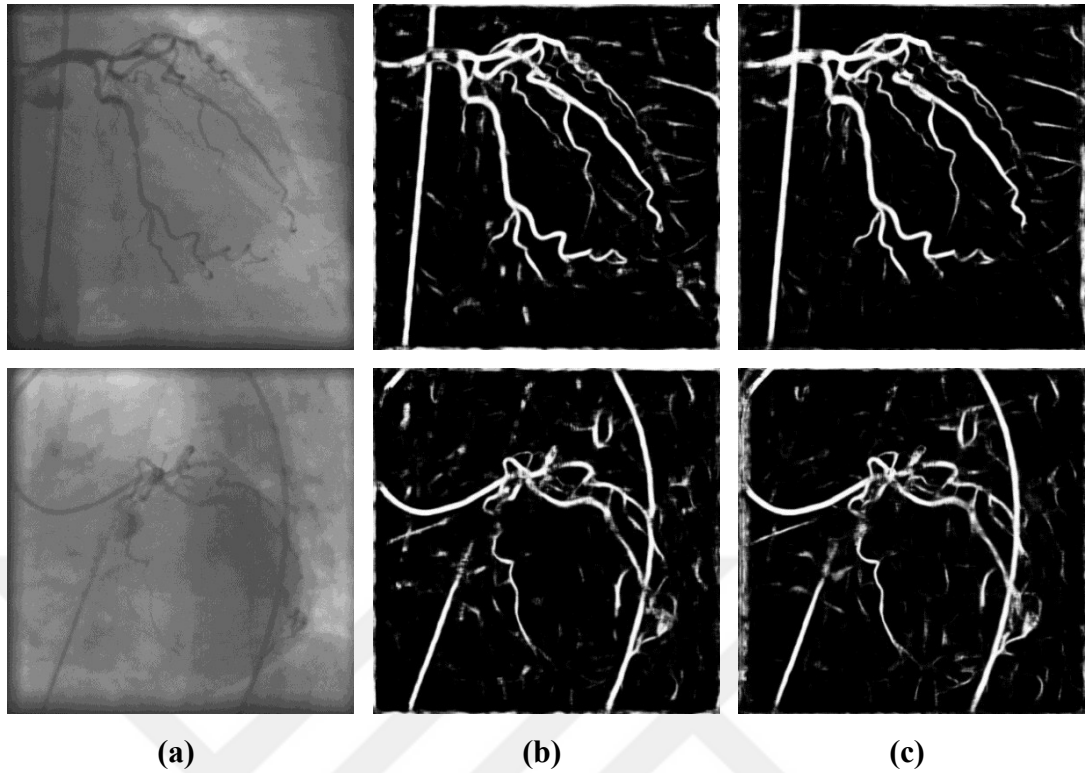
Figure 5.11 The segmentation of angiogram images for hybrid database training. (a) shows angiogram image, (b) and (c) show predictions with and without pre-processing of Sine-Net for (a).

The proposed network (Sine-Net) has learned to differentiate background and vessel pixels from retinal images as shown in Figure 5.11. We observe that Sine-Net performs better without pre-processing the input. This is because it is very hard to find an appropriate pre-processing method for all forms of vessels. The results demonstrate that deep learning architectures can adapt their input and produce satisfying predictions with raw data. If we compare with and without pre-processing results for hybrid database training, we conclude that pre-processing increases the potential to capture noise existing in the background due to tissues and organs in X-Ray angiogram images. This is expected behavior because pre-processing is achieved mainly for enhancing contrast in images. If we compare blood vessel training through this study for angiogram images, we conclude that hybrid database training visually produces better predictions than single database training (see Figure 5.6 and Figure 5.11).

5.1.4 Execution Time

The tests are conducted on the same server as mentioned in Section 4.1.6 and a Linux based notebook computer with the following hardware properties: Intel(R) Core (TM) i7-4700HQ CPU @ 2.40GHz and 16 GB 1600 MHz DDR3 RAM. The average execution time for all databases is summarized in Table 5.4 for providing an idea about the utilization of our network in a real application. It is better to keep in mind that there are 4 predictions that the network produces for every single image due to the variable image size of databases.

Table 5.4 Average prediction time for a single image of the proposed network. Note that a single image consists of at least 4 sub-images for our case due to variable image size.

Tests	Average Running Time (in seconds)
Proposed (Server, GPU)	0.3501
Proposed (Notebook, CPU)	9.3217
Li *et al*. (CPU) [76]	70
Mo *et al*. (GPU) [82]	92
Hu *et al*. (GPU) [83]	1.1
Liskowski *et al*. (GPU) [86]	0.4
Feng *et al*. (GPU) [92]	0.063

We compare the proposed network with the latest reported works in terms of average running time for a single image prediction. Table 5.4 shows that Sine-Net is roughly 263 times faster than the work of Liskowski [86], 3.14 times faster than the work in [83] and 1.14 times faster than that in [82] for GPU implementations. Our work is approximately 7.5 times faster than the work published by Li *et al.* [76] for CPU implementations. [76] and [86] utilize sliding window approach to predict vessel maps which cause neighboring pixels to be calculated multiple times. These redundant computations degrade the performance of execution time. Since the number of layers is lower in [92], segmentation time is shorter than ours. Sine-Net is computationally efficient due to its fully convolutional structure and selection of bigger input patch

size. Contrary to work in [86], it predicts all pixels at once instead of using a patch to predict the central pixel alone. The network we proposed can be used with GPU in clinical applications which produce segmentation image in nearly one-third of a second. Sine-Net GPU implementation is 26.6 times faster than CPU implementation in our setups.

5.1.5   Comparison

Numerous works exist in the related literature for segmentation of retinal blood vessels, and we compare our results with the state-of-the-art works which are specifically tested on DRIVE, STARE and CHASE_DB1 databases. Since our method depends on supervised learning, we focus on mostly supervised solutions for a fair comparison. In addition, we include the segmentation performance of a human to compare our network segmentation capability with observers. We here present our results besides other works in Table 5.5 for single database tests and Table 5.6 for hybrid database tests.

Table 5.5 Single database performance comparisons. Human indicates the 2nd expert segmentation performance with respect to 1st expert. *pp* and *wopp* denote that network training is achieved with and without pre-processing operations.

Database	Methods	Year	Accuracy	Sensitivity	Specificity	AUC
DRIVE	Wang *et al.* [78]*	2014	**0.9767**	0.8173	0.9733	0.9475
	Li *et al.* [76]*	2015	0.9527	0.7569	0.9816	0.9738
	Liskowski *et al.* [86]	2016	0.9515	0.7520	0.9806	0.9710
	Mo *et al.* [82]	2017	0.9521	0.7779	0.9780	0.9782
	Jiang *et al.* [11]	2018	0.9624	0.7540	0.9825	0.9810
	Feng *et al.* [92]	2018	0.9528	0.7625	0.9809	0.9678
	Hu *et al.* [83]	2018	0.9533	0.7772	0.9793	0.9759
	Human	-	0.9637	0.7757	0.9819	-
	Our method (wopp)	2020	0.9689	0.7987	**0.9854**	0.9851
	Our method (pp)	2020	0.9685	**0.8260**	0.9824	**0.9852**
STARE	Wang *et al.* [78]	2014	**0.9813**★	0.8104	0.9791	0.9751
	Li *et al.* [76]	2015	0.9628	0.7726	0.9844	0.9879
	Liskowski *et al.* [86]	2016	0.9696	0.8145	0.9866	0.9880
	Mo *et al.* [82]	2017	0.9674	0.8147	0.9844	0.9885
	Jiang *et al.* [11]	2018	0.9734	0.8352	0.9846	**0.9900**★
	Feng *et al.* [92]	2018	0.9633	0.7709	0.9848	0.9700
	Hu *et al.* [83]	2018	0.9632	0.7543	0.9814	0.9751
	Human	-	0.9568	**0.9182**★	0.9589	-
	Our method (wopp)	2020	0.9682	0.6574	0.9933	0.9748
	Our method (pp)	2020	0.9711	0.6776	**0.9946**★	0.9807
CHASE_DB1	Li *et al.* [76]	2015	0.9581	0.7507	0.9793	0.9716
	Mo *et al.* [82]	2017	0.9599	0.7661	0.9816	0.9812
	Jiang *et al.* [11]	2018	0.9668	**0.8640**	0.9745	0.9810
	Human	-	**0.9682**	0.8339	0.9792	-
	Our method (wopp)	2020	0.9678	0.8011	0.9815	**0.9833**
	Our method (pp)	2020	0.9676	0.7856	**0.9825**	0.9828

* Field of view (FOV) defined for performance measurement.

★ The best value for the corresponding metric

Table 5.6 Hybrid database comparisons. *pp* and *wopp* denote that network training is achieved with and without preprocessing operations, respectively. D: DRIVE, S: STARE and C: CHASE_DB1 databases.

Train Set	Test Set	Methods	Year	Accuracy	Sensitivity	Specificity	AUC
D	D+S	Liskowski *et al.* [86]*	2016	0.9491	-	-	0.9700
S	D+S			0.9566	-	-	0.9776
D	S+C	Jiang *et al.* [11]	2018	0.9593	0.7121	0.9832	0.9680
S	D+C			0.9653	**0.7820**	0.9798	**0.9870**
C	D+S			0.959s1	0.7217	0.9770	0.9580
D, S, C	D, S, C	Our method (wopp)**	2020	**0.9686**	0.7670	**0.9868**	0.9841
D, S, C	D, S, C	Our method (pp)**		0.9685	0.7676	0.9865	0.9832

* FOV defined for performance measurement
** Weighted average performance of DRIVE, STARE and CHASE test data-sets

We divide Table 5.5 into 3 parts to compare the same database results together. The best value for single database results are highlighted in bold, and the best values for each metric are marked with ★. For DRIVE database, our network outperforms human performance in all metrics. It also surpasses other methods proposed in the literature in terms of all metrics except accuracy of [78]. It is noted that this may be because Wang *et al.* [78] use FOV definition (only pixels inside eye mask). Especially, Sine-Net achieves better sensitivity with pre-processing by approximately **0.01** better than the closest study proposed in [78]. For STARE database, the network achieves the best result in specificity and produces close values to the-state-of-the-art methods in terms of accuracy and AUC metrics. However, the nearest sensitivity metric to ours is achieved by Hu *et al.* in [83], it surpasses the proposed network by approximately **0.077**. This is not due to the reason that our network is less capable to detect vessels, but the number of training image samples with disease and poor contrast is lower in quantity. Some of the studies leave images out with vessels complicated by disease but we did not remove any of them. If the number of images with pathology and poor contrast is increased, the network will perform better. Our network is sensitive to vessel-like structures as DRIVE and CHASE_DB1 results support this in the sensitivity metric in Table 5.5. The network achieves the best specificity and AUC

metric for CHASE_DB1 database. It produces extremely close accuracy metrics to the second observer which has the highest score in corresponding database training. If this is omitted, the network outperforms other methods in terms of accuracy, specificity and AUC performance measures.

For the hybrid-database test results shown in Table 5.6, the proposed network achieves the highest accuracy and specificity metrics without pre-processing. The work of Jiang *et al*. in [11] proves that STARE database causes a decrease in sensitivity metric drastically due to challenging images included in the corresponding cross-database setup. By the definition of Jiang *et al*. cross-database set, the test database is not included in the training set. For example, if DRIVE and STARE databases are in training-set, they performed test on CHASE_DB1. They compared their network with unsupervised methods in this way. However; we preferred combining all data-sets into one for obtaining a more generalized and complete model for the segmentation of blood vessels. We evaluated the weighted average for a fair comparison because the number of test images is not the same in databases. Our method with the proposed network achieved superior performance in terms of accuracy and specificity and stayed on the top in terms of AUC metric for hybrid-training with the work of [11].

AUC is a preferable metric for comparison because the classes are unbalanced (see Table 4.2), and there are annotation differences inherently among databases. For instance, vessel labels marked by the second expert in STARE are thinner than the first experts as mentioned in Section 4.1.1. If we examine our network in terms of only AUC metric in Table 5.5, it has superior performance on DRIVE and CHASE_DB1 databases. For STARE and hybrid-database training, our network achieves very close performance on to the best AUC metric.

5.2 Experimental Results for 3D Reconstruction

In this section, we show the validation of our 3D reconstruction methods using synthetic data-set we produced from real subjects as mentioned earlier in 4.2.2. We share the results of several experiments performed on networks and compare their reconstruction performances in terms of CD and MSE quantitatively.

We present network comparisons under two subsections. The first section compares predictions quantitatively using statistical measures. The second section compares the average running time for each network to predict 3D model of coronary vessels.

5.2.1 Quantitative Comparison for 3D Reconstruction

We quantify the performance of each network in multi-view 3D reconstruction of coronary arteries and report the results in terms of CD and MSE. CD is used for the evaluation of center-line points in 3D and MSE is used for the evaluation of radius estimations. The lower values of both metrics indicate better 3D reconstruction that networks estimate.

Table 5.7 Reconstruction performances of networks proposed in this study.

Model Name	Chamfer Distance		Radius Error	
	μ	σ	μ	σ
MvFCNN	**0.00998328**	**0.00299526**	**0.00000580**	**0.00000226**
TDAEn_Dense	0.03288964	0.02404985	0.00014654	0.00039439
En_LSTM_Dec	1.54436864	0.17509677	0.29157797	0.15928418
En_GRU_Dec	1.64607289	0.19792573	0.31827632	0.16060889

There are 4 different 3D reconstruction networks described in Section 4.2.4. The first network, MvFCNN is a fully convolutional architecture that accepts inputs as channels. TDAEn_Dense is a variation of MvFCNN that accepts images in time-distributed fashion. In this architecture, we encode each image using the same auto-encoder and obtain hidden representations of input images. Next, we combine these feature representations using fully connected layer (dense layer) to estimate 3D shape of the coronary tree in tubular shape representation. The other two architectures are designed to have an internal memory for holding input features coming from multi-view images as in [37]. We use the same auto-encoder as in TDAEn_Dense to generate latent features. Then we feed them in LSTM or GRU to keep necessary features between multi-view images. Finally, we use the decoder network to decode the information inside LSTMs or GRUs. Table 5.7 shows the results of each network. In

order to show network comparisons fairly and clearly, we present results with statistical measures; namely, mean (μ) and standard deviation (σ).

As shown in Table 5.7, MvFCNN outperforms others. The second architecture which achieves relatively good performance is TDAEn_Dense. These two architectures have no memory structure and data is provided explicitly as part of their input. However; in LSTM and GRU based architectures, data is provided in series and they are expected to remember past observations. We observe that the architectures with internal memories (LSTM and GRU based networks) have the largest errors in both metrics. The reasons why LSTM and GRU based networks perform poorly on 3D reconstruction might be that (1) the number of epoch or (2) memory units are not enough (it was 500 epoch and 750 memory unit for LSTM and GRU in this study) for them to learn hidden representations or (3) they are not suited for auto-regression type problems as stated in [149].

Increasing the number of training epochs or using a stack of LSTMs or GRUs may have improved the results but the network sizes are quite large as in the current configuration. Prediction time increases as the number of parameters are increased. Thus, we do not add more layers to memory-based networks. We perform an additional test by increasing the number of units in LSTMs and GRUs to 1500 and 3000, but it does not improve the reconstruction performance for at least learning the vessel tree structure.

We depict the reconstruction performances of each network in Figure 5.12. It is clear from the figure that LSTM and GRU based networks produce worse predictions than only CNN based networks. We observe that they fail to learn to estimate 3D positions of centers and corresponding radiuses of coronaries. The reasons why LSTM and GRU based networks failed are discussed previously in the above paragraphs.
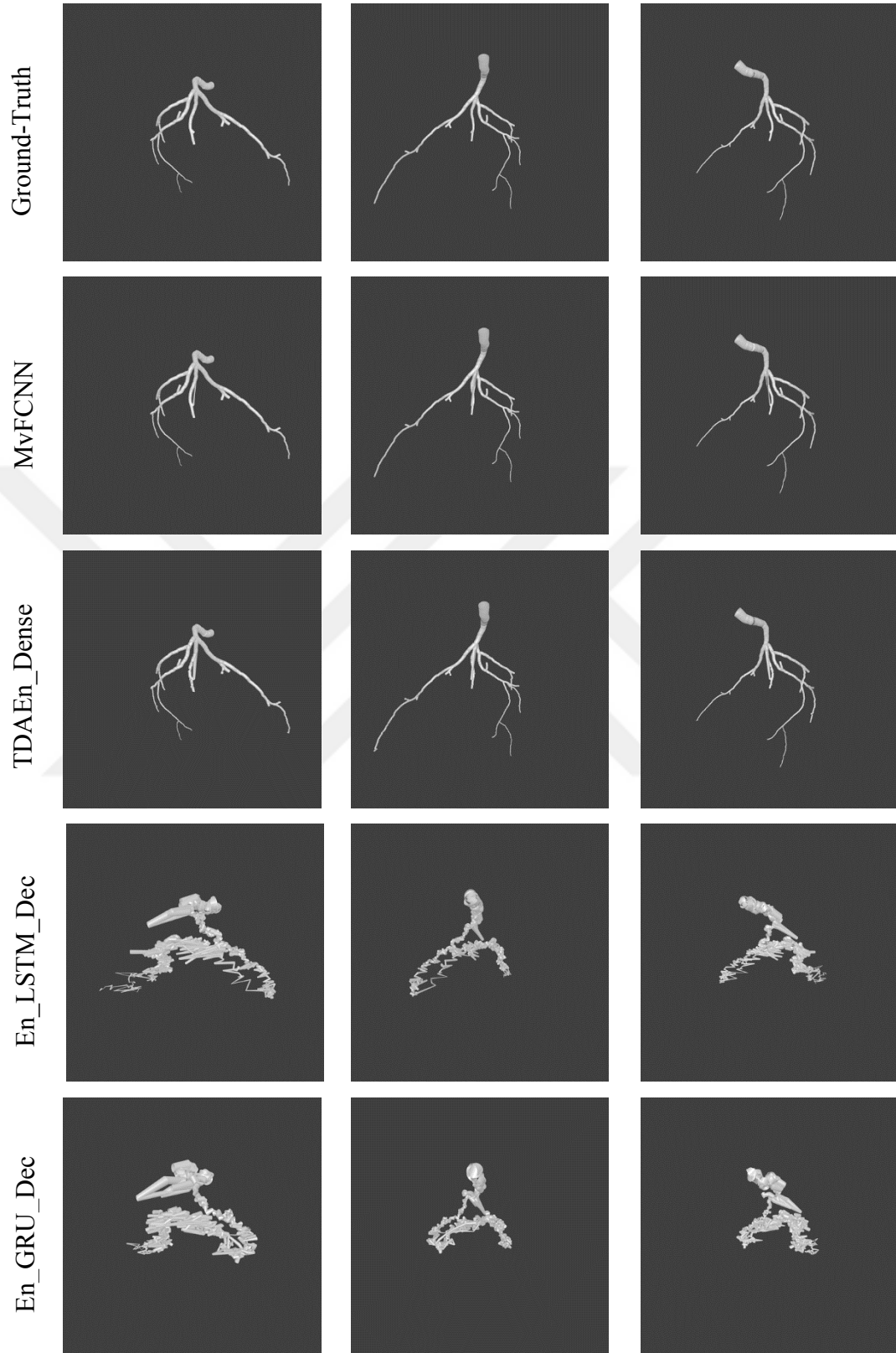
Figure 5.12 The prediction of proposed networks for test data 00015. The columns show the same object from different views.

We are inspired by a recently presented study [150] to improve 3D reconstruction performance. The authors in [150] claim that common activation functions such as ReLU or *tanh* non-linearities fail to capture high-frequency details present in natural signals. Instead, they propose using a periodic activation function (sine-activation). They show that results are improved in many tasks including 3D scene representation, image and sound generation, solving wave equations, etc. in a feed-forward network. Therefore, we train the same networks with changing Leaky-ReLU activations to sine-activations. Table 5.8 summarizes the results produced by sine non-linearity function.

Table 5.8 Reconstruction performances of networks with sine-activation functions proposed in this study.

Model Name	Chamfer Distance		Radius Error	
	μ	σ	μ	σ
MvFCNN	5.56192181	1.803033493	27.61617898	24.45407693
TDAEn_Dense	3.67000E-06	5.96833E-07	4.36000E-13	2.86247E-13
En_LSTM_Dec	6.967926226	0.308885457	3.450022877	0.681069428
En_GRU_Dec	6.696995766	0.329006134	3.565043504	0.719475302

We observe from Table 5.8 that using sine-activation improves 3D reconstruction performance on TDAEn_Dense network significantly, but the results are not improved MvFCNN and memory-based networks. On the contrary, these networks perform worse.

5.2.2 Runtime Comparison for Predictions

The experiments for running time have been implemented on two Linux-based machines. The first one is a notebook computer with the specifications provided in Section 5.1.4. We use this machine for testing CPU-based platforms. The second one is a server machine where training is achieved. The hardware specifications are given for this machine in Section 4.1.6. We use this machine for testing GPU-based platforms.

We measure prediction time on test data for each network and calculate its average time. Table 5.9 reports the average running-time on CPU and GPU based platforms for a single prediction.

Table 5.9 Run-time comparison of proposed networks on CPU and GPU based platforms.

Model Name	CPU (seconds)	GPU (seconds)	Ratio (CPU / GPU)
MvFCNN	0.033588	0.001923	17.47
TDAEn_Dense	0.120523	0.004286	28.12
En_LSTM_Dec	0.243891	0.005843	41.74
En_GRU_Dec	0.176236	0.003330	52.93

Since memory-based (LSTM, GRU) networks are larger (more complex, more weights, more calculation), they produce predictions slower than the other two networks. We observe that GRUs are faster than LSTMs as expected because GRUs use less gating mechanisms as mentioned earlier in Section 4.2.4.4. If we compare run-time for different platforms, GPU based platform produces at least 17.74 times faster than CPU based platform on average. This gain increases to 52.93 for larger and more complex deep networks. The fastest network appears to be MvFCNN with approximately 30 and 520 predictions per second on CPU and GPU, respectively.

CHAPTER 6

CONCLUSION

In this thesis, we have proposed a fully automated 3D reconstruction pipeline for coronary artery tree from 2D X-Ray angiograms. The pipeline mainly includes 2 steps. Namely, the segmentation of coronary vessels from their background and utilize these images in 3D reconstruction network to obtain an individual 3D model of these coronaries.

In the first step of the pipeline, we have proposed fully convolutional deep network architecture, called Sine-Net, for the segmentation of blood vessels. Since there are no publicly available XRA images, we adapted retinal blood vessels in our work to train the network. Sine-Net architecture is novel to the literature in its construction style, which it applies down-sampling after up-sampling with strided convolution operation. We have evaluated the segmentation performance of this network quantitatively on publicly available STARE, DRIVE and CHASE_DB1 databases in terms of specificity, accuracy, sensitivity and AUC metrics. The proposed network shows superior performance on the segmentation of blood vessels and outperforms the-state-of-the-art methods in the literature as presented in Section 5.1.5. We have also demonstrated the effect of preprocessing on input signals with additional tests performed on the network. From our tests, we conclude that the segmentation performance, in general, is slightly improved on the sensitivity metric for the pre-processed input, but it shows a very limited decrease in some of the other metrics. We can tell that there is a trade-off. The results have been proved that the network can also produce predictions well without pre-processing. In addition, we introduce hybrid-database training to generalize network performance on blood vessel segmentation with combining all databases and measure execution time for the segmentation. The network has the potential to be used in clinics thanks to its decent execution time, high accuracy and robustness.

In the second step of the pipeline, we have proposed 4 different deep network architectures for solving 3D reconstruction problem of coronary artery trees. Previously, this problem is solved using analytical methods in related literature, but such methods have some limitations discussed earlier in Section 2.3.1. We have solved the same problem with the learning-based method. In this study, we demonstrate that coronary arteries can be reconstructed using 3D deep network for the first time. We have proposed variations of 3D-R2N2 network [37] and multi-view CNN based networks. 3D-R2N2 network produces voxel grids and the resolution is very low and not appropriate for our problem. Thus, we modify object representation and present well-defined data-structure for 3D representation of objects in tubular/cylindrical shapes described in Section 4.2.1. The networks have learned this representation and showed promising results. Training is achieved using synthetic data we have produced in Section 4.1.2. We prove the validation of the data-structure and architectures proposed in this study for 3D reconstruction of coronaries. The input to networks is multi-view images of coronary vessels in black and white (segmented) format and the output is the data structure we have defined in Section 4.2.1.

In this study, 3D reconstruction task is designed as a regression problem in which output contains continuous values. We conduct experimental tests to quantitatively evaluate and compare the performance of networks. We observe that only CNN-based networks yield better 3D reconstruction performances in terms of CD and MSE metrics. The results for memory-based networks (LSTMs and GRUs) worse than for only CNN-based networks and they fail to construct coronary tree shapes. We consider this is because memory-based networks must remember past observations thanks to its internal cells whereas CNN-based networks are provided this data explicitly. We conclude that LSTMs and GRUs are not appropriate for auto-regression problems as stated in [149].

We show and validate our 3D reconstruction pipeline using synthetic data-set produced from real subjects. We believe that 3D reconstruction with deep learning architecture can be improved further as with the evaluation of technologies and

variants of network architectures. There is no doubt that the topic becomes a trending problem shortly which is already started.

6.1 Future Work

In this study, we proposed a 3D reconstruction pipeline and proved the validation of our method using 3D synthetic data-set. In the future, the first plan is to implement the same networks on real data using transfer learning. Next, we try to optimize the pipeline to speed up execution time on CPU-based low-cost computers and make it reachable for every clinic. To realize this plan, networks can be combined and simplified to build in a single network. In the current configuration of the pipeline, we have two networks stacked together. The first one segments vessels and the second one uses segmented images to produce 3D model. This approach takes less than 10 seconds (9.74s) on CPU and takes 0.355 seconds on Tesla P100 GPU. We would like to emphasize here that most of the time for prediction is exhausted in the segmentation task (9.73s on CPU, 0.353s on GPU). Thus, it would be better to remove the segmentation step from the method and propose a single network directly for 3D reconstruction.

The third plan is that optimizing the network further using integer weights while preserving the accuracy of the network. This process will increase the speed and decrease the network size.

REFERENCES

[1]   "World Health Organization," [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).   [Accessed   3   July   2020].

[2]   M. Demiray, "A Novel method for the quantification of coronary artery stenosis: a 2D QCA system," MIDDLE EAST TECHNICAL UNIVERSITY, Ankara, 2015.

[3]   D. Hahn and E. Kuon, "Medical Radiation," [Online]. Available: https://www.medicalradiation.com/types-of-medical-imaging/imaging-using-x-rays/angiography/. [Accessed 05 01 2020].

[4]   H. R. Fazlali, N. Karimi, S. M. R. Soroushmehr, S. Sinha, S. Samavi, B. Nallamothu and K. Najarian, "Vessel region detection in coronary X-ray angiograms," *IEEE International Conference on Image Processing,* pp. 1493-1497, 2015.

[5]   M. Nichols, N. Townsend, P. Scarborough and M. Rayner, "Cardiovascular disease in Europe: epidemiological update," *European Heart Journal,* vol. 34, no. 39, pp. 3028-3034, 2013.

[6]   S. S. Virani, A. Alanso, E. J. Benjamin, M. S. Bittencourt and e. al., "Heart Disease and Stroke Statistics—2020 Update: A Report From the American Heart Association," *AHA STATISTICAL UPDATE,* vol. 141, no. 9, pp. e139-e596, 2020.

[7]   D. Mark, D. Berman, M. Budoff, J. Carr, T. Gerber, H. Hecht, M. Hlatky, J. Hodgson,   M.   Lauer   and   J.   Miller, "ACCF/ACR/AHA/NASCI/SAIP/SCAI/SCCT   2010   expert   consensus document on coronary computed tomographic angiography: a report of the American College of Cardiology Foundation Task Force on Expert Consensus

Documents," *Journal of the American College of Cardiology,* vol. 55, no. 23, pp. 2663-2699, 2010.

[8] M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. Rudnicka, C. Owen and S. Barman, "Blood vessel segmentation methodologies in retinal images a survey," *Computer Methods and Programs in Biomedicine,* vol. 108, no. 1, pp. 407-433, 2012.

[9] S. Moccia, E. D. Momi, S. E. Hadji and L. S. Mattos, "Blood vessel segmentation algorithms—review of methods, datasets and evaluation metrics," *Computer methods and programs in biomedicine,* vol. 158, pp. 71-91, 2018.

[10] K. S. Sreejini and V. K. Govindan, "Retrieval of pathological retina images using Bag of Visual Words and pLSA model.," *Engineering Science and Technology, an International Journal,* vol. 22, no. 3, pp. 777-785, 2019.

[11] Z. Jiang, H. Zhang, Y. Wang and S. B. Ko, "Retinal blood vessel segmentation using fully convolutional network with transfer learning," *Computerized Medical Imaging and Graphics,* vol. 68, pp. 1-15, 2018.

[12] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical," *International Conference on Medical image computing and computer-assisted intervention,* pp. 234-241, 2016.

[13] İ. Atli and O. Gedik, "VESUNETDeep: A Fully Convolutional Deep Learning Architecture for Automated Vessel Segmentation," in *27th Signal Processing and Communications Applications Conference*, Sivas, 2019.

[14] R. Smithuis and T. Willems, "Radiology Assistant," [Online]. Available: https://radiologyassistant.nl/cardiovascular/coronary-anatomy-and-anomalies. [Accessed 15 1 2020].

[15] A. Fathala, "Potential role of coronary computed tomography angiogram in cardiac preoperative evaluation," *Saudi journal of anaesthesia,* vol. 5, no. 3, p. 342, 2011.

[16] E. Verna, M. Santarone, M. Boscarini, I. Ghezzi and S. Repetto, "Unusual origin and course of the first septal branch of the left coronary artery: angiographic recognition," *Cardiovascular and interventional radiology,* vol. 11, no. 3, pp. 146-149, 1998.

[17] "Atlas of Human Cardiac Anatomy," University of Minnesota, [Online]. Available: http://www.vhlab.umn.edu/atlas/coronary-arteries/right-coronary-artery/index.shtml. [Accessed 16 1 2020].

[18] "Radiology Key," 29 Jun 2017. [Online]. Available: https://radiologykey.com/anatomy-5/. [Accessed 13 Feb 2020].

[19] MayoClinic, "Coronary artery disease," [Online]. Available: https://www.mayoclinic.org/diseases-conditions/coronary-artery-disease/symptoms-causes/syc-20350613. [Accessed 13 Feb 2020].

[20] A. Felman, "Medical News Today," 22 Jan 2020. [Online]. Available: https://www.medicalnewstoday.com/articles/237191. [Accessed 20 Feb 2020].

[21] I. I. Abubakar, T. Tillmann and A. Banerjee, "Global, regional, and national age-sex specific all-cause and cause-specific mortality for 240 causes of death, 1990-2013: a systematic analysis for the Global Burden of Disease Study 2013," *Lancet,* vol. 385, no. 9963, pp. 117-171, 2015.

[22] U. D. o. H. &. H. Services, "What Are the Signs and Symptoms of Coronary Heart Disease?," [Online]. Available: https://www.nhlbi.nih.gov/health-topics/coronary-heart-disease#signs-symptoms-and-complications. [Accessed 13 Feb 2020].

[23] "MayoClinic," [Online]. Available: https://www.mayoclinic.org/diseases-conditions/heart-failure/symptoms-causes/syc-20373142. [Accessed 20 Feb 2020].

[24] F. F. Ferri, Ferri's Clinical Advisor 5 Books in 1, Elsevier Health Sciences, 2017.

[25] K. Patel, J. Tarkin, P. W. Serruys, E. Tenekecioglu, N. Foin, Y.-J. Zhang, T. Crake, J. Moon, A. Mathur and C. V. Bourantas, "Invasive or non-invasive imaging for detecting high-risk coronary lesions?," *Expert review of cardiovascular therapy,* vol. 15, no. 3, pp. 165-179, 2017.

[26] J. D. Schuijf, J. M. van Werkhoven, G. Pundziute, J. W. Jukema, I. Decramer, M. P. Stokkel, P. Dibbets-Schneider, M. J. Schalij, J. H. C. Reiber, E. E. v. d. Wall, W. Wijns and J. J. Bax, "Invasive Versus Noninvasive Evaluation of Coronary Artery Disease," *JACC: Cardiovascular Imaging,* vol. 1, no. 2, pp. 190-199, 2008.

[27] J. D. Schuijf, L. J. Shaw, W. Wijns, H. J. Lamb, D. Poldermans, A. .. Roos, E. E. v. d. Wall and J. J. Bax, "Cardiac imaging in coronary artery disease: differing modalities," *Heart,* vol. 91, no. 8, pp. 1110-1117, 2005.

[28] P. D. Adamson and D. E. Newby, "Non-invasive imaging of the coronary arteries," *European Heart Journal,* vol. 40, no. 29, pp. 2444-2454, 2019.

[29] "RadiologyInfo.org For Patients," [Online]. Available: www.radiologyinfo.org. [Accessed 23 Feb 2020].

[30] "Heart Disease," Centers for Disease Control and Prevention, [Online]. Available: https://www.cdc.gov/heartdisease/coronary_ad.htm. [Accessed 22 Feb 2020].

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[32] F.-F. Li, J. Johnson and S. Yeung, "Standford Online Course Material Lecture 10," Stanford University, San Francisco, 2017.

[33] N. Donges, "Recurrent Neural Networks and LSTM," 25 Feb 2018. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5. [Accessed 15 Jan 2019].

[34] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv preprint arXiv:1406.1078,* 2014.

[35] J. Chung, Gulcehre, C. K. C. and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555,* 2014.

[36] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.

[37] C. B. Choy, D. Xu, J. Gwak, K. Chen and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," *European conference on computer vision,* pp. 628-644, 2016.

[38] J. K. Kim, S. Y. Kim and S. H. Lee, "New Proposal for Adlt's Normal Sinus Heart Rates," *Journal of the Korean Academy of Family Medicine,* vol. 19, no. 2, pp. 134-140, 1998.

[39] S. Çimen, A. Gooya, M. Grass and A. F. Frangi, "Reconstruction of coronary arteries from X-ray angiography:A review," *Medical Image Analysis,* vol. 32, pp. 46-88, 2016.

[40] N. E. Green, S. Y. J. Chen, A. R. H. J. C. Messenger, B. M. Groves and J. D. Carroll, "Angiographic views used for percutaneous coronary interventionsA three-dimensional analysis of physician-determined vs. computer-generated views," *Catheterization and cardiovascular interventions,* vol. 64, no. 4, pp. 451-459, 2005.

[41] R. Rigamonti and V. Lepetit, "Accurate and efficient linear structure segmentation by leveraging ad hoc features with learned filters," *International Conference on Medical Image Computing and Computer-Assisted Intervention,* pp. 189-197, 2012.

[42] C. Kirbas and F. Quek, "A review of vessel extraction techniques and algorithms," *ACM Computing Surveys (CSUR),* vol. 36, no. 2, pp. 81-121, 2004.

[43] S. Akbar, M. Sharif, M. U. Akram, T. Saba, T. Mahmood and M. Kolivand, "Automated techniques for blood vessels segmentation through fundus retinal images: A review," *Microscopy research and technique,* vol. 82, no. 2, pp. 153-170, 2019.

[44] A. Frangi, W. Niessen, K. Vincken and M. Viergever, "Multiscale vessel enhancement filtering," *Medical Image Computing and Computer-Assisted Intervention,* pp. 130-137, 1998.

[45] M. Law and A. Chung, "Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux," *Computer Vision–ECCV,* pp. 368-382, 2008.

[46] A. Sironi, E. Türetken, V. Lepetit and P. Fua, "Multiscale centerline detection," *IEEE transactions on pattern analysis and machine intelligence,* vol. 38, no. 7, pp. 1327-1341, 2016.

[47] A. Kerkeni, A. Benabdallah, A. Manzanera and M. H. Bedoui, "A coronary artery segmentation method based on multiscale analysis and region growing," *Computerized Medical Imaging and Graphics,* vol. 48, pp. 49-61, 2016.

[48] S. Mabrouk, C. Oueslati and F. Ghorbel, "Multiscale Graph Cuts Based Method for Coronary Artery Segmentation in Angiograms," *IRBM,* p. In press, 2017.

[49] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Greig, S. Yoshida and R. Kikinis, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical image analysis,* vol. 2, no. 2, pp. 143-168, 1998.

[50] T. M. Koller, G. Gerig, G. Szekely and D. Dettwiler, "Multiscale detection of curvilinear structures in 2-D and 3-D image data," *IEEE ICCV,* pp. 864-869, 1995.

[51] H. Mirzaalian and G. Hamarneh, "Vessel scale-selection using MRF optimization," *Computer Vision and Pattern Recognition,* pp. 3273-3279, 2010.

[52] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever and B. v. Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging,* vol. 23, pp. 501-509, 2004.

[53] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: a survey," *Medical image analysis,* vol. 1, no. 2, pp. 91-108, 1996.

[54] D. Lesage, E. D. Angelini, I. Bloch and G. Funka-Lea, "A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes," *Medical image analysis,* vol. 13, no. 6, pp. 819-845, 2009.

[55] P. J. Yim, J. J. Cebral, R. Mullick, H. B. Marcos and P. L. Choyke, "Vessel surface reconstruction with a tubular deformable model," *IEEE transactions on medical imaging,* vol. 20, no. 12, pp. 1411-1421, 2001.

[56] J. Mille, R. Boné and L. Cohen, "Region-based 2D deformable generalized cylinder for narrow structures segmentation," *2008,* pp. 392-404, Computer Vision–ECCV.

[57] D. Geiger, L. C. A. Gupta and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *PAMI,* vol. 17, no. 3, pp. 294-302, 1995.

[58] J. Brieva, M. Galvez and C. Toumoulin, "Coronary extraction and stenosis quantification in X-ray angiographic imaging," *Engineering in Medicine and Biology Society, IEMBS '04. 26th Annual International Conference of the IEEE* , vol. 1, pp. 1714-1717, 2004.

[59] L. M. Lorigo, O. D. Faugeras, W. E. L. Grimson, R. Keriven, R. Kikinis, A. Nabavi and C. F. Westin, "Curves: curve evolution for vessel segmentation," *Medical image analysis,* vol. 5, no. 3, pp. 195-206, 2001.

[60] M. S. Hassouna, A. A. Farag, S. Hushek and T. Moriarty, "Cerebrovascular segmentation from TOF using stochastic models," *Medical Image Analysis,* vol. 10, no. 1, pp. 2-18, 2006.

[61] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine,* vol. 13, no. 6, pp. 47-60, 1996.

[62] E. Goceri, Z. K. Shah and M. N. Gurcan, "Vessel segmentation from abdominal magnetic resonance images: adaptive and reconstructive approach," *International journal for numerical methods in biomedical engineering,* vol. 33, no. 4, pp. 1-16, 2017.

[63] P. Bruyninckx, D. Loeckx, D. Vandermeulen and P. Suetens, "Segmentation of liver portal veins by global optimization," *International Society for Optics and Photonics,* vol. 7624, p. 76241Z, 2010.

[64] B. P., L. D., V. D. and S. P., "Segmentation of lung vessel trees by global optimization," *International Society for Optics and Photonics,* vol. 7259, p. 725912, 2009.

[65] A. Asad, A. Azar and A. Hassanien, "A new heuristic function of ant colony system for retinal vessel segmentation," *International Journal of Rough Sets and Data Analysis,* vol. 1, no. 2, pp. 15-30, 2014.

[66] T. Mapayi, J. Tapamo and S. Viriri, "Retinal vessel segmentation: a comparative study of fuzzy C-means and sum entropy information on phase congruency," *International Journal of Advanced Robotic Systems,* vol. 12, no. 9, p. 133, 2015.

[67] K. S. Sreejini and V. K. Govindan, "Improved multiscale matched filter for retina vessel segmentation using PSO algorithm," *Egyptian Informatics Journal,* vol. 16, no. 3, pp. 253-260, 2015.

[68] L. Xu and S. Luo, "A novel method for blood vessel detection from retinal images," *Biomedical engineering online,* vol. 9, no. 1, p. 14, 2010.

[69] E. Ricci and R. Perfetti, "Retinal blood vessel segmentation using line operators and support vector classification," *IEEE transactions on medical imaging,* vol. 26, no. 10, pp. 1357-1365, 2007.

[70] J. I. Orlando and M. Blaschko, "Learning fully-connected CRFs for blood vessel segmentation in retinal images," in *Springer*, Cham, 2014.

[71] C. A. Lupascu, D. Tegolo and E. Trucco, "FABC: retinal vessel segmentation using AdaBoost," *IEEE Transactions on Information Technology in Biomedicine,* vol. 14, no. 5, pp. 1267-1274, 2010.

[72] J. V. Soares, J. J. Leandro, R. M. Cesar, H. F. Jelinek and M. J. Cree, "Retinal vessel segmentation using the 2-D Gabor wavelet and supervised

classification," *IEEE Transactions on medical Imaging,* vol. 25, no. 9, pp. 1214-1222, 2006.

[73] D. Marín, A. Aquino, M. E. Gegúndez-Arias and J. M. Bravo, "A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features," *IEEE Transactions on medical imaging,* vol. 30, no. 1, pp. 146-158, 2011.

[74] C. Sinthanayothin, J. F. Boyce, H. L. Cook and T. H. Williamson, "Automated localisation of the optic disc, fovea and retinal blood vessels from digital colour fundus images," *British Journal of Ophthalmology,* vol. 83, no. 8, pp. 902-910, 1999.

[75] R. Nekovei and Y. Sun, "Back-propagation network and its configuration for blood vessel detection in angiograms," *IEEE Transactions on Neural Networks,* vol. 6, no. 1, pp. 64-72, 1995.

[76] Q. Li, B. Feng, L. Xie, P. Liang, H. Zhang and T. Wang, "A cross-modality learning approach for vessel segmentation in retinal images," *IEEE transactions on medical imaging,* vol. 35, no. 1, pp. 109-118, 2015.

[77] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," *Advances in neural information processing systems,* pp. 109-117, 2011.

[78] S. Wang, Y. Yin, G. Cao, B. Wei, Y. Zheng and G. Yang, "Hierarchical retinal blood vessel segmentation based on feature and ensemble learning," *Neurocomputing,* vol. 149, pp. 708-717, 2015.

[79] A. F. Khalaf, I. A. Yassine and A. S. Fahmy, "Convolutional neural networks for deep feature learning in retinal vessel segmentation," *IEEE International Conference on Image Processing (ICIP),* pp. 385-388, 2016.

[80] H. Fu, Y. Xu, D. W. K. Wong and J. Liu, "Retinal vessel segmentation via deep learning network and fully-connected conditional random fields," *Biomedical Imaging,* pp. 698-701, 2016.

[81] S. Xie and Z. Tu, "Holistically-nested edge detection," *Proceedings of the IEEE international conference on computer vision,* pp. 1395-1403, 2015.

[82] J. Mo and L. Zhang, "Multi-level deep supervised networks for retinal vessel segmentation," *International journal of computer assisted radiology and surgery,* vol. 12, no. 12, pp. 2181-2193, 2017.

[83] K. Hu, Z. Zhang, X. Niu, Y. Zhang, C. Cao, F. Xiao and X. Gao, "Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function," *Neurocomputing,* pp. 179-191, 2018.

[84] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS,* 2012.

[85] Y. Guo, Ü. Budak and A. Şengür, "A novel retinal vessel detection approach based on multiple deep convolution neural networks," *Computer methods and programs in biomedicine,* vol. 167, pp. 43-48, 2018.

[86] P. Liskowski and K. Krawiec, "Segmenting Retinal Blood Vessels With Deep Neural Networks," *IEEE transactions on medical imaging,* vol. 35, no. 11, pp. 2369-2380, 2016.

[87] A. Dasgupta and S. Singh., "A fully convolutional neural network based structured prediction approach towards the retinal vessel segmentation," *IEEE 14th International Symposium on Biomedical Imaging,* pp. 248-251, 2017.

[88] Z. Feng, J. Yang and L. Yao, "Patch-based fully convolutional neural network with skip connections for retinal blood vessel segmentation," in *IEEE International Conference on Image Processing (ICIP)*, 2017.

[89] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 3431-3440, 2015.

[90] J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *CVPR,* 2015.

[91] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 4700-4708, 2017.

[92] S. Feng, Z. Zhuo, D. Pan and Q. Tian, "CcNet: A cross-connected convolutional network for segmenting retinal vessels using multi-scale features," *Neurocomputing,* 2019.

[93] İ. Atli and O. S. Gedik, "VESUNETDeep:A Fully Convolutional Deep Learning Architecture for Automated Vessel Segmentation," *2019 27th Signal Processing and Communications Applications Conference,* pp. 1-4, 2019.

[94] Y. Zhang and A. C. Chung, "Deep supervision with additional labels for retinal vessel segmentation task," *International conference on medical image computing and computer-assisted intervention,* pp. 83-91, 2018.

[95] T. B. Sekou, M. Hidane, J. Olivier and H. Cardot, "Retinal blood vessel segmentation using a fully convolutional network--transfer learning from patch-to image-level," *International Workshop on Machine Learning in Medical Imaging,* pp. 170-178, 2018.

[96] T. A. Soomro, A. A. J., L. Zheng, S. Soomro, J. Gao, O. Hellwich and M. Paul, "Deep learning models for retinal blood vessels segmentation: A review," *IEEE Access,* vol. 7, pp. 71696-71717, 2019.

[97] S. Zheng, T. Meiying and S. Jian, "Sequential reconstruction of vessel skeletons from X-ray coronary angiographic sequences," *Computerized Medical Imaging and Graphics ,* vol. 34, no. 5, pp. 333-345, 2010.

[98] W. Cong, J. Yang, D. Ai, Y. Chen, Y. Liu and Y. Wang, "Quantitative analysis of deformable model-based 3-D reconstruction of coronary artery from multiple angiograms," *IEEE Transactions on Biomedical Engineering,* vol. 62, no. 8, pp. 2079-2090, 2015.

[99] J. Yang, W. Cong, Y. Chen, J. Fan, Y. Liu and Y. Wang, "External force back-projective composition and globally deformable optimization for 3-D coronary artery reconstruction," *Physics in Medicine & Biology,* vol. 59, no. 4, pp. 975-1003, 2014.

[100] S. J. Chen and J. D. Carroll, "3-D reconstruction of coronary arterial tree to optimize angiographic visualization," *IEEE transactions on medical imaging,* vol. 19, no. 4, pp. 318-336, 2000.

[101] S. Y. Chen and J. D. Carroll, "Kinematic and deformation analysis of 4-D coronary arterial trees reconstructed from cine angiograms," *IEEE transactions on medical imaging,* vol. 22, no. 6, pp. 710-721, 2003.

[102] A. Andriotis, A. Zifan, M. Gavaises, P. Liatsis, I. Pantos, A. Theodorakakos, E. P. Efstathopoulos and D. Katritsis, "A new method of three-dimensional coronary artery reconstruction from X-ray angiography: Validation against a virtual phantom and multislice computed tomography," *Catheterization and cardiovascular interventions,* vol. 71, no. 1, pp. 28-43, 2008.

[103] J. Yang, Y. Wang, Y. Liu, S. Tang and W. Chen, "Novel approach for 3-D reconstruction of coronary arteries from two uncalibrated angiographic images," *IEEE Transactions on Image Processing,* vol. 18, no. 7, pp. 1563-1572, 2009.

[104] X. Liu, F. Hou, A. Hao and H. Qin, "A parallelized 4D reconstruction algorithm for vascular structures and motions based on energy optimization," *The Visual Computer,* vol. 31, no. 11, pp. 1431-1446, 2015.

[105] L. Sarry and J.-Y. Boire, "Three-dimensional tracking of coronary arteries from biplane angiographic sequences using parametrically deformable models," *IEEE transactions on medical imaging,* vol. 20, no. 12, pp. 1341-1351, 2001.

[106] C. Cañero, F. Vilariño, J. Mauri and P. Radeva, "Predictive (Un)distortion model and 3-D reconstruction by biplane snakes," *IEEE Transactions on Medical Imaging,* vol. 21, no. 9, pp. 1188-1201, 2002.

[107] K. Hoffmann, A. Sen, L. Lan, K. G. Chua, J. Esthappan and M. Mazzucco, "A system for determination of 3D vessel tree centerlines from biplane images," *International Journal of Cardiac Imaging,* vol. 16, no. 5, pp. 315-330, 2000.

[108] G. Shechter, F. Devernay, E. Coste-Manière, A. Quyyumi and E. R. McVeigh, "Three-dimensional motion tracking of coronary arteries in biplane cineangiograms," *IEEE Transactions on Medical Imaging,* vol. 22, no. 4, pp. 493-503, 2003.

[109] P. Fallavollita and F. Cheriet, "Optimal 3D reconstruction of coronary arteries for 3D clinical assessment," *Computerized Medical Imaging and Graphics,* vol. 32, no. 6, pp. 476-487, 2008.

[110] W. K., B. K., N. N., D. P., B. T., S. O. and S. W., "Enhance3-D-reconstruction algorithm for C-arm systems suitable for interventional procedures," *IEEE transactions on medical imaging,* vol. 19, no. 5, pp. 391-403, 2000.

[111] B. M. t. H. R. D. Ruijters and P. Suetens, "Vesselness-based 2D-3D registration of the coronary arteries," *International journal of computer assisted radiology and surgery,* vol. 4, no. 4, pp. 391-397, 2009.

[112] U. Jandt, D. Schäfer, M. Grass and V. Rasche, "Automatic generation of 3D coronary artery centerlines using rotational X-ray angiography," *Medical image analysis,* vol. 13, no. 6, pp. 846-858, 2009.

[113] R. Liao, D. Luc, Y. Sun and K. Kirchberg, "3-D reconstruction of the coronary artery tree from multiple views of a rotational X-ray angiography," *The International Journal of Cardiovascular Imaging,* vol. 26, no. 7, pp. 733-749, 2010.

[114] U. Jandt, D. Schäfer, M. Grass and V. Rasche, "Automatic generation of time resolved motion vector fields of coronary arteries and 4D surface extraction using rotational x-ray angiography," *Physics in Medicine & Biology,* vol. 54, no. 1, pp. 45-64, 2008.

[115] B. Movassaghi, V. Rasche, M. Grass, M. A. Viergever and W. J. Niessen, "A quantitative analysis of 3-D coronary modeling from two or more projection images," *IEEE transactions on medical imaging,* vol. 23, no. 12, pp. 1517-1531, 2004.

[116] P. H. Torr and D. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International journal of computer vision,* vol. 24, no. 3, pp. 271-300, 1997.

[117] K. L. Navaneet, P. Mandikal, M. Agarwal and R. V. Babu, "CAPNet: Continuous approximation projection for 3D point cloud reconstruction using 2D supervision," in *AAAI Conference on Artificial Intelligence*, 2019.

[118] H. Fan, H. Su and L. J. Guibas, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," in *IEEE conference on computer vision and pattern recognition*, 2017.

[119] M. Zamorski, M. Zięba, P. Klukowski, R. Nowak, K. Kurach, W. Stokowiec and T. Trzciński, "Adversarial Autoencoders for Compact Representations of 3D Point Clouds," *arXiv,* 2018.

[120] P. Mandikal, M. Agarwal and R. V. Babu, "CAPNet: Continuous Approximation Projection For 3D Point Cloud Reconstruction Using 2D Supervision," arXiv preprint arXiv:1811.11731, 2018.

[121] Z. Lun, M. Gadelha, E. Kalogerakis, S. Maji and R. Wang, "3D shape reconstruction from sketches via multi-view convolutional networks," in *International Conference on 3D Vision (3DV)*, 2017.

[122] A. Arsalan Soltani, H. Huang, J. Wu, T. D. Kulkarni and J. B. Tenenbaum, "Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks," in *IEEE conference on computer vision and pattern recognition*, Honolulu, 2017.

[123] G. Riegler, A. Osman Ulusoy and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 3577-3586, 2017.

[124] M. Tatarchenko, A. Dosovitskiy and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," *Proc. of the IEEE International Conf. on Computer Vision (ICCV),* vol. 2, p. 8, 2017.

[125] X. Yan, J. Yang, E. Yumer, Y. Guo and H. Lee, "Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision," *Advances in Neural Information Processing Systems,* pp. 1696-1704, 2016.

[126] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Transactions on Graphics,* vol. 36, no. 4, p. 72, 2017.

[127] H. Xie, H. Yao, X. Sun, S. Zhou and X. Tong, "Weighted voxel: a novel voxel representation for 3D reconstruction," in *10th International Conference on Internet Multimedia Computing and Service*, 2018.

[128] C. Häne, S. Tulsiani and J. Malik, "Hierarchical surface prediction for 3d object reconstruction," in *International Conference on 3D Vision*, 2017.

[129] D. Paschalidou, O. Ulusoy, C. Schmitt, L. Van Gool and A. Geiger, "Raynet: Learning volumetric 3d reconstruction with ray potentials," in *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 2018.

[130] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu and Y. G. Jiang, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images," in *European Conference on Computer Vision (ECCV)*, 2018.

[131] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson and C. Fookes, "Image2mesh: A learning framework for single image 3d reconstruction," in *Asian Conference on Computer Vision*, 2018.

[132] M. Tatarchenko, A. Dosovitskiy and T. Brox, "Multi-view 3d models from single images with a convolutional network," *European Conference on Computer Vision,* pp. 322-337, 2016.

[133] R. Girdhar, D. F. Fouhey, M. Rodriguez and A. Gupta, "Learning a predictable and generative vector representation for objects," in *European Conference on Computer Vision*, 2016.

[134] A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi and F. Yu, "ShapeNet:An Information-Rich 3D Model Repository. Technical report, Stanford Univesity - Princeton University - Toyota Technological Institute at Chicago," 2015.

[135] M. Everingham, L. Van Gool, C. Williams, J. Winn and A. Zisserman, "The pascal visual object classes challenge 2012," 2011.

[136] D. Meagher, "Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer," *Electrical and Systems Engineering Department Rensseiaer Polytechnic Institute Image Processing Laboratory,* 1980.

[137] "Structured analysis of the retina," [Online]. Available: http://cecas.clemson.edu/. [Accessed 02 01 2020].

[138] "Retinal image database-retinal image analysis," [Online]. Available: https://blogs.kingston.ac.uk/retinal/chasedb1/. [Accessed 02 01 2020].

[139] A. Budai, R. Bock, A. Maier, J. Hornegger and G. Michelson, "Robust vessel segmentation in fundus images," *International Journal of Biomedical Imaging,* 2013.

[140] S. M. Pizer, R. E. Johnston, J. P. Ericksen, B. C. Yankaskas and K. E. Muller, "Contrast-limited adaptive histogram equalization: speed and effectiveness," *In Visualization in Biomedical Computing,* pp. 337-345, 1990.

[141] X. Bai, F. Zhou and B. Xue, "Image enhancement using multi scale image features extracted by top-hat transform," *Optics & Laser Technology,* vol. 44, no. 2, pp. 328-336, 2012.

[142] A. M. Vukicevic, S. Çimen, N. Jagic, G. Jovicic, A. F. Frangi and N. Filipovic, "Three-dimensional reconstruction and NURBS-based structured meshing of coronary arteries from the conventional X-ray angiography projection images," *Scientific reports,* vol. 8, no. 1, pp. 1-20, 2018.

[143] T. Jayalakshmi and A. Santhakumaran, "Statistical normalization and back propagation for classification," *International Journal of Computer Theory and Engineering,* vol. 3, no. 1, pp. 1793-8201, 2011.

[144] F. a. o. Chollet, *Keras,* GitHub, 2015.

[145] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv,* vol. 1412.6980, 2014.

[146] N. Shirish Keskar and R. Socher, "Improving Generalization Performance by Switching from Adam to SGD," *arXiv:1712.07628,* 2017.

[147] P. J. Blanco, Bulant, M. C. A., L. O., G. M. Talou, C. G. Bezerra, P. L. Lemos and R. A. Feijóo, " Comparison of 1D and 3D models for the estimation of fractional flow reserve," *Scientific reports,* vol. 8, no. 1, p. 17275, 2018.

[148] V. Belagiannis, C. Rupprecht, G. Carneiro and N. Navab, "Robust Optimization for Deep Regression," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[149] F. A. Gers, D. Eck and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," *Neural Nets WIRN Vietri-01,* pp. 193-200, 2002.

[150] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell and G. Wetzstein, "Implicit Neural Representations with Periodic Activation Functions," *arXiv preprint arXiv:2006.09661,* 2020.

CURRICULUM VITAE

PERSONAL INFORMATION

İbrahim ATLI, 1989

Faculty of Engineering and Natural Sciences
Ankara Yıldırım Beyazıt University, Ankara, Turkey
E-mail: iatli@ybu.edu.tr

EDUCATION

Ph.D., 2020, Department of Computer Engineering, Ankara Yıldırım Beyazıt University, Ankara, Turkey
GPA: 4.00/4.00
Thesis: A Novel Approach in 3D Reconstruction of Coronary Artery Tree From 2D X-Ray Angiograms

M.S., 2014, Department of Computer Engineering, Karabük University, Karabük, Turkey
GPA:3.57/4.00
Thesis: Development of Educational Power Electronics Computer Package

B.S., 2012, Department of Computer Engineering, Anadolu University, Eskişehir, Turkey
GPA:3.62/4.00

B.S., 2012, Department of Electrical and Electronics Engineering, Anadolu University, Eskişehir, Turkey
GPA:3.74/4.00

WORK EXPERIENCE

Ankara Yıldırım Beyazıt University - Research Assistant *(2014-)*
Department of Computer Engineering

Kuartis Teknoloji ve Danışmanlık Ltd. Şti. – Biomedical Engineer *(2016-2017)*
Research and Development

Beyin Elektrodinamiği Analiz Yazılım ve Bilişim Ticaret Ltd. Şti. – Biomedical Engineer *(2014-2015)*
Research and Development

Karabük University - Research Assistant *(2012-2014)*
Department of Computer Engineering

PUBLICATIONS FROM THESIS

1. ATLI, İbrahim; GEDİK, O. Serdar. The performance evaluation of histogram of oriented gradients as a feature on vessels detection using artificial neural network. In: 2018 26th Signal Processing and Communications Applications Conference (SIU). IEEE, 2018. p. 1-4.

2. ATLI, İbrahim; GEDİK, O. Serdar. VESUNETDeep: A Fully Convolutional Deep Learning Architecture for Automated Vessel Segmentation. In: 2019 27th Signal Processing and Communications Applications Conference (SIU). IEEE, 2019. p. 1-4.

3. ATLI, İbrahim; GEDİK, O. Serdar. Sine-Net: A Fully Convolutional Deep Learning Architecture for Retinal Blood Vessel Segmentation, Engineering Science and Technology, an International Journal, 2020 (Available online)

4. ATLI, İbrahim; GEDİK, O. Serdar. 3D Reconstruction of Coronary Arteries using Deep Networks from Synthetic X-Ray Angiogram Data, (Ready to submit)