Identification of coronary artery disease in CCTA volumes using neural networks

AALBORG
UNIVERSITY

STUDENT REPORT

Group 22gr10-16
Biomedical Engineering and Informatics
Aalborg University
June 1st 2022

School of Medicine and Health
Biomedical Engineering and Informatics
Fredrik Bajers Vej 7
9220 Aalborg Øst
http://www.hst.aau.dk/

Title:

Identification of coronary artery disease in CCTA volumes using neural networks

Project:

Master Thesis in
Biomedical Engineering
and Informatics

Project period:

February 2022 - June 2022

Project group:

22gr10-416

Authors:

Julie Rask Madsen
Mia Hebsgaard Ransborg

Supervisor:

Alex Skovsbo Jørgensen

Sub supervisor:

Samuel Emil Schmidt

Number of total pages: 84
Appendix: A - D
Date of submission: 01.06.22

Abstract:

Background and aim: CAD is the leading cause of death worldwide. CAD is a narrowing or blocked areas in the coronary arteries, which can lead to angina, myocardial ischemia, and myocardial infarction. CCTA is often used as a diagnostic tool for identifying patients with CAD. However, the method has a low specificity that causes 22 - 52% of the patients without CAD to undergo a CAG, which is the invasive golden standard for diagnosing CAD. To reduce the number of unnecessary invasive procedures, it is desired to increase the specificity of CCTA detecting CAD. Thus, this study aimed to investigate to identify patients with CAD based on CCTA volumes and corresponding patient data using deep learning.
Method: To identify patients with CAD in CCTA volumes, a CAD network was used. The CAD network consisted of a CNN and an RNN connected in series to perform sequential frame analysis of the CCTA volumes and a FCNN to analyze patient data associated with development of CAD.
Results: The performance of the CAD network showed an F1 score of 0.51 and a AUROC of 0.51 which indicated that the CAD network made random classifications when classifying the CCTA volumes as either containing CAD or not.
Conclusion: The CAD network had the ability to classify half of the CCTA volumes as containing CAD or not. Thus, further improvements of the CAD network are still needed before the CAD network can be used in clinical practise.

AALBORG UNIVERSITET
STUDENTERRAPPORT

Andet studieår kandidat v/ Insitut for Medicin og Sundhedsteknologi
Sundhedsteknologi
Fredrik Bajers Vej 7
9000 Aalborg
http://www.hst.aau.dk

Titel:

Identifikation af koronararteriesygdom i CCTA-volumener ved hjælp af neurale netværker

Projekt:

Speciale i Sundhedsteknologi

Projekt periode:

Februar 2022 - Juni 2022

Projekt gruppe:

22gr10-416

Deltagere:

Julie Rask Madsen
Mia Hebsgaard Ransborg

Vejleder:

Alex Skovsbo Jørgensen

Bi-vejleder:

Samuel Emil Schmidt

Sidetal: 84
Appendiks: A - D
Afsluttet: 01.06.22

Abstrakt:

Baggrund og formål: CAD er den førende dødsårsag på verdensplan. CAD er indsnævring eller blokerede områder i kranspulsårerne, som kan føre til angina, myokardieiskæmi eller myokardieinfarkt. CCTA bruges ofte som et diagnostisk værktøj til at identificere patienter med CAD. Metoden har dog en lav specificitet, der fører til at 22 - 52% af patienterne uden CAD får foretaget en KAG, som er den invasive gyldne standard for diagnosticering af CAD. For at reducere antallet af unødvendige invasive procedurer, skal specificiteten af CCTA øges i forhold til detektering af CAD. Dette studie havde således til formål at identificere patienter med CAD baseret på CCTA-volumener og tilsvarende patientdata ved hjælp af deep learning.
Metode: For at identificere patienter med CAD i CCTA-volumener blev et CAD-netværk udviklet. CAD-netværket bestod af et CNN og et RNN forbundet i serie til at udføre sekventiel analyse af CCTA-volumenerne og en FCNN til at analysere patientdata forbundet med udvikling af CAD.
Resultater: CAD-netværkets performance viste en F1 score på 0,51 og en AUROC på 0,51, hvilket indikerede, at CAD-netværket lavede tilfældige klassifikationer, når det klassificerede CCTA-volumener som enten indeholdende CAD eller ej.
Konklusion: CAD-netværket klassificerede kun halvdelen af CCTA-volumenerne med enten CAD eller ej korrekt. Der er således stadig behov for yderligere forbedringer af CAD-netværket, for at CAD-netværket kan bruges i klinisk praksis.

Preface

This study was performed by group 22gr10-416, as a part of a Master's thesis in Biomedical Engineering and Informatics, at Aalborg University. The master thesis investigated *Identification of coronary artery disease in CCTA volumes using neural networks*. It was performed in collaboration with supervisor Alex Skovsbo Jørgensen and sub supervisor Samuel Emil Schmidt.

Reading instructions

The master thesis contained a problem analysis where the problem domain related to the diagnosis of CAD and the challenges related to the diagnosis was elaborated. Subsequently, the methods used in the master thesis are described. It includes a description of the data, the development of the CAD network, and how the CAD network was validated. At last, the result of the CAD network and discussion are presented. Throughout the master thesis, the Harvard referencing method is used. In the bibliography, the references are present. Moreover, four appendices are attached, consisting of the literature search, layers used in the CAD network, training of the CAD network, and hyper-parameter results for the CAD network.

Julie Rask Madsen

Mia Hebsgaard Ransborg

Contents

1 Introduction **3**

2 Problem analysis **4**
2.1 Coronary artery disease 4
2.2 Diagnosis of coronary artery disease 5
2.3 State of the art regarding detection and classification of CAD 8

3 Method **11**
3.1 General approach of detecting CAD in CCTA volumes 11
3.2 Data and pre-processing 12
3.3 Identification of CAD in CCTA volumes 17
3.4 Training of the CAD network 25
3.5 Validation of the identification of CAD 27

4 Results **31**
4.1 CNN-RNN 31
4.2 Patient data network 37
4.3 CAD network 41

5 Discussion **49**
5.1 Limitations 51

6 Conclusion **53**

Bibliography **54**

A Structures Literature search **60**

B Layers in the CAD network **62**
B.1 Convolution 62
B.2 Batch normalization 63
B.3 Activation functions 63
B.4 Max pooling 67
B.5 Dropout 67
B.6 Fully connected neural network 68

C Training of the CAD network **69**
C.1 Gradient descent algorithm 69

D Hyper-parameter optimization of the CAD network **72**
D.1 Number of recurrent layers 73
D.2 Number of GRUs 77
D.3 Dropout 79

D.4 Activation function 80
D.5 Learning rate 82

1 Introduction

Coronary artery disease (CAD) is the leading cause of death worldwide accounting for 16% of all deaths in 2019 [Zreik et al., 2018b; Fleming, 2022]. CAD is defined as a reduction of the blood supply to the heart due to atherosclerosis in the coronary arteries, thus, leading to angina, myocardial ischemia, and myocardial infarction [Zhao et al., 2021; Moon et al., 2021]. The golden standard for diagnosing CAD is coronary angiography (CAG), which is an invasive method [Cong et al., 2019]. However, in patients with low or intermediate risk of CAD, coronary computed tomography angiography (CCTA) is an accepted method for detecting or excluding CAD Nissen et al. [2016]; Kristanto et al. [2013]. CCTA is a non-invasive method with a high sensitivity and negative predictive value of 70 - 80% but a specificity of 48 - 78% [van Hamersvelt et al., 2019; Ko et al., 2012]. Thereby, 22 - 52% of the conducted CAG examinations are unnecessary [Zreik et al., 2019]. The assessment of CAD in the CCTA volumes is manually inspected by the clinicians, which is time-consuming and prone to inter-and intra-observer variability [Pugliese et al., 2009; Han et al., 2020]. Thus, the interpretation and diagnosis can vary amongst the clinicians [Cheung et al., 2021].

The use of neural networks for the detection of CAD in CCTA volumes has increasingly been investigated [Chen et al., 2020]. In the study by Podgorsak et al. [2020], a 2D convolution neural network (CNN) was trained to map the severity of CAD in CCTA volumes. Their method obtained an accuracy, specificity, sensitivity, F1 score, and an area under the receiver operating characteristic curve (AUROC) of 80.9%, 80%, 83%, 0.804, and 0.862, respectively. Another study by Zreik et al. [2018b] investigated a CNN in series with a recurrent neural network (RNN) for characterization of CAD and determination of the severity in CCTA volumes. The performance of the proposed approach resulted in an accuracy, F1 score, and κ of 0.77, 0.61, and 0.61 for the characterization analysis, and 0.80, 0.75, and 0.68 for the analysis of the severity, respectively. The above mentioned studies showed the potential of using deep learning for detecting CAD in CCTA volumes.

Problem analysis 2

The chapter contains a problem analysis concerning the diagnosis of coronary artery disease along with the state of the art regarding automatic methods for diagnosing CAD, which leads to the aim of the present study.

The most common cardiovascular disease worldwide is coronary artery disease (CAD), which accounted for 8.9 million deaths in 2019 corresponding to 16% of all deaths making CAD the leading cause of death worldwide [Zreik et al., 2018b; Fleming, 2022]. CAD refers to partially or completely blocked areas in the coronary arteries, which supply the muscles of the heart, called the myocardium, with oxygen-enriched blood and nutrients [Podgorsak et al., 2020; Huang and Yin, 2021]. The myocardium needs a constant supply of oxygen and nutrients, thus, a significant hemodynamically reduction of blood flow will cause starvation of the myocardium [Zhao et al., 2021]. CAD leads to angina, myocardial ischemia, and myocardial infarction [Moon et al., 2021; Zhao et al., 2021]

2.1 Coronary artery disease

The narrowing or blockage of the coronary arteries is called stenosis which is caused by atherosclerosis [Mirunalini et al., 2019]. Atherosclerosis is the thickening or hardening of the arteries due to plaque build-up in the inner arterial wall [Moon et al., 2021]. The plaque build-up is incrementally illustrated in figure 2.1. The formation of plaque is initiated when macrophages ingest and oxidize the accumulated lipoproteins. When the macrophages are filled up, they become foam cells. The foam cells deposit to the endothelial walls of the coronary arteries releasing cytokines that will cause inflammation. The inflammation causes the smooth muscle cells to divide resulting in a thickening of the vessel wall. Further, apoptosis of the smooth muscle cells is associated with the formation of calcium, which coalesces into a larger plague over time [Mori et al., 2018]. Additionally, lipid deposition on the arterial wall causes the plaque to grow. [Lawrence, 2003; Hall and Hall, 2011]

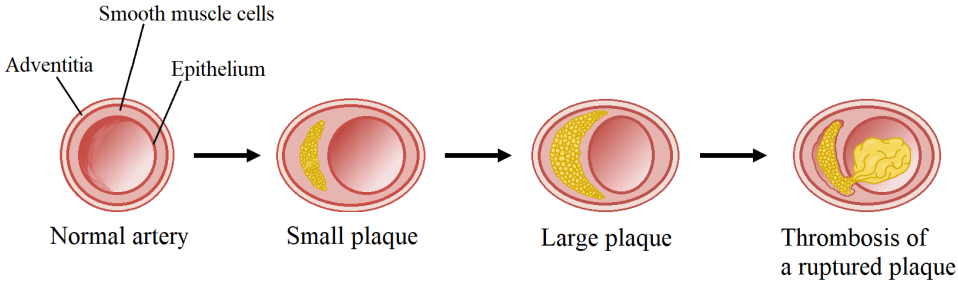


Figure 2.1. Plaque build up in the inner wall of a coronary artery, resulting in thrombosis. Inspired by [Hall and Hall, 2011]

The components of the plaque consist of cholesterol, fat, calcium, and other substances that characterize the plaque as either calcified-, non-calcified- or mixed plaque relative to the composition of substances [Mirunalini et al., 2019]. The calcified plaque is considered a stable type of plaque, whereas non-calcified plaque and mixed plaque are thin capped with fibrous, thus, considered unstable since these types of plaque are more likely to rupture [Kristanto et al., 2013; Mori et al., 2018]. The rupture of a plaque is shown in figure 2.1. The rupturing of the plaque can lead to thrombosis resulting in acute coronary syndrome and may cause irreversible damage to the myocardium due to myocardium infraction. [Zreik et al., 2018b; Cheung et al., 2021].

The plaque build-up causing CAD is associated with multiple risk factors which include high blood pressure, high cholesterol levels, smoking, diabetes, overweight, lack of physical activity, unhealthy diet, and stress. Furthermore, the risk of developing CAD increases with age, men have a greater risk of developing CAD than women, and a family history of early heart disease is also a risk factor. [Hajar, 2017] The lifestyle-related risk factors, e.g smoking, overweight, and unhealthy diet can be reduced with preventative treatment [Hajar, 2017] The first symptom of CAD most often shows as acute myocardial infarction. Other symptoms of CAD include pain or discomfort in the center of the chest and/or in the arms, the left shoulder, elbows, jaw, or back. Further, shortness of breath or difficulty in breathing along with nausea or vomiting, light-headedness or faintness, cold sweats, and turning pale might show as symptoms of CAD. [World Health Organization, 2021] The shortness of breath most likely increases as the CAD advances [Zhao et al., 2021]. In symptomatic patients, the diagnosing of CAD is critical to determine a suitable clinical management [Chen et al., 2020]. Currently, CAD can be treated by using percutaneous coronary intervention or coronary artery bypass grafting together with aggressive medical therapies[Zhao et al., 2021]. However, an early diagnosis of CAD may allow early medicinal intervention and prevent CAD from causing severe or permanent damage to the myocardium [Moon et al., 2021; Cheung et al., 2021]

2.2 Diagnosis of coronary artery disease

When CAD is suspected, the diagnosis of CAD is increasingly based on imaging technologies that provide visualization of the coronary arteries [Hampe et al., 2019]. The golden standard for diagnosing CAD is coronary angiography (CAG) [Cong et al., 2019]. The method of CAG is illustrated in figure 2.2. It is used to evaluate the severity and extent of CAD. When performing CAG, a catheter is inserted into a vessel in the arm or the groin. The catheter is led to the coronary arteries through the vessels and then injects a contrast agent directly into the coronary arteries which are captured by x-ray motion images. [Moon et al., 2021] The extent of CAD is examined by a cardiologist who selects the keyframes from the CAG showing the most visible coronary arteries when the contrast agent is present. In these frames, CAD is then manually assessed by a visual inspection, where anatomically significant CAD is defined as a narrowing of the coronary artery lumen of 50% or more. To assess the severity of CAD, the fractional flow reserve (FFR) is measured during CAG [Moon et al., 2021]. FFR refers to the ratio of flow distal of stenosis to the flow proximal to the stenosis. The scale of the FFR range from 0 referring to occlusion of the artery to 1 referring to the absence of stenosis. Stenosis is defined as

hemodynamic significant when the FFR is 0.80 or lower [Podgorsak et al., 2020].

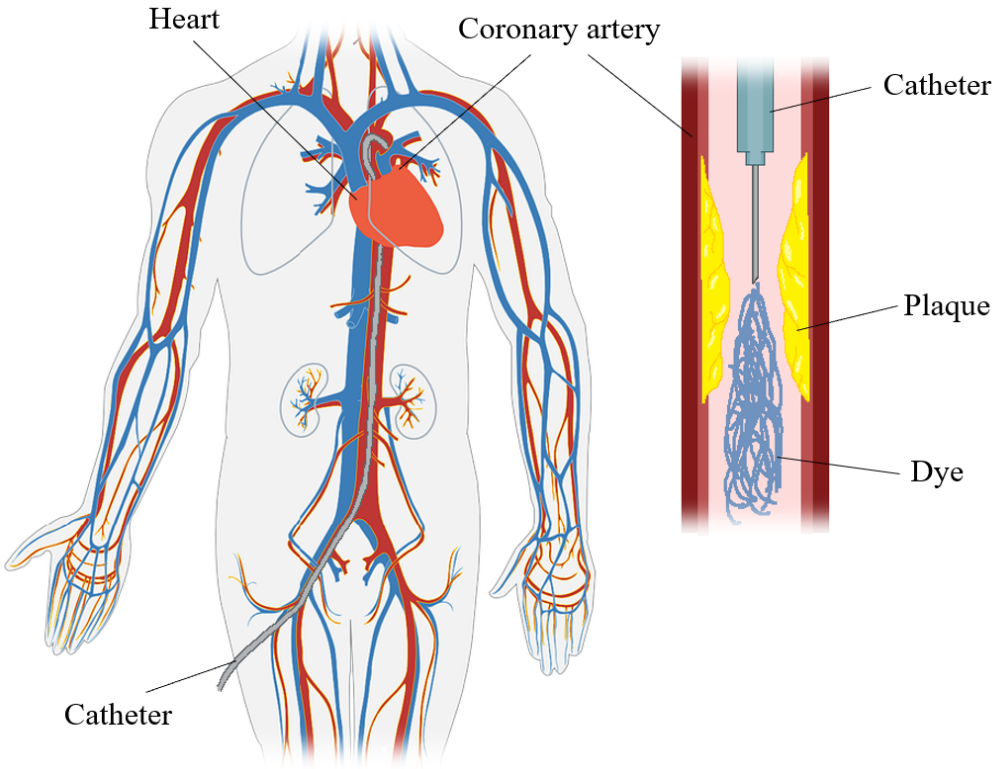


Figure 2.2. Coronary angiography. A catheter is inserted into a vessel in the groin (or in the arm) and led to the coronary arteries. Then a contrast agent is injected into the coronary arteries to highlight the coronary arteries in an X-ray image. The flow before and after stenosis is measured with the catheter. Inspired by Moon et al. [2021]. [Sharma, 2022; Podgorsak et al., 2020; Moon et al., 2021]

Although CAG along with the FFR measurement is the golden standard, the method is invasive and costly. The risk for complications during the CAG, i.e arrhythmia, stroke, and myocardial infarction due to the catheter passing plaque causing it to rupture, and death is present. [Chen et al., 2020; Nissen et al., 2016]

Although CAG is the golden standard, coronary computed tomography angiography (CCTA) is a frequently used procedure for patients with a low or intermediate risk of CAD, since CCTA is an accepted diagnostic tool for detecting CAD and excluding hemodynamically significant CAD. [Nissen et al., 2016; Kristanto et al., 2013] The method of CCTA is shown in figure 2.3. CCTA is non-invasive and utilizes the measurement of X-ray transmission profiles through a patient from numerous angles which are reconstructed into an image of the grey level scale. Additionally, an intravenous injection of a contrast agent is given to the patient to visualize the coronary artery lumen. [Achenbach, 2006] Thus, CCTA can image the coronary artery lumen in cross-sectional images of the heart [Zhang et al., 2008].

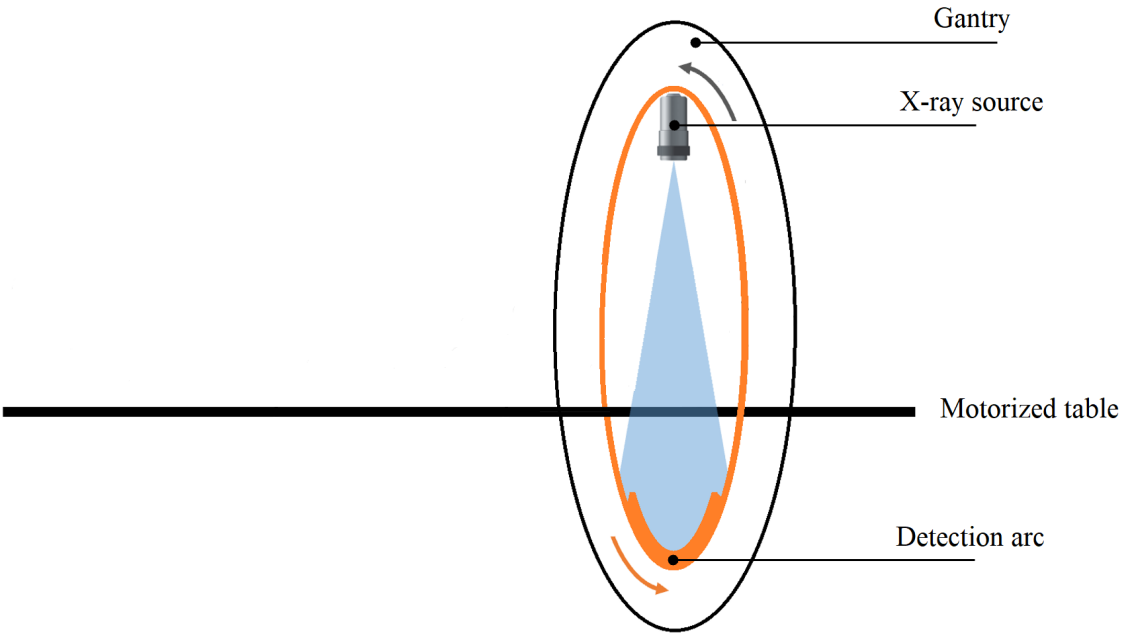


Figure 2.3. Coronary CT angiography. A patient is injected with a contrast agent and placed on a motorized table moving the patient through the gantry. In the gantry, the X-ray source is projecting a X-ray beam detected by the detection arc while rotating around the patient. Inspired by Dance et al. [2014]. [Dance et al., 2014; Sun, 2013]

CCTA has high sensitivity and negative predictive value of 70 - 80% for detection of CAD but the specificity of detecting hemodynamically significant CAD is lower with a range of 48 - 78% [van Hamersvelt et al., 2019; Ko et al., 2012]. Due to the low specificity of CCTA, 22 - 52% of the patients get an unnecessary invasive FFR measurement. [Zreik et al., 2019]. Additionally, the calcified plague can create blooming artifacts in CCTA images. Thus, it is difficult to accurately assess the degree of luminal narrowing using CCTA when heavily calcified plaques are present. Approximately 50 - 70% of all coronary artery plaques are calcified in patients with asymptomatic or suspected CAD. [Zhang et al., 2008; Han et al., 2020] Thereby, calcified plaque can result in overestimation of the degree of stenosis [Wang et al., 2011].

Furthermore, the CCTA volumes are visually inspected by specialized clinicians, since accurate geometric information is needed to assess the severity of stenosis. Thus, the method is subjective and time-consuming which can result in variability in the assessment of the stenosis severity. [Cheung et al., 2021; Zhao et al., 2021; Han et al., 2020; Mirunalini et al., 2019] The inter-and intra-observer agreement of the interpretation of CCTA volumes for CAD has been investigated by Pugliese et al. [2009] and Saur et al. [2010], where Cohen's kappa coefficient, κ, was found. Ideally, κ should be 1 indicating total agreement but the highest κ found in the studies were κ=0.48 and κ=0.66 whereas the inter-and intra-observer agreement for CAG was κ=0.95. [Pugliese et al., 2009; Saur et al., 2010] Furthermore, interpretation and diagnosis are dependent on the individual clinician's experience and expertise. Thereby, the diagnosis can vary between clinicians. [Cheung et al., 2021]

To reduce the number of unnecessary invasive procedures, it is desired to increase the specificity of CCTA detecting hemodynamically significant CAD [Podgorsak et al., 2020].

An increased specificity could be obtained using automatic methods for detecting CAD in CCTA volumes [Han et al., 2020]. Additionally, automatic interpretation of the CCTA volumes may reduce the inter- and intra-observer variability of the interpretation of CCTA volumes [Moon et al., 2021].

2.3 State of the art regarding detection and classification of CAD

The development of automatic interpretation methods of CCTA volumes has increasingly been relying on deep learning methods. Deep learning has been investigated for detection of plaque, stenosis, and hemodynamically significant CAD. [Chen et al., 2020] The use of deep learning and especially neural networks has been explored in several studies, aiming to improve the diagnosis of CAD based on CCTA volumes. A few of the studies investigating the use of neural networks for diagnosing hemodynamically significant CAD or significant narrowing of the coronary arteries are presented in the following. [Podgorsak et al., 2020; Kumamaru et al., 2020; Zreik et al., 2018b,a]

In the study by Podgorsak et al. [2020], they evaluated if a 2D convolutional neural network (CNN) trained on CCTA slices and corresponding FFR values from 64 patients could be used to non-invasively assess the CAD severity. The CCTA slices and FFR values were obtained from the following coronary arteries; left anterior descending, left circumflex, and right coronary artery. Straightened curved planar reformation (SCPR) were obtained from CCTA volumes of the coronary artery branches using eight different rotational states every 45-degrees around the vessel centerline of the artery branch. SCPR is a resampling and visualization technique that ensures displaying of a whole artery within a single image [Kanitsar et al., 2002]. This resulted in 512 slices of the coronary arteries. To asses the CAD severity, a CNN was developed. This was done by using binary classification to classify if the CCTA slices was hemodynamically significant or non-significant based on a FFR threshold of 0.80. Moreover, class activation maps (CAM) was used to determine which features the network was weighting as important for the classification task. This was done to ensure that the network found regions in the slice that were believed to be critical for CAD such as the calcium burden and coronary artery geometry. The information from the class activation maps was used to optimize the image data prior to subsequent network training, such as only calcium burden and the coronary artery geometry were present in the slices. The mean classification accuracy for the CNN in predicting hemodynamically significance of stenosis was 80.9% and an area under the receiver operating characteristic curve (AUROC) of 0.862. The specificity, sensitivity, precision and F1 score was 80%, 83.0%, 77.8% and 0.804, respectively. This demonstrates that the CNN could be used to non-invasively predict the hemodynamically significance of CAD using SCPRs of coronary artery branches.

Another study by Zreik et al. [2018a] proposed a method for identification of patients with at least one hemodynamically significant stenosis in a single CCTA slice acquired at rest. The method was based on analysis of the left ventricular (LV) myocardium by a multi-scale CNN for segmentation of the LV myocardium followed by a convolutional autoencoder to extract features of the LV myocardium. The extracted features were then

used for classification of patients with or without hemodynamically significant stenosis with a support vector machine classifier. The data included in the study consisted of CCTA volumes from 166 patients where 156 patients underwent FFR measurements within one year after the acquisition of the CCTA volumes. The minimum FFR value for each patient was used as reference standard of whether or not hemodynamically significant stenosis was present. Further, in 40 randomly selected CCTA volumes, the LV myocardium was manually segmented by a trained observer to create reference standards for the training, validation and test of the segmentation CNN. The evaluation of the segmentation, performed on 20 test volumes, resulted in a Dice coefficient of $91.4 \pm 2.1\%$ and a MAD of 0.7 ± 0.1 mm. From the manual evaluation each segmentation was graded as very accurate (74.0%), accurate (15.8%), mostly accurate (3.4%), inaccurate (6.1%) and segmentation failed (0.7%). The test of the classification resulted in average accuracy, sensitivity and specificity of 0.71, 0.70 and 0.71 and AUROC of 0.74 ± 0.02. However, the method of analysing LV myocardium only showed moderate performance compared to methods analysing the blood flow in the coronary arteries as studies by Min et al. [2012] and Nørgaard et al. [2014] achieved an accuracy of 0.71 and 0.81 and AUROC of 0.81 and 0.90, respectively.

The study by Zreik et al. [2018b] proposed a multi-task recurrent CNN (RCNN) for both detection and characterization of types of plaque and detection along with determination of significant stenosis in the coronary arteries in multi-planar reformated (MPR) images from CCTA volumes. The study included CCTA volumes from 163 patients containing 1259 manually labeled arterial segments in 534 arteries. The RCNN consisted of a 3D CNN and a recurrent neural network (RNN) connected in series. The type of plaque was categorized as either no plaque, non-calcified plaque, mixed or calcified plaque whereas the classification of stenosis ranged from no stenosis, non-significant stenosis or significant stenosis. As reference standard, the MRP images were manually annotated by an expert. Each plaque was marked with a start- and end-point along with the type. Further, the significance of the stenosis caused by the plaque was marked as either non-significant if luminal narrowing $< 50\%$ and significant if luminal narrowing $\geq 50\%$. To evaluate the performance of the proposed network, the F1 score was used for assessment of both the plaque classification and stenosis classification. Moreover, an unweighted Cohen's κ metric and Cohen's linearly weighted κ metric were used to evaluate the reliability between the predicted labels and the reference standard of the type of plaque and the significance of the stenosis, respectively. Performance of the RCNN resulted in accuracy, F1 score, and κ of 0.77, 0.61, and 0.61 for the plaque analysis, and 0.80, 0.75, and 0.68 for the stenosis analysis, respectively.

From the above-described studies, the potential of deep learning for detecting hemodynamically significant CAD or lumen narrowing of the coronary arteries is promising. The main disadvantage of the studies could be the amount of data used since it is well known that neural networks require a large data amount to be able to generalize to the unseen data [Shin et al., 2016]. However, the study by [Podgorsak et al., 2020] obtained a specificity of 80.0%. An advantage of the study by Podgorsak et al. [2020] is the use of SCPR CCTA images, which contained the whole coronary arteries. Thus, changes in the coronary artery were visible in each image. Moreover, the class activation maps showed activations over the different plaque, but low activations in the lung volumes and surrounding cardiac tis-

sue. Further, the studies by Podgorsak et al. [2020] and Zreik et al. [2018a] used the FFR values as the reference standard, which expresses the hemodynamic significance of CAD. However, the method of segmenting LVM had only moderate performance compared to other studies. The study by Zreik et al. [2018b] also used reconstructed CCTA slices, which contained the whole coronary artery, which was first analyzed with the 3D CNN for feature extraction, and then these features were collectively analyzed in the RNN. Thereby, the temporal dependencies in the CCTA volumes were captured. If deep learning, in particular neural networks, can be exploited when diagnosing CAD in CCTA volumes, the efficiency, quality, and performance of the diagnosis could be improved [Han et al., 2020]. Thus, the aim of the present study is:

2.3.1 Aim

To identify the presence of CAD in CCTA volumes of the heart and the coronary arteries using deep learning

Method 3

3.1 General approach of detecting CAD in CCTA volumes

The chapter contains a description of the method used in the present study. The method includes a general system description, data pre-processing, the CAD network, the training of the CAD network, and validation of the CAD network. For the data pre-processing Matlab version R2020b was used. The architecture of the CAD network was written in Python and for the training and testing of the CAD network was Aalborg University's GPUs (Tesla V100-SXM3-32GB) used.

The present study sought to identify patients with CAD based on CCTA volumes and corresponding patient data using deep learning. The proposed approach is illustrated in figure 3.1. The approach was based on a CNN and an RNN connected in series to perform sequential frame analysis of the CCTA volumes and a fully connected neural network (FCNN) to analyze patient data associated with the development of CAD. Both the CNN-RNN and the FCNN were used to classify if the CCTA volumes contained CAD or not.

The input to the CNN-RNN consisted of CCTA volumes. Each CCTA volume represented cross-sections of the coronary arteries continuously from the top of the heart to the apex. Thus, if lumen narrowing of the coronary arteries was present, it would show as e.g. the diameter of the lumen would decrease over the following slices depending on the previous slices. Therefore, the slices in the CCTA volumes were independently analysed for the shape features of the coronary arteries through the CNN. Then, the feature maps from the CNN were collectively analyzed through the RNN, since the identification of CAD in the CCTA volumes depended on the features of the coronary arteries in each slice within the CCTA volumes.

Simultaneously with the CNN-RNN, the FCNN, called the patient data network, processed the patient data. The input to the patient data network consisted of patient data associated with the development of CAD, which outputted extracted features from the patient data. The output from the CNN-RNN and the patient data network were concatenated and fed as input to a classifier which outputted a prediction of whether or not the CCTA volumes contained CAD. Furthermore, the predictions were validated to ensure the predictions were correct. The combined CNN-RNN and patient data network will be referred to as the CAD network in the following.

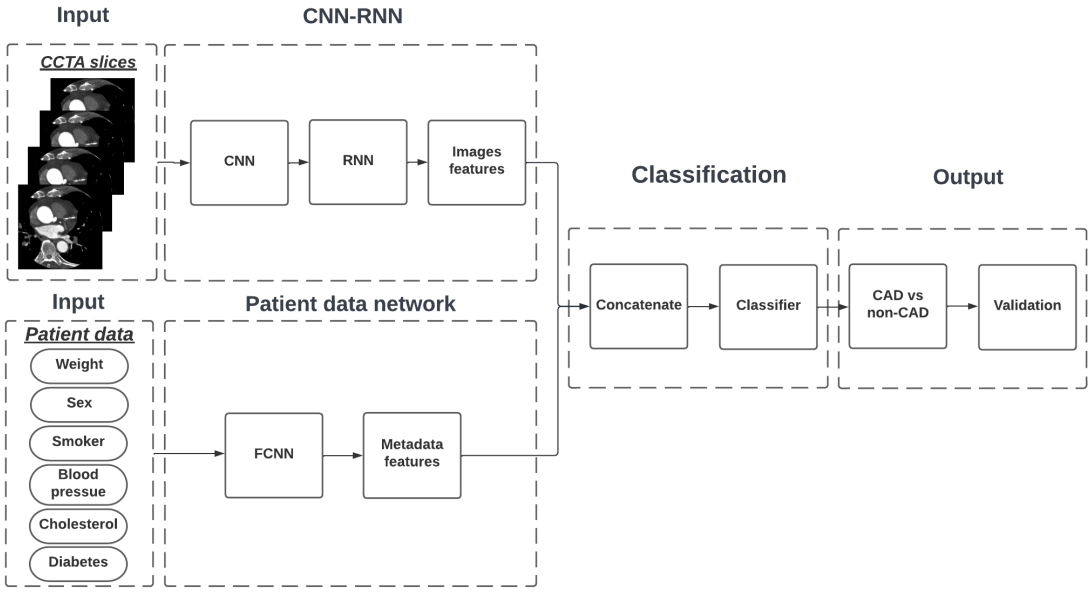
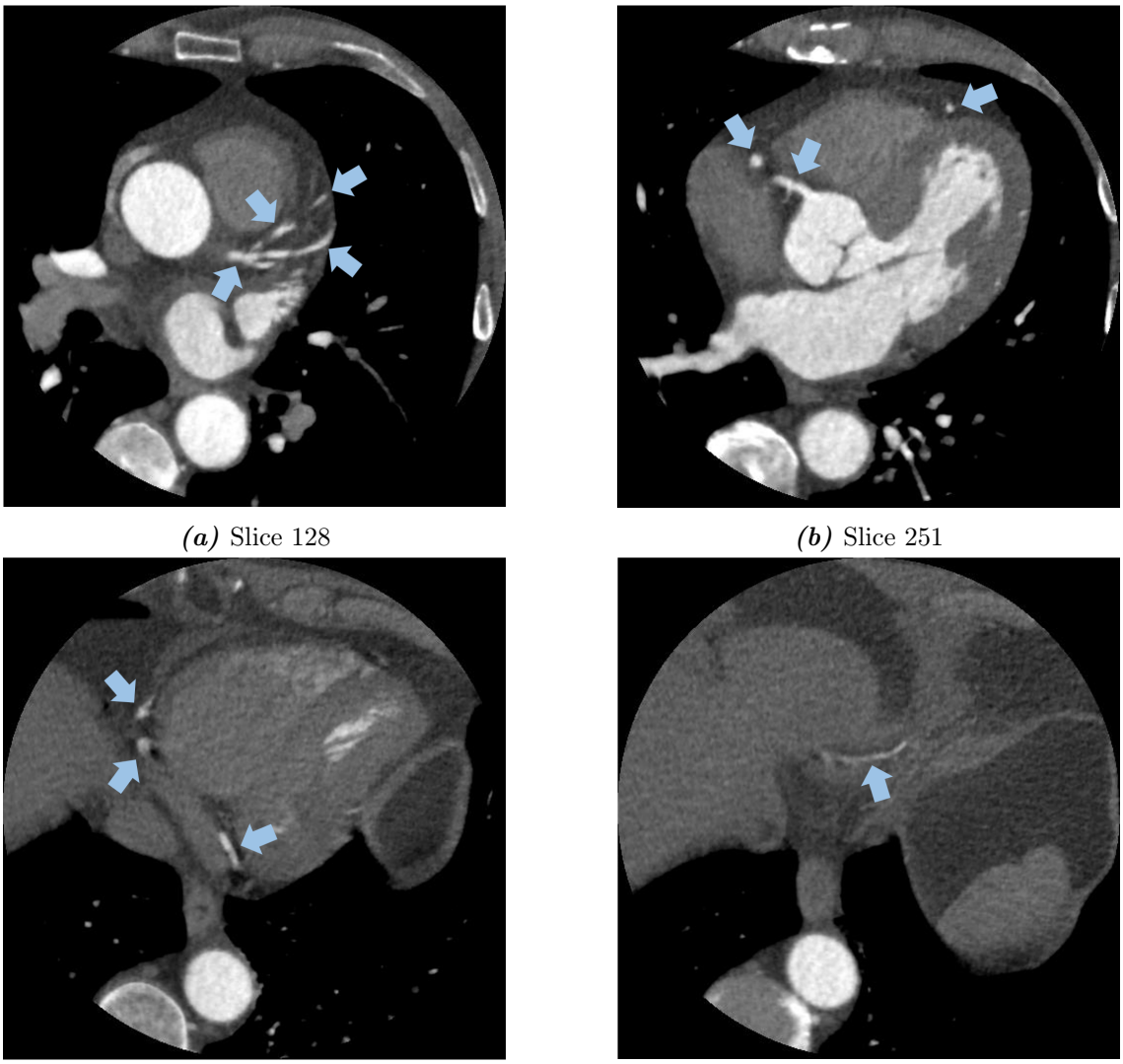


Figure 3.1. The overall approach for finding CCTA volumes with CAD. The approach consists of a CNN-RNN and a patient data FCNN called the CAD network. The CCTA volumes are inputs to a CNN which outputs image feature maps for each slice in the CCTA volume. The feature maps from the CNN are then fed to an RNN that outputs image features for the entire CCTA volume. Simultaneously, the patient data is input to an FCNN that outputs patient data features. The output from the CNN-RNN and the patient data FCNN are then concatenated and fed to a classifier that outputs a prediction of whether or not the CCTA volume contains CAD or not. Finally, the predictions are validated.

3.2 Data and pre-processing

The data for this study consisted of CCTA volumes of the heart from 378 patients and an Excel sheet with the corresponding patient data from Dan-NICAD, which was an investigator-initiated, multi-centered, randomized, cohort trial by Nissen et al. [2016]. From the Dan-NICAD, the study subjects were recruited at two regional hospitals (Department of Cardiology, Regional Hospitals of Herning and Silkeborg, Denmark). The CCTA scans were performed on a 320 multi-slice volume CT scanner (Aquillion One, Toshiba Medical Systems, Japan) using prospective electrocardiogram (ECG) gating. The z-axis coverage of the heart varied from 120 to 160 mm, depending on the length of the heart. The acquisition of the CCTA volumes was performed using a gantry rotation time of 0.35 s, slice thickness of 0.5 mm, and prospective ECG triggering. 50 – 80 mL nonionic contrast agent (Optiray 350 mg/mL, Mallinckrodt, Ireland) was administrated according to the weight of the patients. The data was reconstructed at 75% of the RR interval. [Nissen et al., 2016]

Each CCTA volume consisted of a sequence of slices of a spatial resolution of 512x512 pixels varying between 128 and 640 slices illustrating the same stage in the heart cycle. An example of slices from one CCTA volume can be seen in figure 3.2.



(a) Slice 128 ***(b)*** Slice 251

(c) Slice 413 ***(d)*** Slice 467

Figure 3.2. Examples of slices from one CCTA volume of the heart. The entire CCTA volume contains 560 slices. The slices starts (a) at the top of the heart showing the pulmonary artery and the aorta which the coronary arteries ascents from and ends (d) at the apex. The coronary arteries are marked with blue arrows.

The data set contains 156 patients with CAD and 222 patients without CAD. Thus, the CCTA volumes was labeled CAD or non-CAD depending on whether or not the patients was diagnosed with CAD. A CCTA volume was labeled with CAD if the corresponding patient was diagnosed with hemodynamically obstructive CAD according to the Excel sheet of patient data. Otherwise, a CCTA volume was labeled as non-CAD if the corresponding patient was diagnosed without or with insignificant hemodynamically obstructive CAD.

3.2.1 Extraction of risk factors for CAD from the patient data

From the Excel sheet of patient data, the risk factors associated with development of CAD were found for each patient included in the study. The extracted risk factors for CAD are listed in table 3.1, corresponding to the risk factors described in section 2.1 as these were available for the study. The risk factors included the male sex, overweight, diabetes,

smoker, high blood pressure, and high cholesterol level. The definition of overweight was based on the Body Mass Index (BMI), calculated for each patient using weight and height [WHO, 2000]. The definition of risk values of high blood pressure and high blood cholesterol in table 3.1 were chosen based on values defined by Messerli et al. [2007]; Ma and Shieh [2006].

Risk factor	**Gender**	**Smoker**	**Diabetic**	**BMI**	**Blood pressure**	**Cholesterol**
Risk value	Male	Active/ former	Yes	$\geq 25\ kg/m^2$	$\geq$ 140/90 mmHg	$\geq$ 5 mmol/L

Table 3.1. The risk factors for development of CAD. Each risk factor has an assigned 'Risk value' indicating the value of which the risk factors have an influence of the development of CAD.

The risk factors for each patient were evaluated from the patient data sheet, where a label of 1 was assigned to a risk factor obtaining a value greater than or equal to the corresponding risk value, e.g. a BMI of 25 kg/m^2 or more was labeled 1. Otherwise, a label of 0 was assigned. The use of risk factors excluded 17 patients as 14 patients were missing cholesterol values, two patients were missing values for weight and one patient was missing values for both weight and height. The weight and height were necessary to calculate the BMI for each patient. Thereby, 8 patients with CAD and 9 patients without CAD were excluded. This resulted in 361 patients in total, where 148 patients had CAD and 213 without CAD. The number of the included patients with and without CAD are shown in baseline table 3.2.

	No. of subjects	**Male/ Female**	**Age (years)**	**Weight (kg)**	**Height (cm)**	**BMI (kg/m^2)**
CAD	148	103/45	60.1 ± 9.3	82.3 ± 14.5	173.8 ± 8.4	27.2 ± 4.2
Non-CAD	213	110/103	60.8 ± 7.4	80.2 ± 14.9	172.1 ± 9.3	27 ± 4.1
	Systolic blood pressure (mmHg)	**Diastolic blood pressure (mmHg)**	**Cholesterol (mmol/L)**	**Smoker/ Non smoker**	**Diabetic/ Non diabetic**	**Risk factors**
CAD	145.3 ±19.7	85 ±11.6	5.5 ± 1.5	84/64	19/129	3.1 ± 1.1
Non-CAD	141.2 ± 18.3	83.3 ± 10.4	5.6 ±1.2	114/99	12/101	2.7 ± 1

Table 3.2. Baseline of the study subjects. The mean of the age, weight, height, Body Mass Index (BMI), systolic and diastolic blood pressure, cholesterol level, and the number of risk factors were found along with the standard deviation.

3.2.2 Dividing the CCTA volumes into a training-, validation-, and test set

Afterward, the CCTA volumes were divided into approximately 80% of training data, 10% of validation data, and 10% of test data using stratified randomization depending on the distributions of CAD versus non-CAD and the number of risk factors. The distribution of the CAD and non-CAD along with the number of risk factors can be seen in table 3.3. This resulted in a training data set with 286 patients, a validation data set with 37 patients, and a test data set with 38 patients. The division of the three data set can be seen in figure 3.3, 3.4, and 3.5. If the data sets were not independent of each other, it could cause bias in the model. [Chollet et al., 2018] Independence among the data sets was ensured by not having the same patient in more than one data set. Furthermore, the data sets must

represent the population of the available data [Chollet et al., 2018]. This was achieved by placing the same distribution of patients with CAD and without CAD in each data set. Moreover, the distributions of patients, both with CAD and without CAD, were divided by the number of risk factors for CAD to ensure the distribution of the number of risk factors were kept in the data sets providing best possible representation of the population. The patients could have between 0-6 risk factors for CAD, although none of the patients had all 6 risk factors.

Patients = 361 (100%)						
	CAD = 148 (41%)					
No. of risk factors	0	1	2	3	4	5
No. of patients	0 (0%)	9 (6.1%)	33 (22.3%)	53 (35.8%)	39 (26.3%)	14 (9.5%)
	Non-CAD = 213 (59%)					
No. of risk factors	0	1	2	3	4	5
No. of patients	2 (1%)	26 (12.2%)	53 (24.9%)	85 (39.9%)	39 (18.3%)	8 (3.7%)

Table 3.3. The distribution of CAD and non-CAD along with the number of risk factors in the data set for the present study.

The distributions of patients with and without CAD along with their number of risk factors were followed in the training data set, the validation data set, and the test data set. The distribution of the patients in the three data set is illustrated in figure 3.3, 3.4, and 3.5.

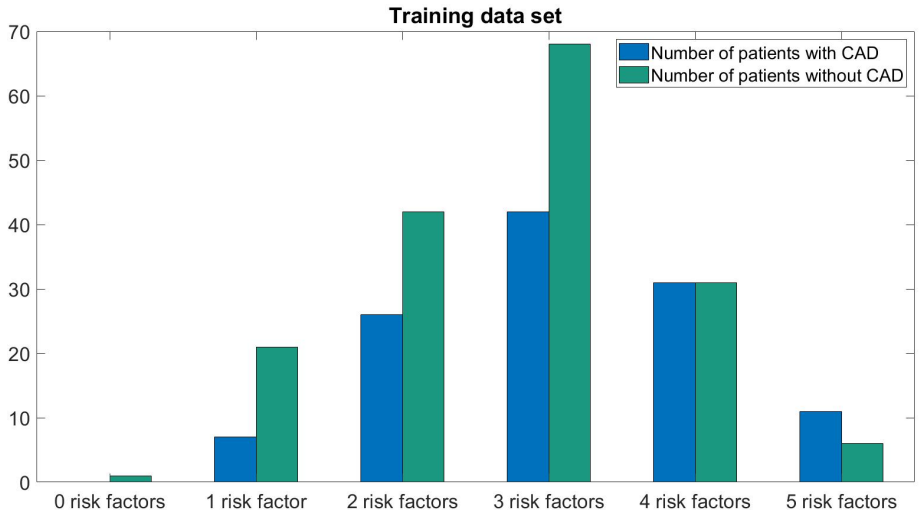


Figure 3.3. The distribution of patients with CAD and without CAD in relation to risk factors in the training data set.

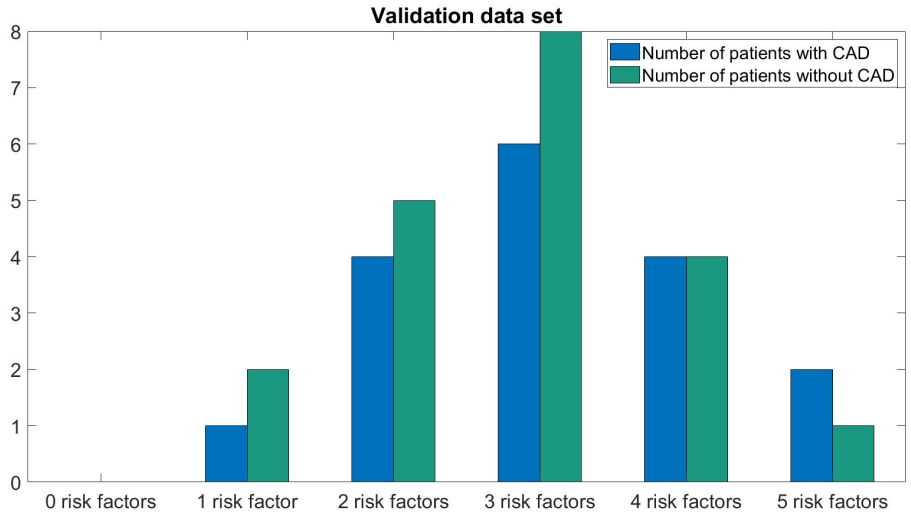


Figure 3.4. The distribution of patients with CAD and without CAD in relation to risk factors in the validation data set.

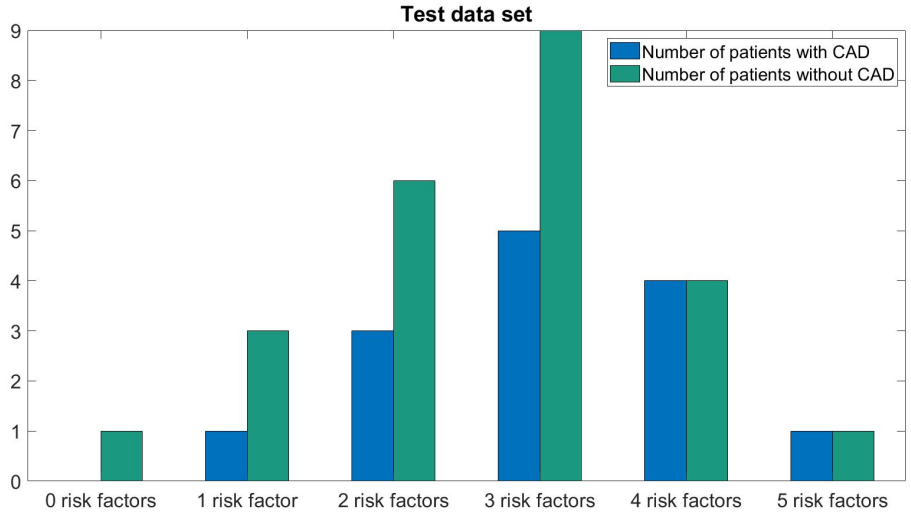
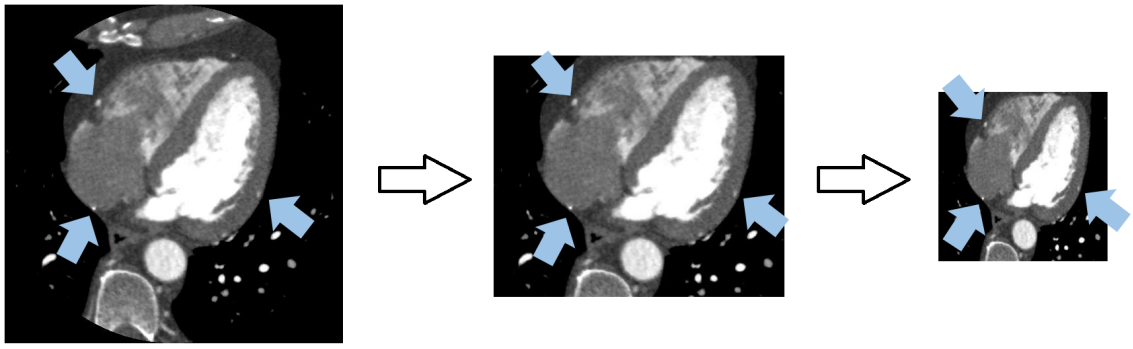


Figure 3.5. The distribution of patients with CAD and without CAD in relation to risk factors in the test data set.

3.2.3 Pre-processing of the slices in the CCTA volumes

The pre-processing of the slices in the CCTA volumes consisted of cropping followed by resizing and finally stacking. In figure 3.6, the pre-processing is illustrated. The CCTA slices were cropped to find the region of interest, which in this case was the heart. The resizing and stacking of the slices were done to decrease the computational burden without compromising the information in the slices of the coronary arteries since their representation in the slices were relatively small as seen in figure 3.6*(c)*. The cropping was performed using MATLAB by indiscriminately choosing a representative CCTA volume, where the region of interest was marked in one of the slices from the CCTA volume. The chosen slice contained the heart at the largest diameter to ensure the information of the coronary arteries remained in the cropped CCTA slices. This resulted in a region of interest

of 440x365 pixels at the same location in every slice in every CCTA volume, as shown in figure 3.6*(b)*.



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

Figure 3.6. Pre-processing of the CCTA volumes. (a) A slice representing the largest diameter of the heart in the CCTA volume (slice 363 of 512 slices) is used to find the region of interest (b) to which the slices are cropped. (c) The slices are then resized. The blue arrows mark the coronary arteries.

Moreover, each of the cropped slices in every CCTA volume was spatially resized from 440x365 pixels to 256x256 pixels, using the Python imaging library Pillow when loading the CCTA volumes in the CAD network. [Pillow, 2022]. Further, the slices in each CCTA volume were stacked as an array where one array represented one CCTA volume from one patient.

3.3 Identification of CAD in CCTA volumes

In the following, the architectures of the sub parts in the CAD network; the CNN, the RNN, and the patient data network are described along with argumentation for the chosen architectures. Finally, the classification of the concatenation between the CNN-RNN and the patient data network in the CAD network is accounted for.

The combination of a CNN and RNN was inspired by the study of [Zreik et al., 2018b], described in section 2.3. In the study by Zreik et al. [2018b], the analysis of CCTA volumes for anatomically significant stenosis using a CNN-RNN showed promising results. Since the purpose of the present study was to identify CCTA volumes containing CAD, and thereby the narrowing of the coronary arteries in the CCTA volumes, the architecture described by Zreik et al. [2018b] was used as a starting point.

3.3.1 Convolutional neural network for feature extraction

The CNN was a sub part of the CAD network which had the purpose of extracting patterns or so-called image features from each slice in the CCTA volumes, independently. The starting CNN architecture by Zreik et al. [2018b] consisted of three blocks, where each block contained a 3D convolutional layer, a batch normalization layer, a Rectified Linear Unit (ReLU) activation function layer, and a max pooling layer. The CNN architecture used in this study differed from the 3D CNN used by Zreik et al. [2018b] as a 2D CNN was used and the final layer in block 3 was a global max pooling layer. An overview of

the CNN architecture of the present study is presented in table 3.4. The 2D convolutional layers were used because of the computational limits of the GPU, thus, preventing the use of 3D convolutional layers. Moreover, a global max pooling layer in block 3 was used to enable the output of the CNN to be given as input to the RNN. The number of layers and filters used by Zreik et al. [2018b] were preserved to avoid an increased computational burden on the GPU.

CNN		
Layers	**Operations**	**Output shape**
Input layer		256x256x1
Block 1, layer 1	Convolution, kernel 3x3, 32 filters	256x256x32
Block 1, layer 2	Batch normalization	256x256x32
Block 1, layer 3	ReLU activation function	256x256x32
Block 1, layer 4	Max pooling, kernel 2x2, stride 2	128x128x32
Block 2, layer 1	Convolution, kernel 3x3, 64 filters	128x128x64
Block 2, layer 2	Batch normalization	128x128x64
Block 2, layer 3	ReLU activation function	128x128x64
Block 2, layer 4	Max pooling, kernel 2x2, stride 2	64x64x64
Block 3, layer 1	Convolution, kernel 3x3, 128 filters	64x64x128
Block 3, layer 2	Batch normalization	64x64x128
Block 3, layer 3	ReLU activation function	64x64x128
Output layer	Global max pooling	1x128

Table 3.4. An overview of the CNN's architecture.

The first layer in the CNN was the convolutional layer, which had the purpose of extracting features such as edges and shapes from the input CCTA slices. [Chollet et al., 2018] The inputs were three-dimensional CCTA slices with a height, width, and depth of 256x256x1 pixels. The depth corresponded to the color channel of the CCTA slices, where a depth of one corresponded to the grey level scale of color information in the CCTA slices. The convolutional layers learned features hierarchically, thus, the first convolutional layer learned local patterns such as shapes, the next convolutional layer would learn larger patterns composed of features extracted in the first convolutional layer, and so on. The convolutional layer extracted features by applying an operation called convolution. The convolution was performed by sliding a match filter over the input CCTA slice making an element-wise multiplication. This resulted in an array that illustrated how well the filter coefficients and pixel intensities in the input CCTA slice match. [Chollet et al., 2018] Elaborated information concerning convolutions can be seen in appendix B.1. The filter applied in the convolution was of the size 3x3 and it moved one pixel at a time, thus, it had a stride of one [Zreik et al., 2018b]. The result of the convolution was a two-dimensional array with output values, called a feature map. Thus, the input CCTA slice was converted to a feature map. The feature map had a height, width, and depth, however, now the depth represented the number of match filters computed over the input

CCTA slice. [Chollet et al., 2018] The filters varied from 32 to 128 in the three blocks of the CNN [Zreik et al., 2018b]. When the input feature map was computed in the convolutional layers, the output feature map was to tiles smaller than the input feature map e.g. the spatial resolution of the output feature map after the first convolution was 254x254. To ensure the output feature map maintained the same spatial resolution as the input feature map, zero padding was implemented. [Chollet et al., 2018] Zero-padding is explained in appendix B.1.1. After the convolutional layer, batch normalization was applied to the output feature map from the convolutional layer.

Batch normalization was used as it overcame the problem, internal covariate shift, by normalizing the input feature maps for the subsequent layer, which resulted in a stable distribution of inputs. Batch normalization is further explained in appendix B.2. To achieve a stable distribution of inputs during training, batch normalization was placed before the activation function. [Ioffe and Szegedy, 2015]

The purpose of the activation function was to introduce non-linearity to the CNN. In this study, the ReLU activation function was used. The ReLU activation function performed a threshold operation on each element of the input feature map. The ReLU activation function resulted in all elements of the input feature map with a negative correlation being removed, thus, ensuring features passed on to the subsequent layer had a significantly high output. Thereby, irrelevant input features were removed. [Nwankpa et al., 2018] The theoretical review of the ReLU activation function can be seen in appendix B.3.1.

The final operation in the first two blocks was max pooling. The purpose of max pooling was to downsample the output feature maps from the ReLU activation function where the most present features were highlighted. This was done by outputting the maximum value of the elements of the feature map in a window, which was slid across the output feature map from the ReLU activation function. Thereby, the max pooling operation halved the spatial resolution of the input feature map while the feature channels were preserved in the output. [Chollet et al., 2018] The sliding window had the size of 2x2 and moved two tiles at a time, thus, it had a stride of two, as illustrated in table 3.4 [Zreik et al., 2018b]. For more information about max pooling, see appendix B.4.

In the last block in the CNN, the last layer was a global max pooling layer instead of max pooling. The global max pooling layer was used to downsample the entire output feature map from the last ReLU activation function to a single value. This was done by outputting the maximum value in the feature map. [Chollet et al., 2018] The feature map outputted from the global max pooling layer was given as input to the RNN.

3.3.2 Recurrent neural network for sequential analysis of CCTA volumes

The feature maps extracted by the CNN were given as input to the RNN, similar to the approach by Zreik et al. [2018b]. The purpose of the RNN was to extract temporal dependencies from the slices in the CCTA volumes. The RNN in the study by Zreik et al. [2018b] originally consisted of two recurrent layers with 64 gated recurrent units (GRU) and a dropout rate of 0.5, each followed by the ReLU activation function except for the output layer where a multi-class softmax activation function was used. In the present study, the

first layer of the RNN was a time distributed layer, which allowed each recurrent layer to be applied to every slice in the CCTA volumes [TensorFlow, 2022]. The architecture of the RNN used in the present study is presented in table 3.5. The RNN architecture from this study contained three recurrent layers with 128 GRU based on the hyper-parameter optimization results in appendix D.1 and D.2. The same activation function that was used in the CNN was also used in each recurrent layer since the same activation function typically was in all the hidden layers. Thus, the ReLU activation function was used in the RNN.The dropout rate of 0.5 was kept based on the hyper-parameter optimization in appendix D.3.

RNN		
Layers	**Operations**	**Output shape**
Layer 1	Timedistributed	1x1x128
Layer 2	GRU, 128 hidden units, ReLU, recurrent dropout=0.5	1x128
Layer 3	GRU, 128 hidden units, ReLU, recurrent dropout=0.5	1x128
Layer 4	GRU, 128 hidden units, ReLU, recurrent dropout=0.5	1x128

Table 3.5. An overview of the RNN's architecture.

The RNN differed from the CNN as the output of the current input feature map depended on both the current input feature map and the prior input feature maps. Thereby, the RNN was designed to utilize sequential data. [Chollet et al., 2018] An illustration of the conceptual process of the CCTA volume through the RNN is shown in figure 3.7.

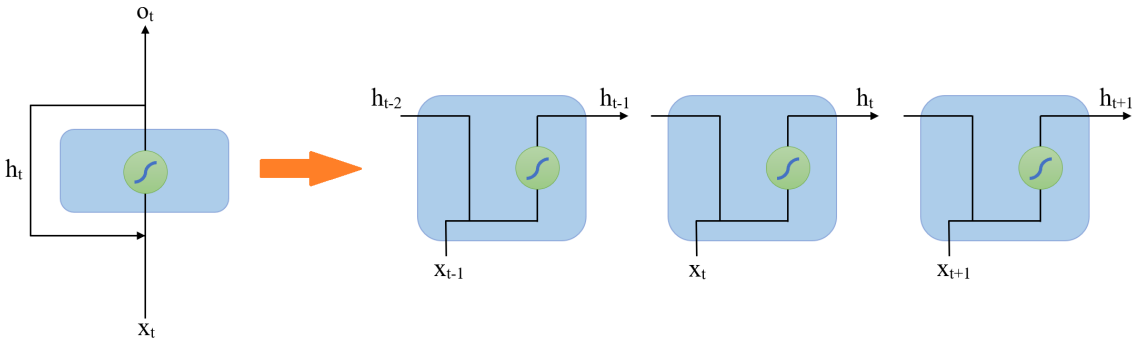


Figure 3.7. A simple RNN architecture. The output o_t from the hidden unit equals the hidden state h_t, which is concatenated with the input x_t to the hidden unit. When the RNN is unfolded, the previous hidden state h_{t-1} and h_{t-2} calculated from the previous input x_{t-1} is concatenated with the current input x_t and passed through an activation function. The output from the activation function becomes the new hidden state h_t, which is passed on to the next hidden unit with a new input x_{t+1}. When the final input in the CCTA volume has been processed, the hidden state becomes the output of the RNN which is representative of the entire CCTA volume. Inspired by [Chollet et al., 2018; Cho et al., 2014]

To incorporate the learned features from prior input feature maps, the RNN used hidden states to obtain a memory of the previous input feature map. Thus, the RNN could be suitable for the identification of CAD in the entire CCTA volumes of cross-sectional slices of the heart and the coronary arteries. [Chollet et al., 2018; Cho et al., 2014] The

hidden state contained information on previous input feature maps which is updated by the following equation

$$h_{(t)} = f(h_{(t-1)}, x_t), \tag{3.1}$$

where the hidden state calculated from the previous CCTA slices h_{t-1} is concatenated with the current input feature map x_t and then passed through an activation function f. The output of the activation function is the updated hidden state h_t. This process continued until the last feature map in the CCTA volume had been processed and, thus, the output is representative of the entire CCTA volume. [Chollet et al., 2018; Cho et al., 2014]

The hidden units used in the recurrent layers were GRU. The overall concept of GRU utilized mechanisms, called gates, to control the flow of information through the RNN. An example of a GRU is illustrated in figure 3.8. The flow of information was controlled by an update- and a reset gate, which controlled what information to keep or forget among each input feature map. These gates contributed to the so-called cell memory as they updated the hidden state that transferred relevant information from unit to unit processing the feature maps in the CCTA volume. Thus, the GRU removed irrelevant information to the prediction of CAD or non-CAD in a CCTA volume. [Chung et al., 2014; Cho et al., 2014]

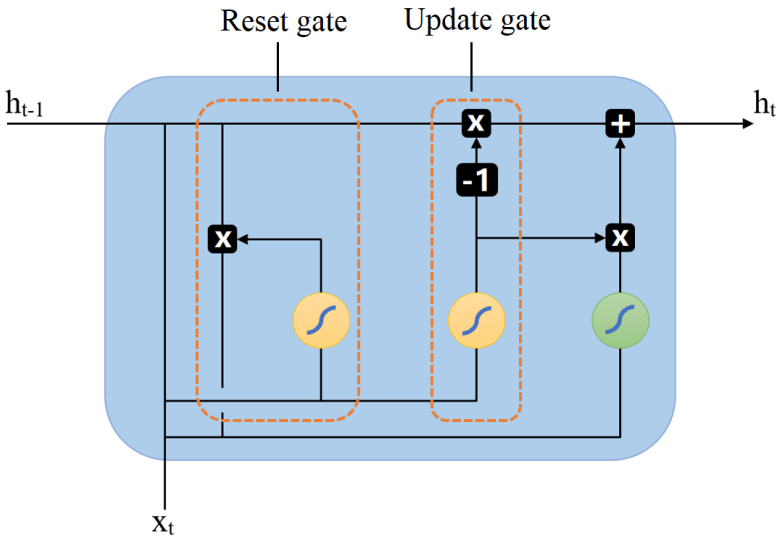


Figure 3.8. Gated recurrent unit. The GRU consists of a hidden state, a reset gate, and an update gate. In the GRU, the sigmoid (yellow) and tanh (green) activation functions are used. The GRU computes point-wise multiplications several times (marked with an 'x' in a black box) and one addition (marked as a '+' in a black box). The current input feature map to the GRU is x_t whereas the previous hidden and updated hidden state are marked as h_{t-1} and h_t, respectively. Inspired by [Chung et al., 2014]

When an input feature map entered the GRU, it was concatenated with the hidden state from the previous GRU. The concatenated information was passed through a sigmoid activation function in the reset gate. Through the sigmoid activation function, a point-wise multiplication was performed, where the values closer to 1 were kept, otherwise forgotten. The sigmoid activation function transformed the concatenation into values between 0 and

1. For more information about the sigmoid activation function, see appendix B.3.3. The computation of the reset gate, r_j, was as followed:

$$r_j = \sigma([W_r x]_j + [U_r h_{(t-1)}]_j), \tag{3.2}$$

where σ is the sigmoid activation function, $[.]_j$ is the j-th element in a matrix, x is the input, $h_{(t-1)}$ is the previous hidden state, and W_r and U_r are weights that are learned. [Cho et al., 2014]

Next, the concatenation of the input feature map and the hidden state was processed through the update gate. The update gate protected the memory content in the hidden state from the perturbation of irrelevant features in the input feature maps. The update gate consisted of a sigmoid activation function to decide which information from the concatenation should be added to the hidden state and which should be forgotten. The computations from the update gate, z_j, were as the following:

$$z_j = \sigma([W_z x]_j + [U_z h_{t-1}]_j). \tag{3.3}$$

where W_z and U_z are weights that were learned. Afterward the hidden state h_j was updated by

$$h_j^{(t)} = z_j \tilde{h}_j^t + (1 - z_j) h_j^{(t-1)}, \tag{3.4}$$

where $\tilde{h}_j^t$ was used to protect the following units from irrelevant features in the current input feature map by concatenation of the output from the reset gate and the current input using a tangent hyperbolic (tanh) activation function. The tanh transformed the concatenation into values between -1 and 1 to normalize the values of each element in the input feature map [Chollet et al., 2018]. For extended information about the tanh, see appendix B.3.5. The equation of the $\tilde{h}_j^t$ was computed by

$$\tilde{h}_j^t = tanh([Wx]_j + [U(r \odot h_{t-1})]_j), \tag{3.5}$$

where *tanh* is the tanh activation function and W and U are the learnable weights. Thus, the GRU carried important features from the early input feature maps in a CCTA volume over the entire sequence, hence, keeping the long-term dependencies [Chung et al., 2014]

Components of the RNN

The RNN achieved the effect of adding layers much like the CNN: the addition of more layers improved the expressive ability of the RNN. However, because of the difficulties associated with the training of an RNN due to the vanishing gradient problem, the

number of recurrent layers was kept at three recurrent layers. [Long and Zeng, 2022] The vanishing gradient problem refers to the disappearance of the error when this propagates back through the RNN to update the network weights [Hochreiter, 1998]. The number of GRU in each recurrent layer referred to the projection of input data onto an n-dimensional representation space. Thus, the more GRU the recurrent layers contained, the more complex information could be learned. However, too many GRU had a bigger computationally expense and may have led to the learning of unwanted patterns specific to the training data set. [Chollet et al., 2018]

In the RNN, recurrent dropout was applied, which ensure the same pattern of zeroed hidden units was applied to each input feature map from a CCTA volume, instead of randomly zeroing hidden units per input feature map. Recurrent dropout was used in the inner recurrent activations. The use of temporally randomly dropout would course disruption in the propagation learning error. [Chollet et al., 2018] For further explanation of dropout, see appendix B.5.

3.3.3 Fully connected convolutional network for analysis of patient data

The use of a FCNN to process the patient data simultaneously with the sequential frame analysis by the CNN-RNN in the CAD network was inspired by Spasov et al. [2018], who used a multi-modal CNN to predict Alzheimer's disease in MRI scans. The inspiration of a study regarding MRI scans was drawn as the literature search of the present study, showed in appendix A, did not reveal any study using CCTA volumes and the corresponding patient data to identify CAD. The purpose of using patient data to guide the identification of a disease was similar between the study by Spasov et al. [2018] and this study. Thereby, the FCNN architecture from the study by Spasov et al. [2018] was used as a starting point for the patient data network in this study.

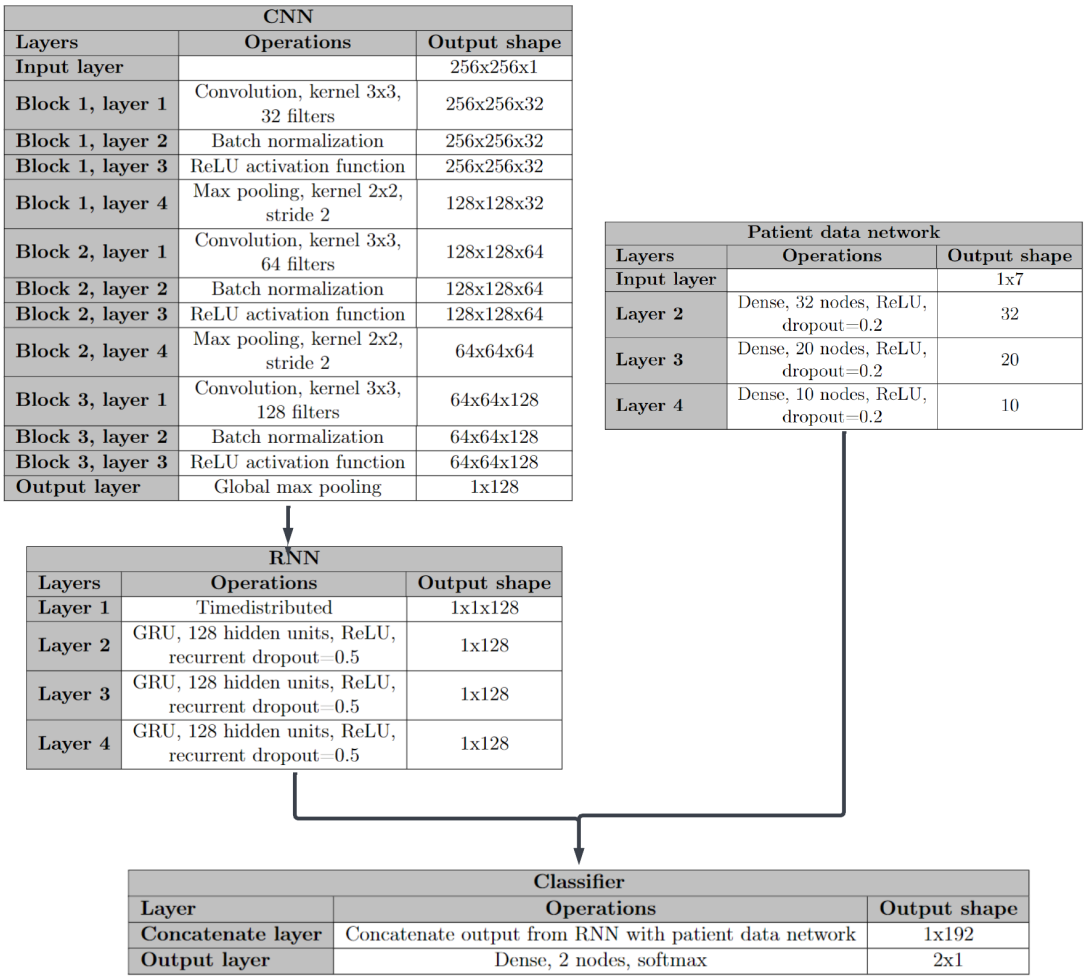
Originally, the network by Spasov et al. [2018] consisted of three fully connected layers, also called dense layers, that connect each neuron in one layer to each neuron in the following layer [Chollet et al., 2018]. For elaborated information about FCNN, see appendix B.6. The dense layers contained 32, 20, and 10 nodes, and each layer was followed by an exponential linear unit (ELU) as the activation function. The ELU activation function maintained input elements above the bias while introducing an exponential parameter slope for input elements below the bias. Thereby, input elements below the bias were preserved compared to the ReLU activation function. Sharma et al. [2017] The theoretical review of the ELU activation function can be seen in appendix B.3.2. The architecture by Spasov et al. [2018] was used for a patient data network in the present study as well. However, it differs by applying dropout with a dropout rate of 0.2 between the two first dense layers to prevent overfitting to the training data. In table 3.6, the architecture of the patient data network can be seen.

Patient data network		
Layers	**Operations**	**Output shape**
Input layer		1x7
Layer 1	Dense, 32 nodes, ReLU, dropout=0.2	32
Layer 2	Dense, 20 nodes, ReLU, dropout=0.2	20
Layer 3	Dense, 10 nodes, ReLU, dropout=0.2	10

Table 3.6. An overview of the patient data network architecture.

3.3.4 Classification of the CCTA volumes

When the CCTA volumes and the patient data had been processed through the CNN-RNN and the patient data network, the outputs from the networks were concatenated. This was done to use both the CCTA volumes and the patient data to classify the volumes as either containing CAD or non-CAD. The concatenated layer was then given as input to a dense layer with a softmax activation function. The study by Zreik et al. [2018b] used a softmax activation function as the classification was multi-classed. However, the present study was a binary classification problem, i.e., if the CCTA volumes were CAD or non-CAD. Thus, a sigmoid activation function should be used. See appendix B.3.3 for more information about the sigmoid activation function. However, based on the results from the hyper-parameter optimization in appendix D.4 should the softmax activation function be applied. The output of the softmax activation function was a probability of the CCTA volume belonging to the CAD class and the non-CAD class. Thus, the predicted label of each CCTA volume was the label with the highest probability. [Sharma et al., 2017] See appendix B.3.4 for more information regarding the softmax activation function. An overview of the collected architecture of the full CAD network is presented in figure 3.9.



CNN		
Layers	**Operations**	**Output shape**
Input layer		256x256x1
Block 1, layer 1	Convolution, kernel 3x3, 32 filters	256x256x32
Block 1, layer 2	Batch normalization	256x256x32
Block 1, layer 3	ReLU activation function	256x256x32
Block 1, layer 4	Max pooling, kernel 2x2, stride 2	128x128x32
Block 2, layer 1	Convolution, kernel 3x3, 64 filters	128x128x64
Block 2, layer 2	Batch normalization	128x128x64
Block 2, layer 3	ReLU activation function	128x128x64
Block 2, layer 4	Max pooling, kernel 2x2, stride 2	64x64x64
Block 3, layer 1	Convolution, kernel 3x3, 128 filters	64x64x128
Block 3, layer 2	Batch normalization	64x64x128
Block 3, layer 3	ReLU activation function	64x64x128
Output layer	Global max pooling	1x128

Patient data network		
Layers	**Operations**	**Output shape**
Input layer		1x7
Layer 2	Dense, 32 nodes, ReLU, dropout=0.2	32
Layer 3	Dense, 20 nodes, ReLU, dropout=0.2	20
Layer 4	Dense, 10 nodes, ReLU, dropout=0.2	10

RNN		
Layers	**Operations**	**Output shape**
Layer 1	Timedistributed	1x1x128
Layer 2	GRU, 128 hidden units, ReLU, recurrent dropout=0.5	1x128
Layer 3	GRU, 128 hidden units, ReLU, recurrent dropout=0.5	1x128
Layer 4	GRU, 128 hidden units, ReLU, recurrent dropout=0.5	1x128

Classifier		
Layer	**Operations**	**Output shape**
Concatenate layer	Concatenate output from RNN with patient data network	1x192
Output layer	Dense, 2 nodes, softmax	2x1

Figure 3.9. An overview of the CAD network architecture.

For the CAD network to predict the correct labels for the CCTA volumes, the networks were trained.

3.4 Training of the CAD network

The CAD network was trained to label the input CCTA volume as either CAD or non-CAD. The training of the CAD network consisted of parameter adjustments in the layers of the CAD network, such that the CAD network would correctly map the input CCTA volumes to the associated targets. The parameter adjustments referred to the filter coefficients in the convolution kernel, the weights in the reset- and update gate in the GRU, and the biases of the activation functions. Initially, the parameters were randomly initialized. [Chollet et al., 2018] The hyper-parameters influencing the training of the CAD network are listed in table 3.7 and accounted for in the following.

Batch size	Optimization algorithm	Learning rate
1	Adam	0.00001

Table 3.7. Learning hyper-parameters of the CAD network, controlling the training process.

The training of the CAD network was an iterative process that consisted of a training loop with the following four steps:

1. Send a batch of CCTA volumes with the corresponding target label through the CAD network to obtain predictions.
2. Calculate the loss of the CAD network on the batch by comparing the predicted labels of the CCTA volume with the target label of the CCTA volume.
3. Calculate the gradient of the loss
4. Update the parameters in the CAD network by using the loss magnitude controlled by the learning rate in a way that reduces the loss on the current batch.

The training loop of the CAD network was repeated until the loss converged towards the minimum loss, indicating that the learning of the CAD network did not improve further. [Chollet et al., 2018]

The gradient update was based on the Adam algorithm, an extension to the stochastic gradient descent (SGD) optimization algorithm [Kingma and Ba, 2014; Chollet et al., 2018]. The use of the optimizer was not changed from the original architecture by Zreik et al. [2018b]. Adam was an adaptive learning rate optimization algorithm, which calculates individual learning rates for each parameter in the CAD network [Kingma and Ba, 2014]. For more information about the Adam algorithm, see appendix C.1.1. The Adam algorithm used a forward pass to process a batch of the training data through the CAD network to obtain a prediction of whether the CCTA volume was containing CAD or not. In the present study, a batch size of one was used due to limited memory on the used GPU. Thus, a batch consisted of one CCTA volume. The training data was passed through the CAD network until every CCTA volume in the training set had been processed. For each batch of the training data, the parameters were adjusted in a direction that would lower the loss value based on a feedback signal. The loss value should be as small as possible indicating that the predicted and the targeted label for the CCTA volume were similar and zero if they were the same. The feedback signal was the loss value calculated by a loss function at each time step. The loss function used in the CAD network was categorical cross-entropy (CCE). How much the parameters were adjusted according to the loss function was controlled by the learning rate. [Chollet et al., 2018] The present study applied a learning rate of 0.00001 based on the hyper-parameter optimization results in appendix D.5. The adjustment of the parameters in the CNN-RNN was achieved through the backpropagation through time (BPTT) algorithm, which implemented a gradient descent optimization algorithm. BPTT was used when training an RNN. However, the patient data network used the backpropagation algorithm instead for the adjustment of parameters. BPTT differed from backpropagation by unfolding the CNN-RNN, shown in figure 3.7 in section 3.3.2, to perform a backward pass through the whole input sequence to adjust the CNN-RNN parameters. This was done by using the gradient descent optimization algorithm which aimed to minimize the loss function. For more information about the gradient descent optimization algorithm, see appendix C.1. The gradient descent algorithm calculated the gradient of the loss function at every time step and accumulated them in the CNN-RNN. The CNN-RNN was then folded again to update the parameters. [Guo, 2013; Chollet et al., 2018] Whereas the backpropagation algorithm operated the loss value backward through the patient data networks by layerwise calculating the gradient by using the gradient

descent algorithm to minimize the loss. [Chollet et al., 2018]

3.4.1 Categorical cross-entropy loss

The loss function used in the CAD network was CCE since the activation function in the output layer was softmax. CCE calculated the distance between the probability of the predicted output label and the target label for the CCTA volumes. [Chollet et al., 2018] The function for CCE is shown in equation 3.6:

$$CCE = -\frac{1}{N}\sum_{i=1} y_i \cdot log(\hat{y}_i) \tag{3.6}$$

where y_i is the targeted CAD or non-CAD output label and $p(y_i)$ is the probability of the predicted output label corresponding to the targeted output label for the CCTA volume. [Martinez and Stiefelhagen, 2018]

3.5 Validation of the identification of CAD

To validate the performance of the CAD network and to compare the performance to similar studies, the validation method of the AUROC, confusion matrix, and F1 score had been used. These validation methods were often used by other studies as performance measurements, as seen in section 2.3, thus, for comparison among studies, they are used in this study as well. Moreover, the CNN-RNN and patient data network were validated separately to see how each network influenced the performance of the CAD network.

3.5.1 Confusion matrix

The confusion matrix was used to illustrate the performance of the CAD network by visualizing the relationship between targeted labels and the predicted labels of CAD and non-CAD CCTA volumes. A conceptual confusion matrix is illustrated in figure 3.10. The rows of the confusion matrix represented the targeted labels of the CCTA volumes and the columns represented the predicted labels of the CCTA volumes. [Novaković et al., 2017] The threshold used for the confusion matrix was 0.5 as the threshold used for binary classification was 0.5 [Thanaki, 2018].

		Predicted labels	
		Containing CAD	**Not containing CAD**
Targeted labels	**Containing CAD**	True positive (TP)	False negative (FN)
	Not containing CAD	False positive (FP)	True negative (TN)

Figure 3.10. The confusion matrix. The relationship between the targeted labels and the predicted labels of the CCTA volumes as either containing CAD or non-CAD is shown as a true positive (TP), false negative (FN), false positive (FP), and true negative (TN). This visualizes how well the CAD network predicted the CCTA volumes labels compared to their targeted labels.

In the confusion matrix, the performance of the CAD network was measured by the number of true positive, false positive, true negative, and false negative predicted labels. The number of true positives showed how many predicted CCTA volumes with CAD were correct and the number of false-positive indicated the number of predicted CCTA volumes without CAD but labeled so. Likewise, the number of true negatives illustrated the correctly predicted CCTA volumes without CAD and the number of false-negative labels showed the number of CCTA volumes with CAD that should have been predicted as CCTA volumes with CAD but were predicted as CCTA volumes without CAD. [Novaković et al., 2017]

From the confusion matrix, the sensitivity and the specificity were derived, also known as the true positive rate (TPR) and the true negative rate (TNP)

$$TPR = \frac{TP}{TP + FN}, \tag{3.7}$$

where TP is the true positive and FN is the false negative, and

$$TNR = \frac{TN}{FP + TN}, \tag{3.8}$$

where the TN is the true negative and the FP is the false positive. The sensitivity and the specificity described the accuracy of the CAD network. [Novaković et al., 2017]

3.5.2 F1 score

The F1 score was another metric used to measure the accuracy of the CAD network. The F1 score was the harmonic mean of the TPR from equation 3.8 and the positive predicted value (PPV):

$$PPV = \frac{TP}{PP}, \tag{3.9}$$

where PP is the positively predicted labels. Thereby, the F1 score was calculated by

$$F_1 = \frac{2}{PPV^{-1} + TPR^{-1}}. \tag{3.10}$$

In case of perfect performance of the CAD network, where every label of the CCTA volumes was predicted correctly, the F1 score would be 1. Contrary, the F1 score would be 0, if non of the predicted labels were correct. The same threshold from the confusion matrix was used for the F1 score. [Novaković et al., 2017]

3.5.3 The receiver operating characteristic curve

The receiver operating characteristic (ROC) curve was used to show the ability of the CAD network to correctly classify CCTA volumes as either containing CAD or not at various discrimination thresholds. It is used when having balanced data set. A conceptual illustration of a ROC curve is shown in figure 3.11. [Novaković et al., 2017]

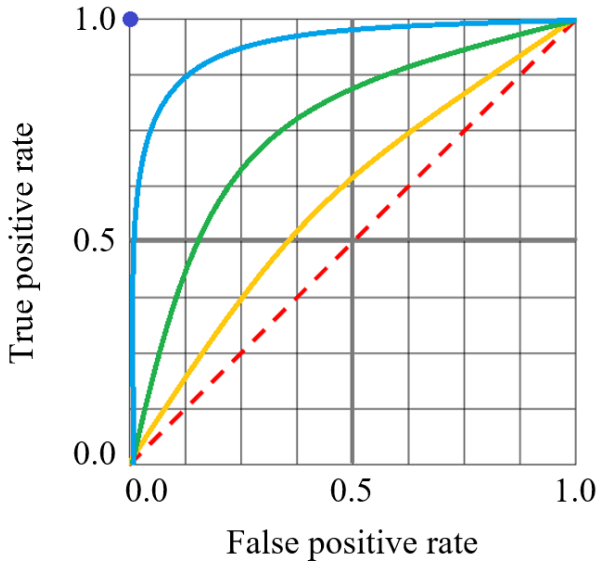


Figure 3.11. The receiver operating characteristic curve. Different curves show different performances, where the blue curve represents a good performance, and the yellow line represents a less accurate performance. The representation of random prediction performance is marked with the red dotted line, called the line of no-discrimination. The dark blue point represents the best possible performance of the CAD network. Inspired by [Novaković et al., 2017]

When plotting the ROC curve, the TPR of the classification reflecting the sensitivity of the classification performance was held against the false positive rate (FPR) representing 1 - specificity. The FPR was calculated by equation 3.11. Thus, the ROC curve illustrated the trade-off relationship between the rate of correctly classified CCTA volumes containing CAD versus the rate of CCTA volumes not containing CAD but classified so. [Novaković et al., 2017]

$$FPR = \frac{FP}{TN + FP} \tag{3.11}$$

When the TPR and the FPR were held against each other, the best possible performance of the CAD network would yield a coordinate of (0,1) in the ROC space, representing a 100% sensitivity and specificity, and the CAD network would have performed a perfect classification. If the CAD network performed random guesses, the ROC curve would be placed along the line of no-discrimination, as shown in figure 3.11, thus, the further above the line of no-discrimination the ROC curve of the CAD network performance was the better the performance of the CAD network. [Novaković et al., 2017]

Results 4

The following chapter includes the results training of the CNN-RNN, the patient data network, and the full CAD network.

In this study, a total of 361 patients were included and 17 patients were excluded due to missing patient data. Out of the 361 patients, 148 patients had CAD, and 213 did not. Baseline characteristic for the included patients are summarised in table 4.1.

	No. of subjects	**Male/ Female**	**Age (years)**	**Weight (kg)**	**Height (cm)**	**BMI (kg/m^2)**
CAD	148	103/45	60.1 ± 9.3	82.3 ± 14.5	173.8 ± 8.4	27.2 ± 4.2
Non-CAD	213	110/103	60.8 ± 7.4	80.2 ± 14.9	172.1 ± 9.3	27 ± 4.1
	Systolic blood pressure (mmHg)	**Diastolic blood pressure (mmHg)**	**Cholesterol (mmol/L)**	**Smoker/ Non smoker**	**Diabetic/ Non diabetic**	**Risk factors**
CAD	145.3 ±19.7	85 ±11.6	5.5 ± 1.5	84/64	19/129	3.1 ± 1.1
Non-CAD	141.2 ± 18.3	83.3 ± 10.4	5.6 ±1.2	114/99	12/101	2.7 ± 1

Table 4.1. Baseline of the study subjects. The mean of the age, weight, height, BMI, systolic and diastolic blood pressure, cholesterol level, and the number of risk factors have been found along with the standard deviation.

An overview of the architecture used for training the final CAD network can be seen in figure 3.9 in section 3.3.4. In the training of the CNN-RNN, the patient data network, and the full CAD network, the method of early stopping was implemented to prevent the model of the networks from overfitting to the training data. Thus, the training stopped if validation loss did not decrease over 10 epochs. The 10 epoch was chosen to ensure that the decrease in validation loss was not a coincidence, thus, the training of the networks was not stopped before the losses converge. In the following, the results from the performance of the models of the CNN-RNN, the patient data network, and the full CAD network will be presented.

4.1 CNN-RNN

The CNN-RNN was trained separately to investigate how the CNN-RNN influenced the performance of the CAD network. The curves of the training- and validation losses during training of the final model of the CNN-RNN are presented in figure 4.1.

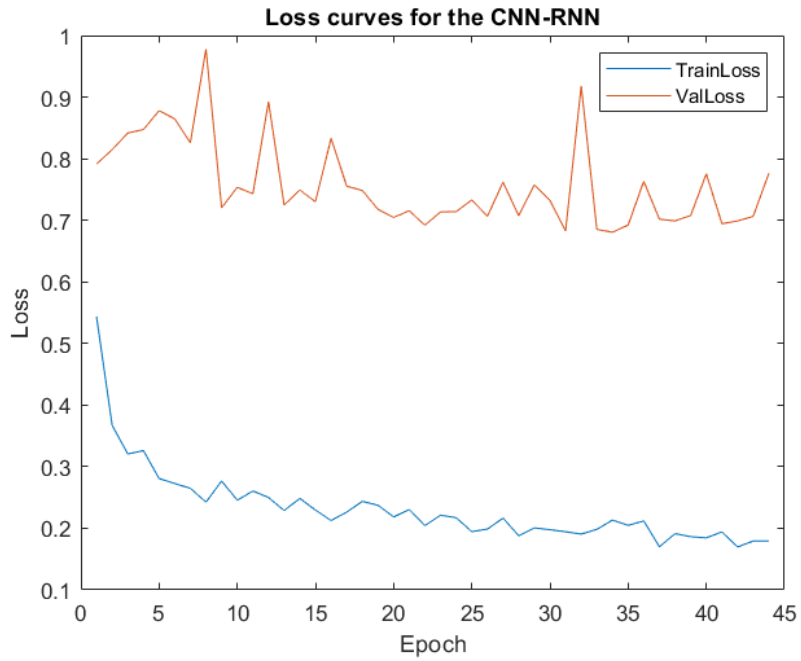


Figure 4.1. Loss curves for the training- and validation loss from the training of the final model of the CNN-RNN. The y-axis represents the loss and the x-axis represents the number of epochs. The validation loss curve is illustrated in orange and the training loss curve is marked in blue.

The loss curves for the CNN-RNN showed a descending tendency for the training- and validation loss. However, the validation loss only decreased 0.1 until epoch 22 hereafter it flattened. The validation loss ranged between 0.7 to 1 whereas the training loss decreased in the range 0.6 to 0.2. This created a gap between the losses, which indicated overfitting. Thus, the CNN-RNN could not generalize to unseen data. The final model of the CNN-RNN was obtained after the last epoch, where the training loss was 0.1794 and the validation loss was 0.7764.

From the accuracy curves for the CNN-RNN in figure 4.2, an ascending tendency in the training accuracy was observed whereas the validation accuracy generally did not change. The validation accuracy decrease 0.08 in accuracy from epoch 1 to 2 where after it remained constant. The accuracy curve for the training loss illustrated that the CNN-RNN performance on the training data improved over time. However, the CNN-RNN did not change in performance on the validation data through the whole training of the CNN-RNN.

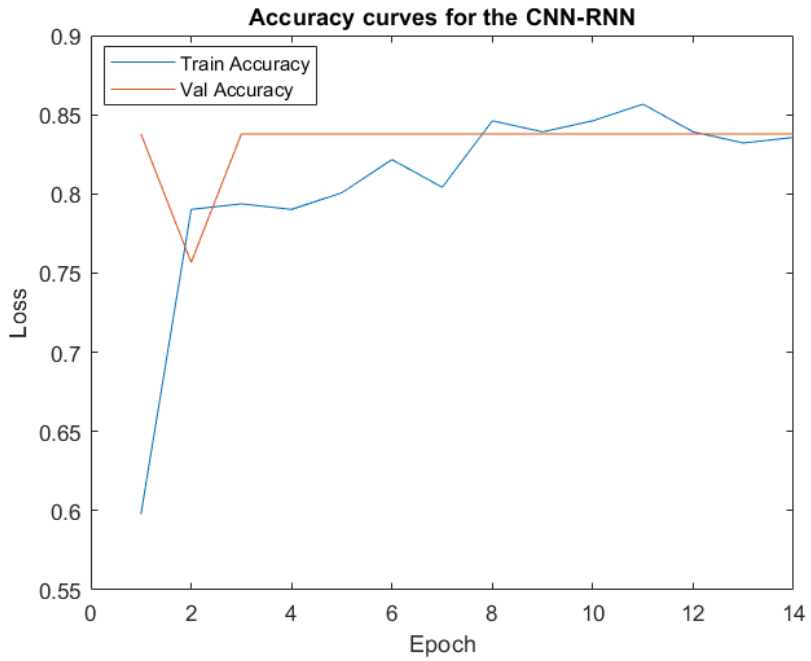


Figure 4.2. Accuracy curves for the training- and validation accuracy after training the final model of the CNN-RNN. The y-axis represents the accuracy and the x-axis represents the number of epochs. The validation accuracy curve is illustrated with orange and the training accuracy curve is marked as blue.

The confusion matrix presented in figure 4.3 with a threshold of 0.5, indicated the performance of the final model of the CNN-RNN on the test data set with 38 patients. The confusion matrix illustrated that 7 CCTA volumes containing CAD and 14 CCTA volumes without CAD were classified correctly. Moreover, 10 CCTA volumes without CAD and 7 CCTA volumes with CAD were not classified correctly. The CCTA volumes with CAD contained different amounts of stenoses.

		Predicted labels	
		Containing CAD	**Not containing CAD**
Targeted labels	**Containing CAD**	7	7
	Not containing CAD	10	14

Figure 4.3. Confusion matrix with a threshold of 0.5 for the performance of the CNN-RNN on the test data set. The test set contained 38 patients, where 14 patients had CAD and 24 patients did not.

In table 4.2, the distribution of stenosis in the correctly classified and misclassified CCTA volumes containing CAD by the CNN-RNN is presented.

	CNN-RNN	
	Correctly classified	**Misclassified**
1 stenosis	4	3
2 stenoses	3	1
3 stenoses	0	3

Table 4.2. An overview of the distribution of stenosis in the correctly classified and misclassified CCTA volumes containing CAD by the CNN-RNN.

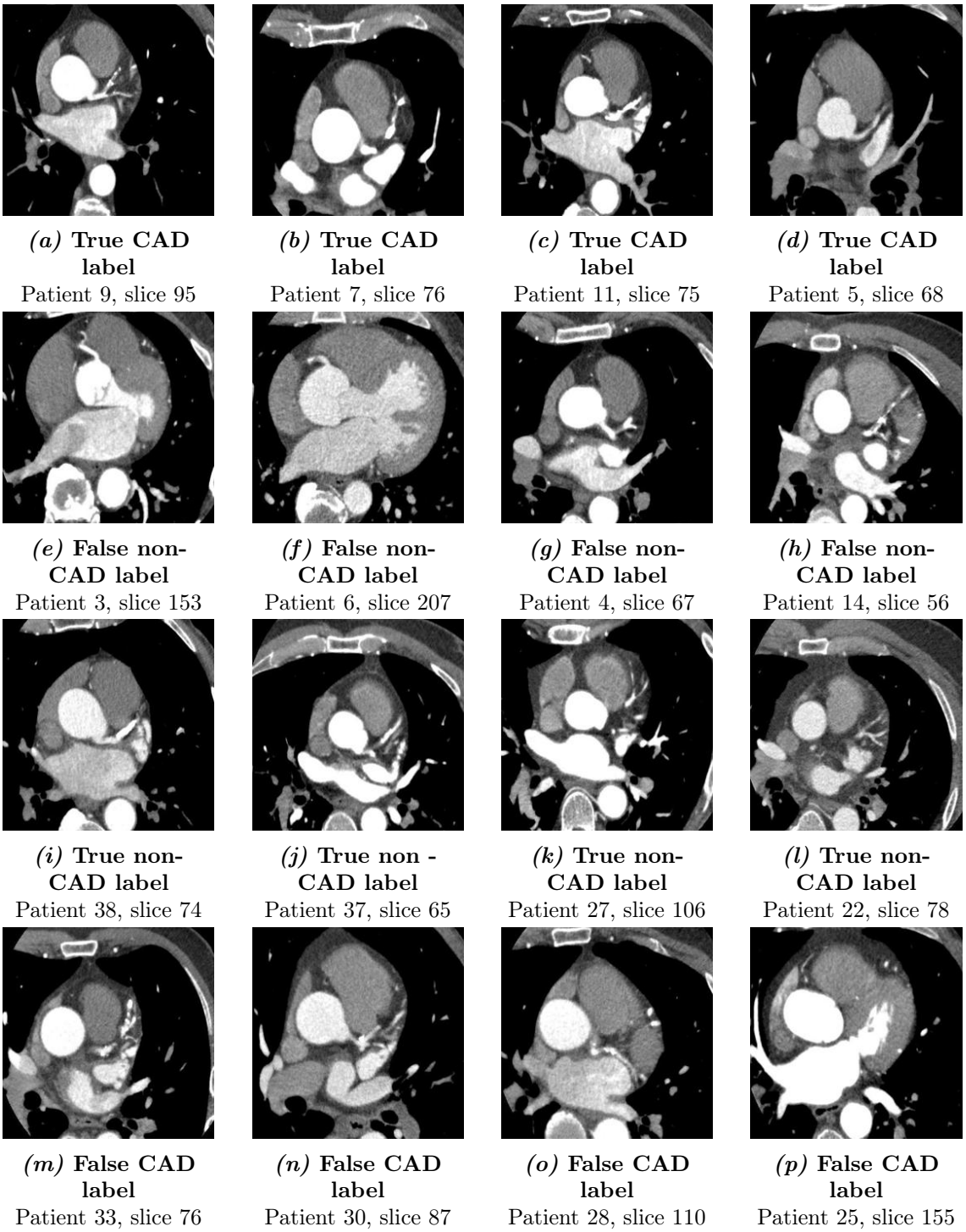
The average F1 score for final model of the CNN-RNN model along with the F1 score for the labels CAD and non-CAD is presented in table 4.3. Additionally, the sensitivity, specificity, accuracy and the AUROC for the CNN-RNN was also presented.

CNN-RNN						
F1 score CAD	**F1 score Non-CAD**	**Average F1 score**	**Sensitivity**	**Specificity**	**Accuracy**	**AUROC**
0.45	0.62	0.54	0.5	0.58	0.55	0.54

Table 4.3. The F1 score for the labels CAD and non-CAD, and the average F1 score for the CNN-RNN along with the sensitivity, specificity, accuracy, and AUROC.

The average F1 score of 0.54 indicated that the CNN-RNN had the ability to classify approximately half of the CCTA volumes correct. This is further supported by an accuracy at 0.55. Moreover, the CNN-RNN can classify half of the CCTA volumes with CAD correctly and correctly identified a bit over half of the CCTA volumes without CAD. This was illustrated with a sensitivity at 0.5. and a sensitivity at 0.58, respectively. The data set used in the present study was imbalanced as the data set contained more CCTA volumes without CAD compared with CCTA volumes with CAD. This was illustrated by the F1 score for CAD and non-CAD as the the F1 score for CAD was 0.45 and 0.62 for non-CAD. This illustrated that the CNN-RNN correctly classified over half of the CCTA volumes without CAD. But the CNN-RNN predicted under half of the CCTA volumes with CAD correctly. Thus, the CNN-RNN was better at predicting CCTA volumes without CAD.

The representative slices from different CCTA volumes from the test data set are illustrated in figure 4.4, to evaluate if tendencies between the true positive-, true negative-, the false positive-, and the false negative predicted labels were present.



(a) **True CAD label**
Patient 9, slice 95

(b) **True CAD label**
Patient 7, slice 76

(c) **True CAD label**
Patient 11, slice 75

(d) **True CAD label**
Patient 5, slice 68

(e) **False non-CAD label**
Patient 3, slice 153

(f) **False non-CAD label**
Patient 6, slice 207

(g) **False non-CAD label**
Patient 4, slice 67

(h) **False non-CAD label**
Patient 14, slice 56

(i) **True non-CAD label**
Patient 38, slice 74

(j) **True non - CAD label**
Patient 37, slice 65

(k) **True non-CAD label**
Patient 27, slice 106

(l) **True non-CAD label**
Patient 22, slice 78

(m) **False CAD label**
Patient 33, slice 76

(n) **False CAD label**
Patient 30, slice 87

(o) **False CAD label**
Patient 28, slice 110

(p) **False CAD label**
Patient 25, slice 155

Figure 4.4. CCTA slices from 16 patients from the test data set. The labels of the slices is predicted by the CNN-RNN. The rows represent (*a*-*d*) the true positive labels (**True CAD label**), (*e*-*h*) the false negative labels (**False non-CAD label**), (*i*-*l*) the true negative labels (**True non-CAD label**), and (*m*-*p*) the false positive labels (**False CAD label**).

The CCTA slices in figure 4.4 showed similar information regarding the heart and the coronary arteries. However, some of the CCTA slices (slice *d*, *f*, *i*, *l*, *n*, and *o*) were a bit more blurry than the others and the contrast of these CCTA slices seemed to be slightly lower compared to the rest of the CCTA slices. Further, the amount of surrounding tissue,

such as bone and lungs among others, is differentiated in each of the CCTA slices. Another differentiating information among the CCTA slices was the anatomical representation according to the slice number. CCTA slices with approximately the same number did not show the same anatomically information.

ROC for the CNN-RNN was used to show the ability of the final model of the CNN-RNN to correctly classify if CCTA volumes contained CAD or not. ROC illustrated in figure 4.5 was made from the test data set.

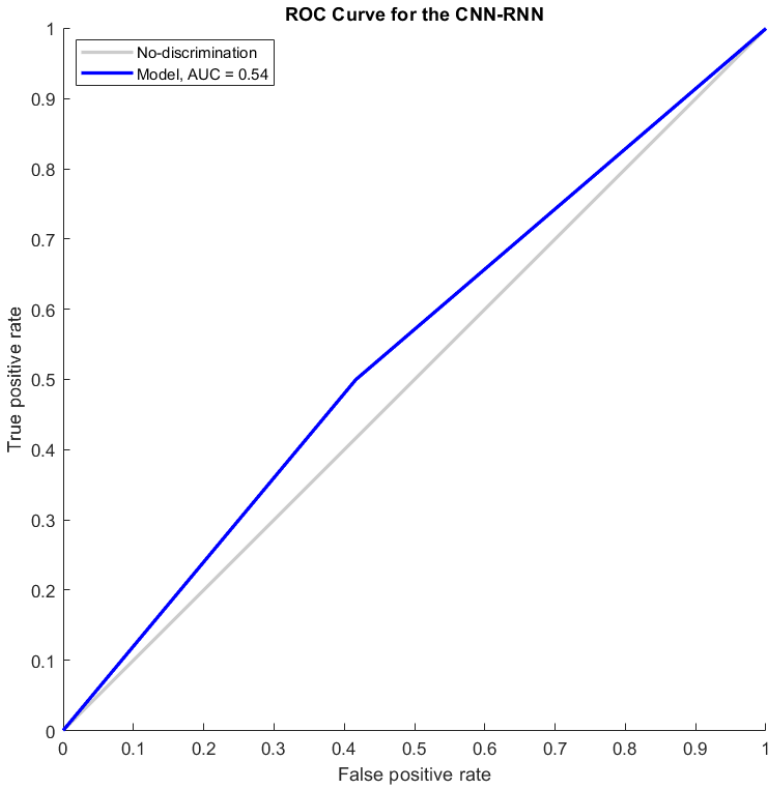


Figure 4.5. ROC for the test data set of the final model of the CNN-RNN. The y-axis represents the true positive rate and the x-axis represents the false positive rate. The blue line illustrates the ROC curve for the test data set and the grey line indicates where the true positive rate and false positive rate are equal.

The ROC and an AUROC of 0.54 indicated that the final model of the CNN-RNN randomly classified the CCTA volumes as either CAD or non-CAD. The ROC for the CNN-RNN illustrated that the curve representing the performance of the CNN-RNN model differed slightly from the line of no-discrimination as the curve for the CNN-RNN model slightly moved up towards the upper left corner of the graph. This indicates that the CNN-RNN performed random classifications.

4.2 Patient data network

The patient data network was trained separately as with CNN-RNN. However, this was done to investigate the impact of the patient data on the performance of the CAD network. The training- and validation loss curve for the training of the patient data network are presented in figure 4.6.

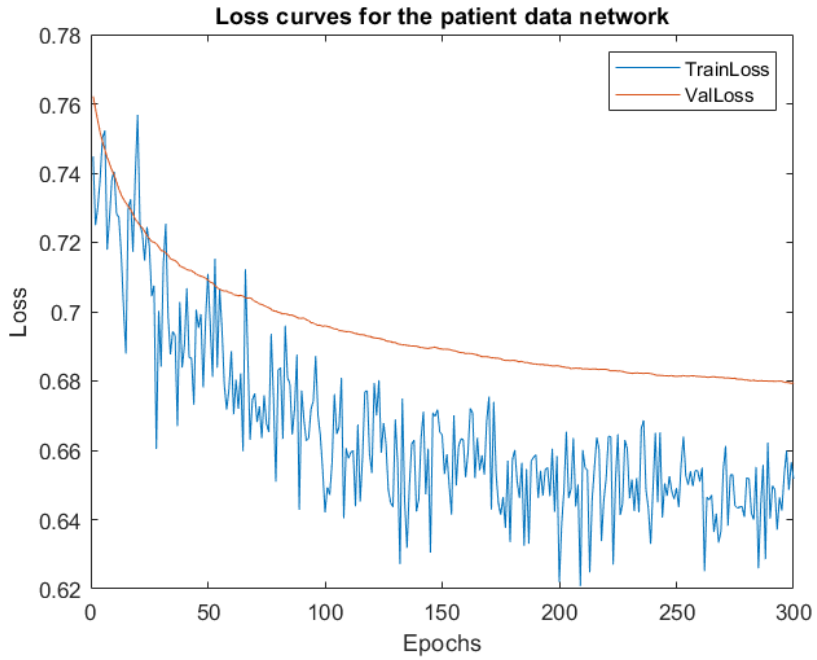


Figure 4.6. Loss curves for the training- and validation loss after training the final model of the CAD network. The y-axis represents the loss and the x-axis represents the number of epochs. The orange line illustrates the validation loss and the blue line illustrates the training loss.

The loss curve from figure 4.6 for the patient data network differed from the CNN-RNN loss curves by having a descending tendency in both the training- validation loss. Furthermore, the loss curves for the patient data network does not indicate overfitting to the training data as the loss curves for the CNN-RNN. After approximately epoch 200, the training loss begins to flatten. Moreover, the validation loss curve flattened more after epoch 50 compared to the training loss which created a gap between the training- and validation loss. From approximately epoch 100 to the last epoch at 300, the loss difference increased in the range from 0,01 to 0,02. The final model of the patient data network was obtained after the last epoch, where the training loss was 0.6517 and the validation loss was 0.6794.

The accuracies of the training and validation during the training of the patient data network model were plotted in figure 4.7.

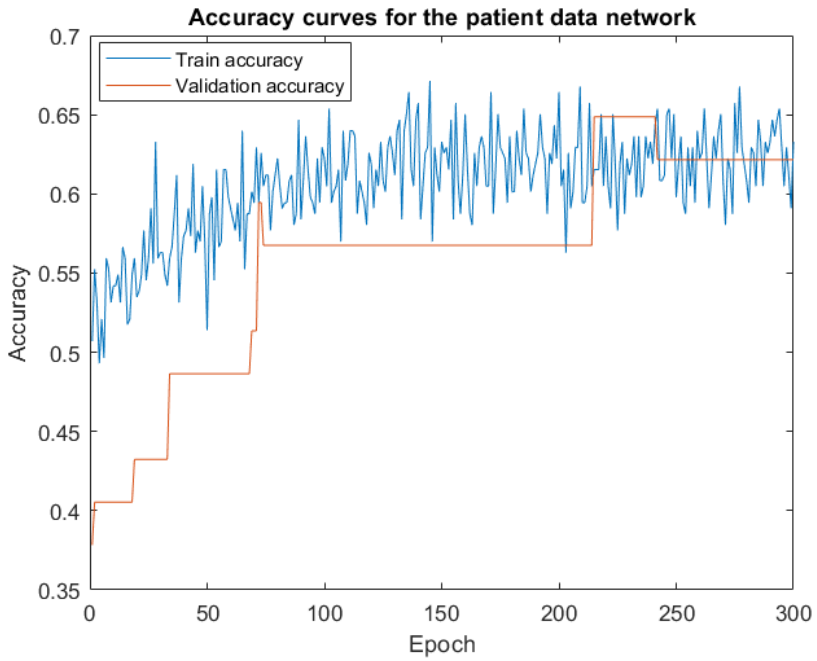


Figure 4.7. Accuracy curves for the training- and validation accuracy after training the final model of the patient data network. The y-axis represents the accuracy and the x-axis represents the number of epochs. The validation accuracy curve is illustrated in orange and the training accuracy curve is marked in blue.

The training- and validation accuracy curves for the patient data network showed an ascending tendency in both the training- and validation accuracy. This indicated that the patient data network performance for both the training- and validation data improved through the training of the network. However, the validation accuracy increased in steps through the whole training indicating periods where patient data network did not improve in performance.

From the confusion matrix with a threshold at 0.5 presented in figure 4.8, it was illustrated that 10 patients without CAD and 13 patients with CAD were classified correctly. Moreover, 14 patients without CAD and one patient with CAD were not classified correctly.

		Predicted labels	
		Containing CAD	**Not containing CAD**
Targeted labels	**Containing CAD**	13	1
	Not containing CAD	14	10

Figure 4.8. Confusion matrix with a threshold of 0.5 for the performance of the patient data network on the test data set. The test set contained 38 patients, where 14 patients had CAD and 24 patients did not.

From table 4.4, the distribution of stenosis in the correctly classified and misclassified CCTA volumes containing CAD by the patient data network can be observed.

	Patient data network	
	Correctly classified	**Misclassified**
1 stenosis	7	0
2 stenoses	4	0
3 stenoses	1	1

Table 4.4. An overview of the amount of stenosis in the correctly classified and misclassified CCTA volumes containing CAD by the patient data network.

A comparison of predicted labels between the CNN-RNN and the patient data network is illustrated in figure 4.9. When comparing if the CNN-RNN and the patient network have classified or misclassified the same CCTA volumes, it was observed that they were likely to classify some of the same CCTA volumes.

Predictions of the CNN-RNN compared to the patient data network

		Predicted labels	
		Containing CAD	**Not containing CAD**
Targeted labels	**Containing CAD**	5	1
	Not containing CAD	4	5

Figure 4.9. The common predictions of the CNN-RNN and the patient data network. The **Predicted labels** are the predictions made by the networks and the **Targeted labels** are the true labels of the CCTA volumes.

The sensitivity, specificity, accuracy, and the AUROC for the final model of the patient data network are presented in table 4.5 along with the F1 score for CAD, non-CAD, and the average F1 score. The F1 scores, sensitivity, specificity, accuracy, and the AUROC of the CNN-RNN are presented as well for comparison.

CNN-RNN						
F1 score CAD	**F1 score Non-CAD**	**Average F1 score**	**Sensitivity**	**Specitivity**	**Accuracy**	**AUROC**
0.45	0.62	0.54	0.5	0.58	0.55	0.54
Patient data network						
F1 score CAD	**F1 score Non-CAD**	**Average F1 score**	**Sensitivity**	**Specificity**	**Accuracy**	**AUROC**
0.63	0.57	0.60	0.92	0.42	0.61	0.67

Table 4.5. The F1 score for the labels CAD and non-CAD, and the average F1 score for CNN-RNN and the patient data network along with the sensitivity, specificity, accuracy, and AUROC.

From table 4.5, it is illustrated by the sensitivity at 0.90 that the patient model has the ability to identify patients with CAD correctly. However, a specificity of the patient data network of 0.42 indicated that the network was mislabeling a lot of patients without CAD as patients with CAD. The average F1 score of 0.60 indicated that the patient data network could classify over half of the patients correctly. Further, the F1 score for CAD was 0.63 and 0.57 for non-CAD. This illustrated that the patient data network classified over half of the patients with CAD and without CAD correctly. The performance between the patient data network and the CNN-RNN differed as the patient data network was better at predicting patients with CAD correctly. The difference in performance was further illustrated by the accuracy of the patient data network and the CNN-RNN was 0.61 and 0.55, respectively, which also illustrated that the performance of the patient data network was slightly better.

The ROC for the patient data network, illustrated in figure 4.10, was made from the test data set and showed the performance of the patient data network. The AUROC and ROC illustrated that the patient data network performed better than the CNN-RNN. This was indicated by an AUROC of 0.67 for the patient data network and the ROC was moving more op to the upper left corner of the graph compared with the ROC of the CNN-RNN in figure 4.5. This illustrated that the final model of the patient data network was more able to classify the patients with and without CAD correctly than misclassifying them.

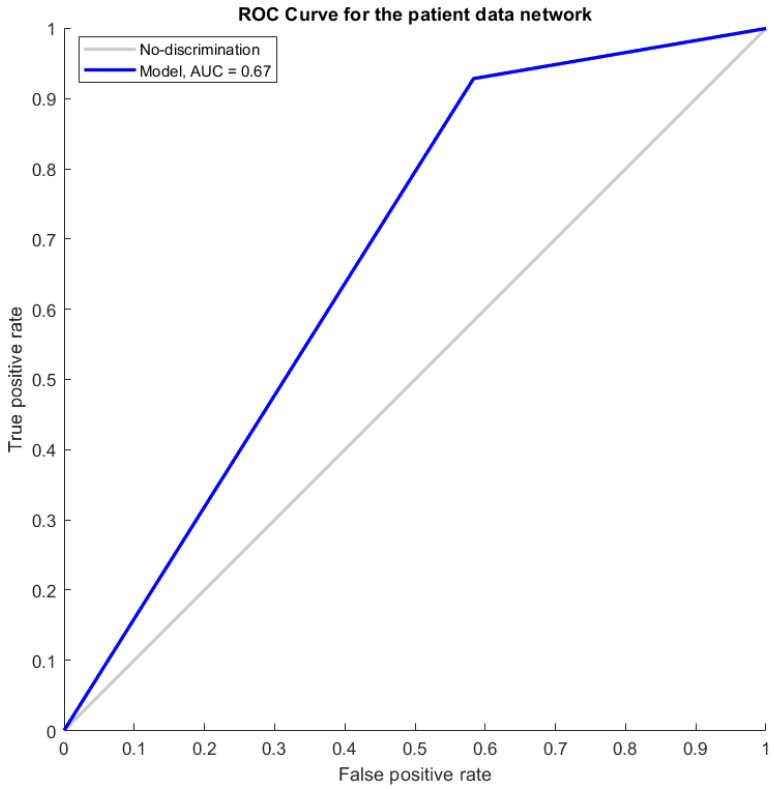


Figure 4.10. ROC for the test data set of the final model of the patient data network. The y-axis represents the true positive rate and the x-axis represents the false positive rate. The blue line illustrates the ROC curve for the test data set and the grey line indicates where the true positive rate and false positive rate are equal.

4.3 CAD network

When combining the CNN-RNN and the patient data network, it became the full CAD network. In figure 4.11 is the loss curves of the training- and validation loss for the training of the final architecture of the CAD network presented.

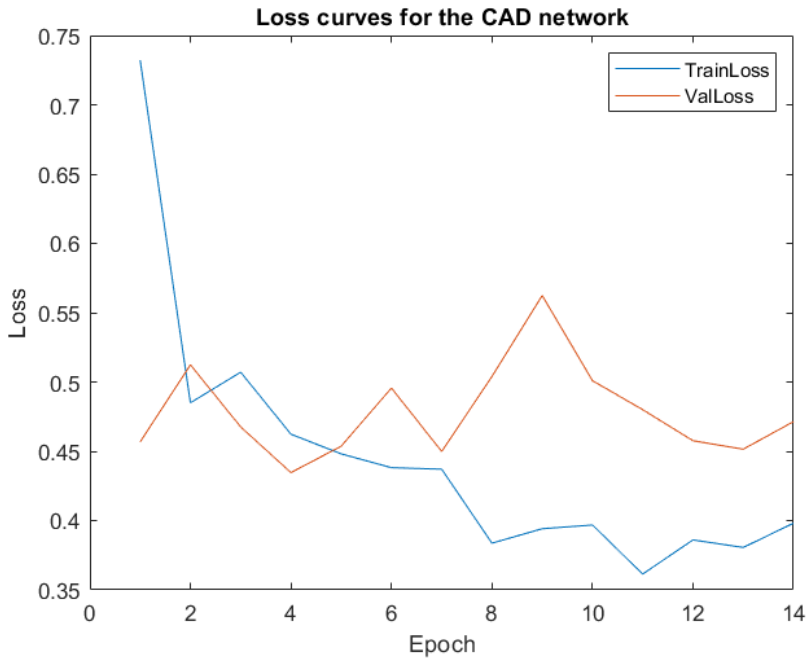


Figure 4.11. Loss curves for the training- and validation loss after training the final model of the CAD network. The y-axis represents the loss and the x-axis represents the number of epochs. The orange line illustrates the validation loss and the blue line illustrates the training loss.

The loss curves for the CAD network illustrated a descending tendency for the training loss, whereas an overall flat tendency in the validation loss was expressed. This was also seen in the loss curves for the CNN-RNN in figure 4.1. However, the loss curves for the CAD network differed from the patient data network loss curves in figure 4.6 by having a flat tendency in the validation loss. From epoch 8 to the last epoch, the difference between training- and validation loss increased in the range of 0.08 to 0.1, indicating a slightly overfitting to the training data. Thus, the CAD network overfitted less to the training data compared to the CNN-RNN. The final model of the CAD network was obtained after the last epoch, where the training loss was 0.3981 and the validation loss was 0.4713.

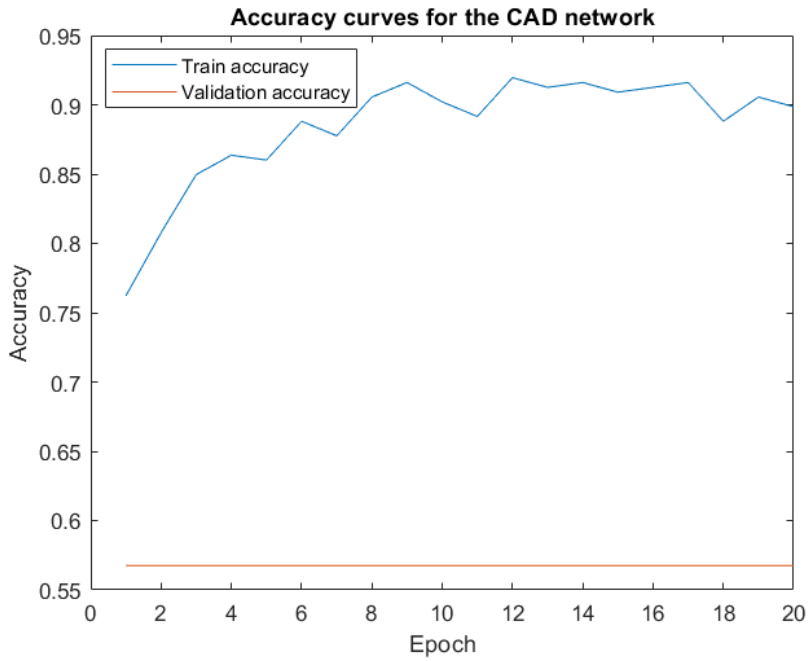


Figure 4.12. Accuracy curves for the training- and validation accuracy after training the final model of the CNN-RNN. The y-axis represents the accuracy and the x-axis represents the number of epochs. The validation accuracy curve is illustrated in orange and the training accuracy curve is marked in blue.

Like the accuracy curves of the CNN-RNN in figure 4.2, the accuracy curves of the CAD network in figure 4.12 illustrated an ascending training accuracy, but a constant validation accuracy. Thereby, the CAD network also improved the performance on the training data, however, the performance on the validation data did not change.

In figure 4.13 the confusion matrix with a threshold of 0.5 for the final model of the CAD network is presented. The CAD network classified 5 CCTA volumes containing CAD and 16 CCTA volumes without CAD correctly whereas it misclassified 9 CCTA volumes with CAD and 8 CCTA volumes without CAD.

		Predicted labels	
		Containing CAD	**Not containing CAD**
Targeted labels	**Containing CAD**	5	9
	Not containing CAD	8	16

Figure 4.13. Confusion matrix with a threshold of 0.5 for the performance of the CAD network on the test data set. The test set contained 38 patients, where 14 patients had CAD and 24 patients did not.

As mentioned, the CCTA volumes with CAD contained a different amount of stenoses. In

table 4.6, the distribution of how many CCTA volumes with one, two, and three stenoses were correctly classified or misclassified by the CAD network, CNN-RNN, and the patient data network was illustrated. The 14 CCTA volumes in the test data set contained either one, two, or three stenoses in the coronary arteries. From table 4.6 it can be observed that the CAD network and CNN-RNN were more equaled in their classifications of the amount of stenosis in the CCTA volumes with CAD. Moreover, a comparison of the CAD network with the patient data network showed that the patient data network was better at correctly classifying CCTA volumes containing one, two, and three stenoses.

	CAD network		**CNN-RNN**		**Patient data network**	
	Correctly classified	**Mis-classified**	**Correctly classified**	**Mis-classified**	**Correctly classified**	**Mis-classified**
1 stenosis	2	4	4	3	7	0
2 stenoses	3	2	3	1	4	0
3 stenoses	0	3	0	3	2	1

Table 4.6. The distribution of how many CCTA volumes with one, two, and three stenoses was correctly classified and misclassified by the CAD network, CNN-RNN, and the patient data network.

In figure 4.14, a comparison of the common predicted labels between the CAD network and the CNN-RNN as well as between the CAD network and the patient data network is illustrated. When comparing if the CNN-RNN and the CAD network had classified or misclassified the same CCTA volumes, it was observed in figure 4.14 that the CNN-RNN and CAD network was more likely to classify the same CCTA volumes correct or not. Whereas the CAD network and the patient data network were not as likely to classify the same CCTA volumes correct or not.

Predictions of the CAD network compared to the CNN- RNN

		Predicted labels	
		Containing CAD	Not containing CAD
Targeted labels	Containing CAD	5	5
	Not containing CAD	7	14

Predictions of the CAD network compared to the patient data network

		Predicted labels	
		Containing CAD	Not containing CAD
Targeted labels	Containing CAD	4	1
	Not containing CAD	3	5

Figure 4.14. Comparison of the common predicted labels between the CAD network and the CNN-RNN and between the CAD network and the patient data network. The **Predicted labels** account for the labels predicted by the networks and the **Targeted labels** is the true labels of the CCTA volumes.

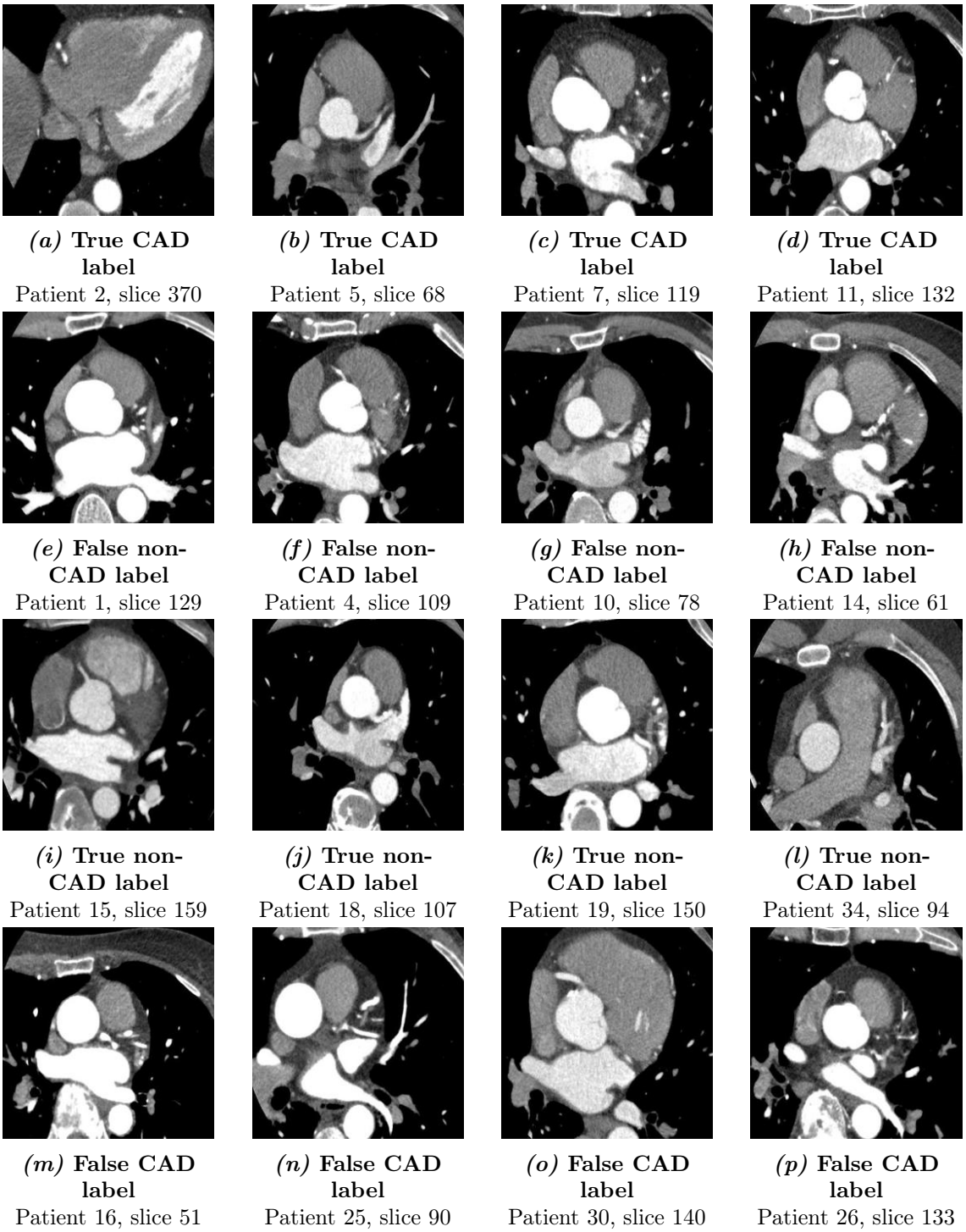
The F1 score for the prediction of CAD, non-CAD, and the average F1 score for the final architecture of the CNN-RNN, the patient data network, and the CAD network on the test data set are presented in table 4.7 along with the sensitivity, specificity, accuracy, and AUROC.

CNN-RNN						
F1 score CAD	**F1 score Non-CAD**	**Average F1 score**	**Sensitivity**	**Specificity**	**Accuracy**	**AUROC**
0.45	0.62	0.54	0.5	0.58	0.55	0.54
Patient data network						
F1 score CAD	**F1 score Non-CAD**	**Average F1 score**	**Sensitivity**	**Specitivity**	**Accuracy**	**AUROC**
0.63	0.57	0.60	0.92	0.42	0.61	0.67
CAD network						
F1 score CAD	**F1 score Non-CAD**	**Average F1 score**	**Sensitivity**	**Specitivity**	**Accuracy**	**AUROC**
0.37	0.65	0.51	0.36	0.67	0.55	0.51

Table 4.7. The F1 score for the labels CAD and non-CAD, and the average F1 score for CNN-RNN, the patient data network, and the CAD network along with the sensitivity, specificity, accuracy, and AUROC.

The average F1 score of 0.51 for the CAD network showed that the CAD network could classify half of the CCTA volumes correctly. Just like the results for the CNN-RNN indicated the F1 score for CAD and non-CAD of 0.37 and 0.65, respectively, that the CAD network was more likely to classify CCTA volumes without CAD correctly. The sensitivity of 0.36 for the CAD network showed that the network was not able to correctly classify the CCTA volumes with CAD. However, the CAD network's specificity of 0.67 illustrated that the network was better at identifying the CCTA volumes without CAD. Comparing the accuracy of the CNN-RNN with the CAD network, illustrated that the CAD network did not differ in performance from the CNN-RNN when the patient data network was combined with the CNN-RNN in the CAD network.

Since the performance of the CAD network showed an almost random classification of the CCTA volumes, examples of slices from the CCTA volumes from the test set and whether or not they were classified correctly are illustrated in figure 4.15.



(a) **True CAD label**
Patient 2, slice 370

(b) **True CAD label**
Patient 5, slice 68

(c) **True CAD label**
Patient 7, slice 119

(d) **True CAD label**
Patient 11, slice 132

(e) **False non-CAD label**
Patient 1, slice 129

(f) **False non-CAD label**
Patient 4, slice 109

(g) **False non-CAD label**
Patient 10, slice 78

(h) **False non-CAD label**
Patient 14, slice 61

(i) **True non-CAD label**
Patient 15, slice 159

(j) **True non-CAD label**
Patient 18, slice 107

(k) **True non-CAD label**
Patient 19, slice 150

(l) **True non-CAD label**
Patient 34, slice 94

(m) **False CAD label**
Patient 16, slice 51

(n) **False CAD label**
Patient 25, slice 90

(o) **False CAD label**
Patient 30, slice 140

(p) **False CAD label**
Patient 26, slice 133

Figure 4.15. CCTA slices from 16 patients from the test data set. The labels of the slices is classified by the CAD network. The rows represent (*a-d*) the true positive labels (**True CAD label**), (*e-h*) the false negative labels (**False non-CAD label**), (*i-l*) the true negative labels (**True non-CAD label**), and (*m-p*) the false positive labels (**False CAD label**).

The classification of the CCTA volumes by the CAD network was similar to the classification of the CNN-RNN, thus, some of the patients recur between figure 4.4 and 4.15. Although new CCTA slices are present in 4.15, the same tendencies amongst the CCTA slices were present. Some of the CCTA slices were slightly blurry and the contrast

between the different tissues varied a bit, as seen in *a*, *i*, *l*, and *o*. Moreover, the amount of surrounding tissue varied, e.g *l* contained significantly more surrounding tissue than *a*.

To show the ability of the final model of the CAD network to correctly classify if CCTA volumes contained CAD or not was the ROC in figure 4.16 created.

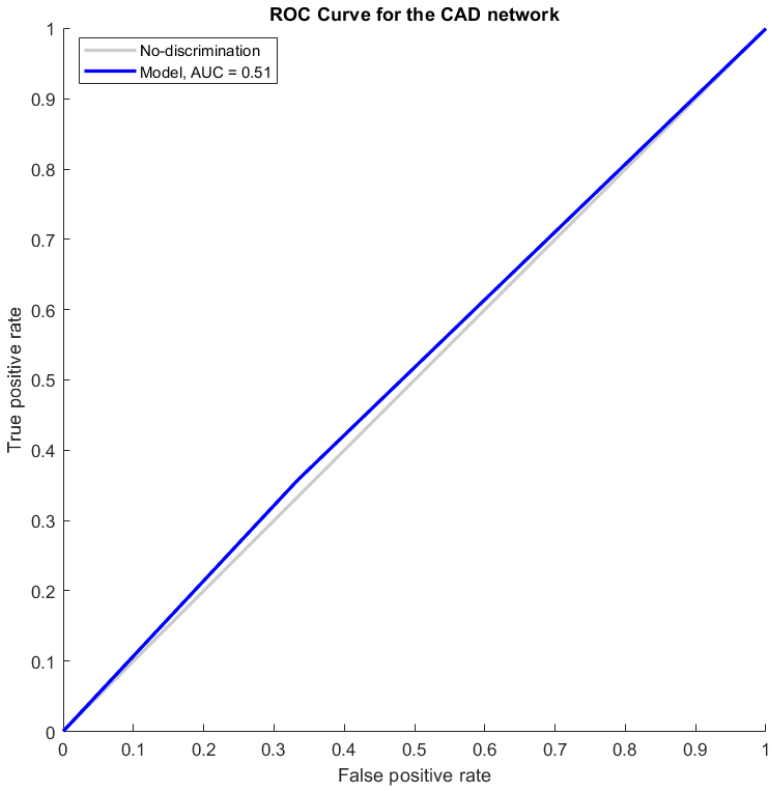


Figure 4.16. ROC for the test data set of the final model of the CAD network. The y-axis represents the true positive rate and the x-axis represents the false positive rate. The blue line illustrates the ROC curve for the test data set and the grey line indicates where the true positive rate and false positive rate are equal.

The AUROC on 0.51 indicated that the final model of the CAD network made random classifications. The ROC illustrated that the model curve differed slightly from the line of no-discrimination as it was moving up towards the upper left corner of the graph. The ROC in figure 4.5 for the CNN-RNN was slightly better than ROC for the CAD network as the performance of the model differed more from the line of no-discrimination. Thus, both networks indicated that they performed random classifications. The performance of the patient data network differed from the CAD network as it was more able to classify the CCTA volumes correctly, as illustrated by the AUROC presented in table 4.7.

Discussion 5

CCTA is often used as a diagnostic tool for identifying patients with CAD. The method has high sensitivity and a negative predictive value of 70 - 80%, but the specificity varies between 48 - 78%. The low specificity causes 22 - 52% of the patients without CAD to undergo a CAG, which is the invasive golden standard for diagnosing CAD. Moreover, the manual review of the CCTA volumes is prone to inter-observer variability. To overcome the aforementioned problems related to diagnosing CAD with CCTA, deep learning could be implemented in the workflow of the clinicians. The present study investigated the use of CCTA volumes and the corresponding patient data to identify patients with CAD by using a CAD network consisting of two neural networks based data processing components: a CNN-RNN combined with a patient data network. The CNN-RNN was based on the architecture used by Zreik et al. [2018b], but it differed by using a 2D CNN and three recurrent layers with 128 GRUs in each. The patient data network was inspired by Spasov et al. [2018] but dropout was added between the two first dense layers. The full CAD network obtained an AUROC of the ROC of 0.51 and an F1 score of 0.51, which indicated that the CAD network made random classifications. The CNN-RNN alone obtained an AUROC of the ROC of 0.54 and an F1 score of 0.54, indicating that the CNN-RNN classified half of the CCTA volumes containing CAD or non-CAD correctly. Finally, the patient data network obtained an F1 score and AUROC of the ROC of 0.6 and 0.67, respectively, which illustrated that the patient data network was better at predicting patients with CAD or non-CAD, correctly.

From the obtained results from the CAD network, a flat tendency in the validation loss was observed. The flat tendency in validation loss was also observed in the loss curves obtained through hyper-parameter optimization. A flat validation loss could result from missing variance among the slices from each CCTA volume. The suspected insignificant variance in the CCTA slices was further indicated in figure 4.4 and 4.15, showing representative CCTA slices from the test data set. The CCTA slice might have shared the same features despite containing CAD or not, which prevented the CAD network from learning the features that varied between the CAD and non-CAD CCTA volumes. If the variance amongst the slices from the different CCTA volumes was insignificant, the possibility of the CAD network detecting any differentiating features between the CAD and non-CAD CCTA volumes would decrease. To manipulate the variance in the CCTA slices, augmentation could have been applied to the CCTA slices, generating input CCTA volumes to the CAD network varying from the original CCTA volumes. Augmentation strategies could be rotation, scaling, flipping, changes in brightness, or random shift. When applying augmentation, the augmented CCTA would have to imitate the original CCTA slices, otherwise, the augmented CCTA slices would not represent reality. An augmentation strategy for the

CCTA slices could be rotation. Thereby, the information in the CCTA slice would be the same but at different locations forcing the CAD network to learn the same features at different locations.

When comparing the results with the study by Zreik et al. [2018b], they gained a higher F1 score of 0.75 for detecting significant stenosis, indicating that the RCNN was better at classifying CAD. The present study utilized a 2D CNN instead of a 3D CNN like the study by Zreik et al. [2018b], and the RNN contained an extra recurrent layer with more hidden units, thus, three recurrent layers with 128 GRUs. Using a 2D CNN instead of a 3D CNN could result in missing information from the CCTA volumes. The 2D CNN modeled spatial information in the CCTA volumes whereas a 3D CNN would also have the ability to model temporal dependencies between the slices in each CCTA volume. Thus, the present study did not model the temporal dependencies in the CCTA slice through the CNN as the study by Zreik et al. [2018b]. However, the temporal information from the CCTA volumes was instead learned in the RNN, thus, the CAD network did not miss the temporal information in the CCTA volumes. To conclude if a 3D CNN performed better than a 2D CNN in the CAD network, a CAD network with a 3D CNN should have been tested. Due to the computational limitation of the used GPU, a 3D CNN was not used in the present study. Another reason for the different performance between the present study and Zreik et al. [2018b] was the use of data. Both studies used CCTA volumes, however, Zreik et al. [2018b] differed by having extracted the centerlines of the coronary arteries from every CCTA slice to reconstruct a straightened MPR of every coronary artery in the CCTA volumes. Thus, the input to RCNN was MPR images of arteries whereas the input to the CAD network was a CCTA volume containing cross-sectional slices of the entire heart. The information in the input to the CAD network contained more noise than the MPR images as the MPR images only contained the region of interest, the coronary arteries. The use of the MPR images of the coronary arteries could have resulted in a better performance of the CAD network as the stenosis could be easier to identify in the MPR image than in CCTA volumes of the entire heart.

The CNN-RNN and patient data network were trained separately to see how each network influenced the CAD network. A comparison of the loss curves from the CNN-RNN with the loss curves of the CAD network showed the same trend in the descending training losses but a flat or slightly descending tendency for the validation losses. However, the patient data network alone had converging loss curves for both the training- and validation. From the comparison of how many common predictions the CAD network and the CNN-RNN and the CAD network and the patient data network had, respectively, in figure 4.14 in section 4.3, the CAD network and the CNN-RNN had more agreeing predictions than the CAD network and the patient data network. Thereby, the CNN-RNN may have had a bigger influence on the predictions of the final model than the patient data network. However, the performance of the patient data network was better than the CNN-RNN and the CAD network. This showed conflicting learning between the patient data network and the CNN-RNN in the CAD network. The patient data network classified most of the CCTA volumes as containing CAD, whereas the CNN-RNN classified more non-CAD CCTA volumes than with CAD. Thereby, the data processing components of the CAD network made unequal classifications, which could have contributed to the almost random classification of the CAD network. Moreover, the better performance of the patient data

network compared to the CNN-RNN could be an indication of the predictive value of the patient data, in particular the risk factors, when identifying CCTA volumes with CAD.

Before the CCTA volumes were given as input to the CNN-RNN, they were pre-processed according to section 3.2.3. The first pre-processing method consisted of cropping each slice in the CCTA volumes. The cropping of all slices in every CCTA volume was performed on behalf of a single slice from one CCTA volume. Thus, the uncertainty of whether or not the cropped CCTA slices contained the entire information of the coronary arteries was present due to the different sizes of the heart of the patients. To accommodate the possibility of one or more coronary arteries being cut off during the cropping, a region beyond the heart was included. However, the position of the heart might differ from patient to patient, thus, when the cropping was executed at the same position every time, the risk of coronary arteries being cut off was present. As the features of the coronary arteries were essential to identify CAD in the CCTA volumes, the cut off of the coronary arteries could have resulted in the lack of crucial information for predicting the correct label of the CCTA volumes. Although the risk of excluding important information from the data was present, the method of cropping the slices from the CCTA volumes was time-saving and effective to create a region of interest of the heart. To ensure all of the necessary information about the heart and the coronary arteries, other approaches for a region of interest could have been used, like segmentation of the heart region.

Second, the CCTA slices were resized to a spatial resolution of 256x256 to limit the computational burden. When resizing the CCTA slices the important information of the coronary arteries became smaller, thus, the significance of the features for the coronary arteries might have been reduced. However, the spatial resolution of 256x256 was the biggest resolution allowed due to the computational limitation of the GPU. Additionally, the cropping of the CCTA slices removed unnecessary information before the the resizing, thereby, the information of the heart and the coronary arteries was more present than if the whole CCTA slices had been resized.

5.1 Limitations

A limitation of this study was the computational limit of the GPU. Due to the computational limitation, a batch size of one was used and submission of more layers and filters to the CNN was not possible. An increased batch size could have increased the ability of the CAD network to generalize to unseen CCTA volumes. However, it was not possible to optimize the batch size. The batch size, which indicated the best performance of the CAD network, should have been found through hyper-parameter optimization. Moreover, the addition of more layers and filters to the CNN could have improved the expressive ability.

The data set of the present study contained more non-CAD CCTA volumes than CAD CCTA volumes. This was also illustrated in the F1 score for CAD and non-CAD for the CAD network. Although the distribution between CAD and non-CAD was 41% and 59%, the learning of non-CAD features was biased. To increase the learning of CAD features, the distribution could have been in favor of the amount of CCTA volumes containing CAD. Since the features of CAD were presumed to be few based on the representation of

the coronary arteries in the CCTA slices and learning of these features was preferred, it might have increased the ability of the network to predict the correct labels. To increase the representation of the CAD features, a data set consisting of an overweight of CCTA volumes containing CAD could enhance the leaning of features associated with CAD. Although, manipulating the distribution between CAD and non-CAD CCTA volumes would not be representative of the population in the data set, the CAD network might learn the difference between CAD and non-CAD CCTA volumes.

Moreover, the patient data consisted of clinical assessment of the symptoms like angina but the use of symptoms as patient data was not possible, since it was not defined for all the patients. Other patient data such as which segments of the coronary arteries were present for the patients with hemodynamically significant CAD, but none of the authors of the present study were experience enough to point out which CCTA slices contained the given segments nor to identify CCTA slices containing significant stenosis. Although the authors were not trained cardiac clinicians, the patient data did contain information about where the stenoses were in the CCTA volumes with CAD and how many were detected. The exact location of the stenoses could not be determined by the authors, but the number of stenoses for each CCTA volume could have been a part of the patient data fed to the CAD network to improve the possibility of the network to predict the correct labels. Additionally, information of which slices contained stenosis could have been used to label each CCTA slice as containing CAD or non-CAD instead of the CCTA volumes. When labeling the CCTA slices as containing CAD or not, the CAD network might learn to map features in each CCTA slices to either belonging to CAD or non-CAD, thus, the location of stenosis might be detected.

As the results from the CAD network indicated randomly classification of whether or not CCTA volumes contained CAD or not, improvement of the performance of the CAD network had to be made before implemented in clinical practise. To improve the performance of the CAD network, the approach of class activation maps proposed by Podgorsak et al. [2020] could have been implemented to identify which features in the CCTA slices were weighted as important. Thereby, it would be visualized if the regions containing the coronary arteries were weighted by the CAD network. If the features of the coronary arteries were not weighted important by the CAD network, it might be necessary to implement anatomical guidance to ensure that features of the coronary arteries were weighted.

Conclusion 6

This study aimed to identify patients with CAD based on CCTA volumes and corresponding patient data using deep learning. From the results, it can be concluded that the CAD network made random classifications according to the AUROC of the ROC of 0.51 and an F1 score of 0.51. The CAD network consisted of two neural networks; the CNN-RNN and the patient data network. The performance of the CNN-RNN and the patient data network obtained an F1 score of 0.45 and 0.63, respectively. This indicated that the CNN-RNN classified half of the CCTA volumes containing CAD or non-CAD correctly and that the patient data network was better at predicting patients with CAD or non-CAD, correctly. For the CAD network to be used in clinical practice, the performance of the CNN-RNN and the patient data work had to be improved.

Bibliography

Achenbach, **2006**. Stephan Achenbach. *Computed tomography coronary angiography.* Journal of the American College of Cardiology, 48(10), 1919–1928, 2006.

Bonaccorso, **2018**. Giuseppe Bonaccorso. *Python: Advanced Guide to Artificial Intelligence.* Packt Publishing, 1st edition edition, 2018. ISBN 1-78995-721-4.

Chen et al., **2020**. Meng Chen, Ximing Wang, Guangyu Hao, Xujie Cheng, Chune Ma, Ning Guo, Su Hu, Qing Tao, Feirong Yao and Chunhong Hu. *Diagnostic performance of deep learning-based vascular extraction and stenosis detection technique for coronary artery disease.* The British journal of radiology, 93(1113), 20191028, 2020.

Cheung et al., **2021**. Wing Keung Cheung, Robert Bell, Arjun Nair, Leon J Menezes, Riyaz Patel, Simon Wan, Kacy Chou, Jiahang Chen, Ryo Torii, Rhodri H Davies et al. *A computationally efficient approach to segmentation of the aorta and coronary arteries using deep learning.* Ieee Access, 9, 108873–108888, 2021.

Cho et al., **2014**. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk and Yoshua Bengio. *Learning phrase representations using RNN encoder-decoder for statistical machine translation.* arXiv preprint arXiv:1406.1078, 2014.

Chollet et al., **2018**. Francois Chollet et al. *Deep learning with Python*, volume 361. Manning New York, 2018.

Chung et al., **2014**. Junyoung Chung, Caglar Gulcehre, KyungHyun Cho and Yoshua Bengio. *Empirical evaluation of gated recurrent neural networks on sequence modeling.* arXiv preprint arXiv:1412.3555, 2014.

Cong et al., **2019**. Chao Cong, Yoko Kato, Henrique Doria Vasconcellos, Joao Lima and Bharath Venkatesh. *Automated stenosis detection and classification in x-ray angiography using deep neural network.* pages 1301–1308, 2019.

Dance et al., **2014**. DR Dance, S Christofides, ADA Maidment, ID McLean and KH Ng. *Diagnostic radiology physics: A handbook for teachers and students. Endorsed by: American Association of Physicists in Medicine, Asia-Oceania Federation of Organizations for Medical Physics, European Federation of Organisations for Medical Physics.* 2014.

Fleming, **2022**. Sean Fleming. *These are the top 10 global causes of death - but two diseases are in decline*, 2022. URL `https://www.weforum.org/agenda/2021/02/top-10-global-causes-death/`.

Guo, **2013**. Jiang Guo. *Backpropagation through time.* Unpubl. ms., Harbin Institute of Technology, 40, 1–6, 2013.

Hajar, **2017**. Rachel Hajar. *Risk factors for coronary artery disease: historical perspectives.* Heart views: the official journal of the Gulf Heart Association, 18(3), 109, 2017.

Hall and Hall, **2011**. John E Hall and Michael E Hall. *Guyton and Hall textbook of medical physiology.* Elsevier Health Sciences, 2011.

Hampe et al., **2019**. Nils Hampe, Jelmer M Wolterink, Sanne GM Van Velzen, Tim Leiner and Ivana Išgum. *Machine learning for assessment of coronary artery disease in cardiac CT: a survey.* Frontiers in cardiovascular medicine, 6, 172, 2019.

Han et al., **2020**. Dan Han, Jiayi Liu, Zhonghua Sun, Yu Cui, Yi He and Zhenghan Yang. *Deep learning analysis in coronary computed tomographic angiography imaging for the assessment of patients with coronary artery stenosis.* Computer Methods and Programs in Biomedicine, 196, 105651, 2020.

Hochreiter, **1998**. Sepp Hochreiter. *The vanishing gradient problem during learning recurrent neural nets and problem solutions.* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02), 107–116, 1998.

Huang and Yin, **2021**. Caiyun Huang and Changhua Yin. *A coronary artery CTA segmentation approach based on deep learning.* Journal of X-ray science and technology, (Preprint), 1–14, 2021.

Ioffe and Szegedy, **2015**. Sergey Ioffe and Christian Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* pages 448–456, 2015.

Kanitsar et al., **2002**. Armin Kanitsar, Dominik Fleischmann, Rainer Wegenkittl, Petr Felkel and Eduard Groller. *CPR-curved planar reformation.* IEEE, 2002.

Khan et al., **2018**. Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, Mohammed Bennamoun, Gerard Medioni and Sven Dickinson. *A Guide to Convolutional Neural Networks for Computer Vision.* Synthesis Lectures on Computer Vision. Morgan Claypool Publishers, San Rafael, 1 edition, 2018. ISBN 1681730227.

Kingma and Ba, **2014**. Diederik P Kingma and Jimmy Ba. *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.

Ko et al., **2012**. Brian S Ko, James D Cameron, Michael Leung, Ian T Meredith, Darryl P Leong, Paul R Antonis, Marcus Crossett, John Troupis, Richard Harper, Yuvaraj Malaiapan et al. *Combined CT coronary angiography and stress myocardial perfusion imaging for hemodynamically significant stenoses in patients with suspected coronary artery disease: a comparison with fractional flow reserve.* JACC: Cardiovascular Imaging, 5(11), 1097–1111, 2012.

Kristanto et al., **2013**. Wisnumurti Kristanto, Peter MA van Ooijen, Marijke C Jansen-van der Weide, Rozemarijn Vliegenthart and Matthijs Oudkerk. *A meta analysis and hierarchical classification of HU-based atherosclerotic plaque characterization criteria.* PLoS One, 8(9), e73460, 2013.

Kumamaru et al., **2020**. Kanako K Kumamaru, Shinichiro Fujimoto, Yujiro Otsuka, Tomohiro Kawasaki, Yuko Kawaguchi, Etsuro Kato, Kazuhisa Takamura, Chihiro Aoshima, Yuki Kamo, Yosuke Kogure et al. *Diagnostic accuracy of 3D deep-learning-based fully automated estimation of patient-level minimum fractional flow reserve from coronary computed tomography angiography.* European Heart Journal-Cardiovascular Imaging, 21(4), 437–445, 2020.

Lawrence, **2003**. Alissa Lawrence. *Fundamentals of anatomy and physiology, ninth edition.* 2003. ISBN 9780321709332.

Long and Zeng, **2022**. Liangqu Long and Xiangming Zeng. *Beginning Deep Learning with TensorFlow: Work with Keras, MNIST Data Sets, and Advanced Neural Networks.* Apress L. P, 2022. ISBN 9781484279144.

Ma and Shieh, **2006**. Hongbao Ma and Kuan-Jiunn Shieh. *Cholesterol and human health.* The Journal of American Science, 2(1), 46–50, 2006.

Martinez and Stiefelhagen, **2018**. Manuel Martinez and Rainer Stiefelhagen. *Taming the cross entropy loss.* pages 628–637, 2018.

Messerli et al., **2007**. Franz H Messerli, Bryan Williams and Eberhard Ritz. *Essential hypertension.* The Lancet, 370(9587), 591–603, 2007.

Min et al., **2012**. James K Min, Jonathon Leipsic, Michael J Pencina, Daniel S Berman, Bon-Kwon Koo, Carlos Van Mieghem, Andrejs Erglis, Fay Y Lin, Allison M Dunning, Patricia Apruzzese et al. *Diagnostic accuracy of fractional flow reserve from anatomic CT angiography.* Jama, 308(12), 1237–1245, 2012.

Mirunalini et al., **2019**. Palaniappan Mirunalini, Chandrabose Aravindan, A Thamizh Nambi, S Poorvaja and V Pooja Priya. *Segmentation of Coronary Arteries from CTA axial slices using Deep Learning techniques.* pages 2074–2080, 2019.

Moon et al., **2021**. Jong Hak Moon, Won Chul Cha, Myung Jin Chung, Kyu-Sung Lee, Baek Hwan Cho, Jin Ho Choi et al. *Automatic stenosis recognition from coronary angiography using convolutional neural networks.* Computer methods and programs in biomedicine, 198, 105819, 2021.

Mori et al., **2018**. Hiroyoshi Mori, Sho Torii, Matthew Kutyna, Atsushi Sakamoto, Aloke V Finn and Renu Virmani. *Coronary artery calcification and its progression: what does it really mean?* JACC: Cardiovascular Imaging, 11(1), 127–142, 2018.

Nissen et al., **2016**. Louise Nissen, Simon Winther, Christin Isaksen, June Anita Ejlersen, Lau Brix, Grazina Urbonaviciene, Lars Frost, Lene Helleskov Madsen, Lars Lyhne Knudsen, Samuel Emil Schmidt et al. *Danish study of Non-Invasive testing in Coronary Artery Disease (Dan-NICAD): study protocol for a randomised controlled trial.* Trials, 17(1), 1–11, 2016.

Nørgaard et al., **2014**. Bjarne L Nørgaard, Jonathon Leipsic, Sara Gaur, Sujith Seneviratne, Brian S Ko, Hiroshi Ito, Jesper M Jensen, Laura Mauri, Bernard De Bruyne, Hiram Bezerra et al. *Diagnostic performance of noninvasive fractional flow reserve derived from coronary computed tomography angiography in suspected coronary*

artery disease: the NXT trial (Analysis of Coronary Blood Flow Using CT Angiography: Next Steps). Journal of the American College of Cardiology, 63(12), 1145–1155, 2014.

Novaković et al., **2017**. Jasmina Dj Novaković, Alempije Veljović, Siniša S Ilić, Željko Papić and Tomović Milica. *Evaluation of classification models in machine learning.* Theory and Applications of Mathematics & Computer Science, 7(1), 39–46, 2017.

Nwankpa et al., **2018**. Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan and Stephen Marshall. *Activation functions: Comparison of trends in practice and research for deep learning.* arXiv preprint arXiv:1811.03378, 2018.

Pillow, **2022**. Pillow. *Pillow*, 2022. URL `https://pillow.readthedocs.io/en/stable/`.

Podgorsak et al., **2020**. Alexander R Podgorsak, Kelsey N Sommer, Abhinay Reddy, Vijay Iyer, Michael F Wilson, Frank J Rybicki, Dimitrios Mitsouras, Umesh Sharma, Shinchiro Fujimoto, Kanako K Kumamaru et al. *Initial evaluation of a convolutional neural network used for noninvasive assessment of coronary artery disease severity from coronary computed tomography angiography data.* Medical Physics, 47(9), 3996–4004, 2020.

Pugliese et al., **2009**. Francesca Pugliese, MG Myriam Hunink, Katarzyna Gruszczynska, Filippo Alberghina, Roberto Malagó, Niels van Pelt, Nico R Mollet, Filippo Cademartiri, Annick C Weustink, Willem B Meijboom et al. *Learning curve for coronary CT angiography: what constitutes sufficient training?* Radiology, 251(2), 359–368, 2009.

Saur et al., **2010**. Stefan C Saur, Hatem Alkadhi, Paul Stolzmann, Stephan Baumüller, Sebastian Leschka, Hans Scheffel, Lotus Desbiolles, Thomas J Fuchs, Gábor Székely and Philippe C Cattin. *Effect of reader experience on variability, evaluation time and accuracy of coronary plaque detection with computed tomography coronary angiography.* European radiology, 20(7), 1599–1606, 2010.

Sharma, **2022**. Rahul Sharma. *Coronary Angiography In Jaipur*, 2022. URL `https://cardiologistjaipur.com/Coronary-Angiography-in-jaipur.html`.

Sharma et al., **2017**. Sagar Sharma, Simone Sharma and Anidhya Athaiya. *Activation functions in neural networks.* towards data science, 6(12), 310–316, 2017.

Shin et al., **2016**. Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura and Ronald M Summers. *Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning.* IEEE transactions on medical imaging, 35(5), 1285–1298, 2016.

Spasov et al., **2018**. Simeon E Spasov, Luca Passamonti, Andrea Duggento, Pietro Lio and Nicola Toschi. *A multi-modal convolutional neural network framework for the prediction of Alzheimer's disease.* pages 1271–1274, 2018.

Stathakis, **2009**. D Stathakis. *How many hidden layers and nodes?* International Journal of Remote Sensing, 30(8), 2133–2147, 2009.

Sun, **2013**. Zhonghua Sun. *Coronary Computed Tomography Angiography in Coronary Artery Disease: A Systematic Review of Image Quality, Diagnostic Accuracy and Radiation Dose - Ebook*. Nova Science Publishers, 2013. 1. edition.

TensorFlow, **2022**. TensorFlow. *TensorFlow Core v2.8.0: tf.keras.layers.TimeDistributed*, 2022. URL `https://www.tensorflow.org/api_docs/python/tf/keras/layers/TimeDistributed`.

Thanaki, **2018**. Jalaj Thanaki. *Machine learning solutions : expert techniques to tackle complex machine learning problems using Python*. Packt Publishing, Birmingham, UK, 2018. ISBN 1-78839-889-0.

van Hamersvelt et al., **2019**. Robbert W van Hamersvelt, Majd Zreik, Michiel Voskuil, Max A Viergever, Ivana Išgum and Tim Leiner. *Deep learning analysis of left ventricular myocardium in CT angiographic intermediate-degree coronary stenosis improves the diagnostic accuracy for identification of functionally significant stenosis*. European radiology, 29(5), 2350–2359, 2019.

Wang et al., **2011**. Rui Wang, Wei Yu, Yongmei Wang, Yi He, Lin Yang, Tao Bi, Jian Jiao, Qian Wang, Liquan Chi, Yang Yu et al. *Incremental value of dual-energy CT to coronary CT angiography for the detection of significant coronary stenosis: comparison with quantitative coronary angiography and single photon emission computed tomography*. The international journal of cardiovascular imaging, 27(5), 647–656, 2011.

WHO, **2000**. WHO. *Obesity: preventing and managing the global epidemic*. World Health Organization technical report series, 894, 1–253, 2000.

World Health Organization, **2021**. World Health Organization. *Cardiovascular diseases (CVDs)*, 2021. URL `https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)`.

Zhang et al., **2008**. Shaoxiong Zhang, David C Levin, Ethan J Halpern, David Fischman, Michael Savage and Paul Walinsky. *Accuracy of MDCT in assessing the degree of stenosis caused by calcified coronary artery plaques*. American Journal of Roentgenology, 191(6), 1676–1683, 2008.

Zhao et al., **2021**. Chen Zhao, Aviral Vij, Saurabh Malhotra, Jinshan Tang, Haipeng Tang, Drew Pienta, Zhihui Xu and Weihua Zhou. *Automatic extraction and stenosis evaluation of coronary arteries in invasive coronary angiograms*. Computers in Biology and Medicine, 136, 104667, 2021.

Zreik et al., **2018a**. Majd Zreik, Nikolas Lessmann, Robbert W van Hamersvelt, Jelmer M Wolterink, Michiel Voskuil, Max A Viergever, Tim Leiner and Ivana Išgum. *Deep learning analysis of the myocardium in coronary CT angiography for identification of patients with functionally significant coronary artery stenosis*. Medical image analysis, 44, 72–85, 2018a.

Zreik et al., **2018b**. Majd Zreik, Robbert W Van Hamersvelt, Jelmer M Wolterink, Tim Leiner, Max A Viergever and Ivana Išgum. *A recurrent CNN for automatic detection and classification of coronary artery plaque and stenosis in coronary CT angiography.* IEEE transactions on medical imaging, 38(7), 1588–1598, 2018b.

Zreik et al., **2019**. Majd Zreik, Robbert W van Hamersvelt, Nadieh Khalili, Jelmer M Wolterink, Michiel Voskuil, Max A Viergever, Tim Leiner and Ivana Išgum. *Deep learning analysis of coronary arteries in cardiac CT angiography for detection of patients requiring invasive coronary angiography.* IEEE transactions on medical imaging, 39(5), 1545–1557, 2019.

Structures Literature search A

To investigate the problem domain within the scope of the present study, a structured literature search had been performed. The structured literature search investigated the diagnosis of CAD and the challenges related to the diagnosis. The first step in the process was to find keywords for the structured literature search. This was done by performing a semi-structured search. A search protocol illustrated in table A.1 had been made to document the structured literature search. The search protocol contained information regarding the problem, limitations, and inclusion and exclusion criteria for the search.

Element	Definition
Background	Coronary artery disease (CAD) is the most common type of cardiovascular disease and the most frequent cause of death worldwide. CAD occurs due to partially or completely blocked areas called stenosis in the coronary arteries causing a reduction of oxygen rich blood to the myocardium. The diagnosis of CAD is important to treat it before the heart is severely or permanently damaged. The diagnosis of CAD is based on imaging technologies that provided visualization of the coronary arteries.
Problem domain	The golden standard for diagnosing CAD is coronary angiography (CAG). However, the method is invasive and has risks of complications such as myocardial infarction due to the catheter passing plaque causing it to rupture. A non-invasive method for detecting stenosis is coronary computed tomography angiography (CCTA). This method is time-consuming as the clinicians manually have to review a large number of images for detecting stenosis.
Information source	PubMed
Search strategies	Block search Chain search
Inclusion criteria	Articles in English or Danish
Exclusion criteria	E1: Full text not available E2: References not specified E3: Studies regarding animals E4: Studies regarding calcium score E5: Studies regarding children
Strategy of selection and critical examination	1) Selection of sources based on title and abstract 2) Selection of source based on full text of the article 3) Chain search based on included literature

Table A.1. Search protocol for the structured literature search.

In the structured literature search, two search strategies were used. The first search strategy was the block search, illustrated in table A.2. The identified keywords from the semi-structured search were used in the block search. They were connected by the boolean operators "AND" and "OR", which will narrow or expand the search. From the

block search, a search string was derived and used in the medical database PubMed.

AND			
OR	Coronary artery disease	Cardiac CT	Deep learning
	Coronary artery stenosis	Computed tomography	CNN
		Coronary angiography	

Table A.2. Block search for the structured literature search.

From the initial search were 11 articles identified and 89 articles were found through the search in PubMed. Initially, the title and abstract of the articles were sorted by the inclusion and exclusion criteria and afterward the articles were sorted by full text. The structured literature search resulted in 38 articles, where 34 articles were found through block search and 4 articles were identified through chain search. The sorting process can be seen in figure A.1.

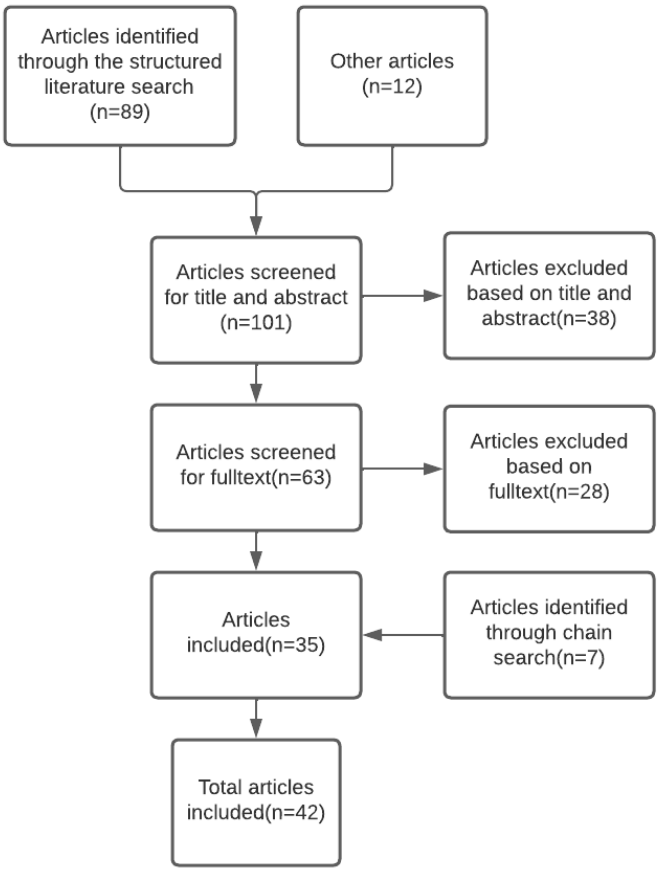


Figure A.1. The sorting process for the literature search.

Layers in the CAD network B

B.1 Convolution

A convolution is a linear operation that involves multiplication between the input CCTA slice and a two-dimensional array consisting of filter coefficients, called a match filter. The filter applied was smaller than the input CCTA slice, but had the same depth. Thereby, the filter only covered a part of the input CCTA slices, as illustrated in figure B.1. [Chollet et al., 2018] Between the filter coefficients and pixels in the covered part of the CCTA slice, an element-wise multiplication was performed. The multiplied values were then summed and divided with the number of filter coefficients, which resulted in a single value. This process continued until the filter passed every possible position in the input CCTA slice. As illustrated in figure B.1, the filter moves from the left to the right and from top to bottom. The next position of the filter was decided by a stride e.g. if the stride was one, the filter shifted one pixel for each movement. [Chollet et al., 2018]

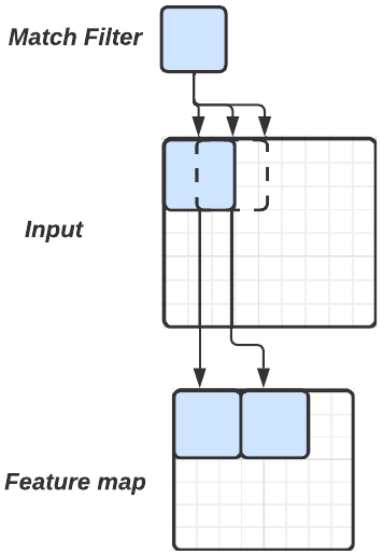


Figure B.1. Concept of the convolution operation. A match filter slides over the input CCTA slice making an element-wise multiplication resulting in a feature map. Inspired by Chollet et al. [2018].

The result of the convolution was a two-dimensional array with output values, called a feature map. [Chollet et al., 2018]

B.1.1 Zero-padding

After the convolution operation, the spatial resolution of the output feature map would be smaller than the spatial resolution of the input CCTA slice or feature map. Zero-padding is a technique to preserve the spatial resolution of an input CCTA slice or feature map. The concept of zero-padding is illustrated in figure B.2. [Chollet et al., 2018]

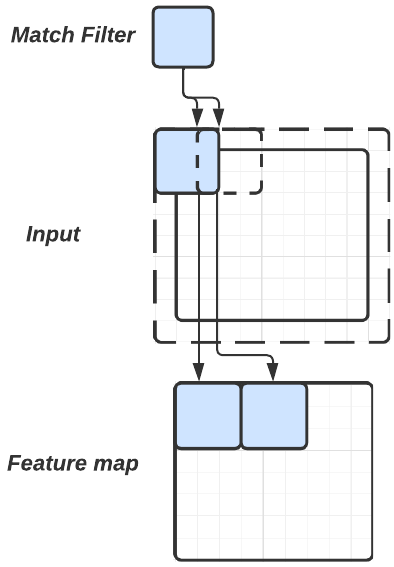


Figure B.2. Concept of the convolution operation with zero-padding. A match filter slides over the input making an element-wise multiplication resulting in a feature map. The input is padded with zeros along the edges to maintain the spatial resolution of the input. Inspired by [Chollet et al., 2018].

When using zero-padding, a border of pixels with the value of zero was added around the edges of the input feature map, as illustrated in figure B.2. [Chollet et al., 2018]

B.2 Batch normalization

The purpose of batch normalization was to reduce internal covariate shift which allowed for fast and stable training. Internal covariate shift is the change in the distribution of internal nodes due to the change in the weights and bias values. This change in distribution presented a problem because the subsequent layer needed to adapt to the new distribution, which affected the CNNs training speed. Batch normalization overcame internal covariate shift by normalizing the inputs for the subsequent layer using a fixed mean and variance. Thus, the distribution of the inputs did not change which stabilized and accelerated the training of the CNN. [Ioffe and Szegedy, 2015]

B.3 Activation functions

Activation functions are non-linear functions that computed the weighted sum of inputs and added bias. Thereby, they were indicating which part of the inputs were relevant for the CAD network to focus on. The choice of activation function depended on the position

of the activation function. Activation functions placed in the hidden layers converted linear mappings to non-linear forms. Typically, the same activation function was used in all the hidden layers. Activation functions placed in the output layer would instead create predictions. The output layer typically used a different activation function than the hidden layers and depended on the type of prediction required by the network.[Nwankpa et al., 2018]

B.3.1 Rectified Linear Unit

The Rectifies Linear Unit (ReLU) activation function performed a threshold operation on each element of the input feature map, as illustrated in figure B.3. [Nwankpa et al., 2018]

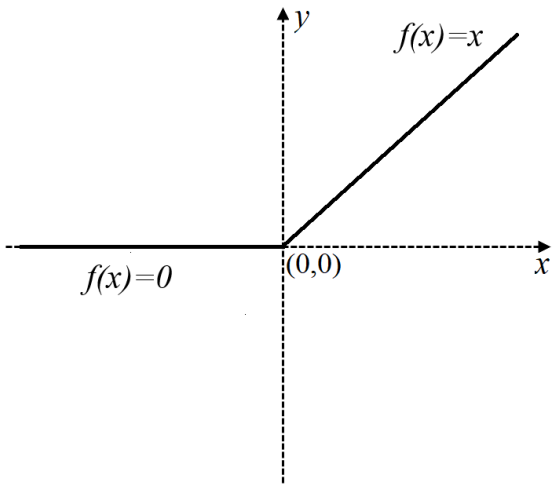


Figure B.3. The ReLU activation function. Inspired by [Nwankpa et al., 2018]

If the input element was below bias, the ReLU outputted zero whereas an input element above bias remained unchanged. ReLU with a bias of zero is given in equation B.1.

$$f(x) = \begin{cases} x_i & if \;\; x_i \geq 0 \\ 0 & if \;\; x_i < 0, \end{cases} \tag{B.1}$$

where $f(x)$ is the output of the activation function and x_i is the input element. [Nwankpa et al., 2018]

B.3.2 Exponential linear unit

The Exponential Linear Unit (ELU) activation function is a variant of the ReLU activation function, which introduced an alpha constant, defining an exponential parameter slope for input elements below the bias, as seen in figure B.4. [Sharma et al., 2017]

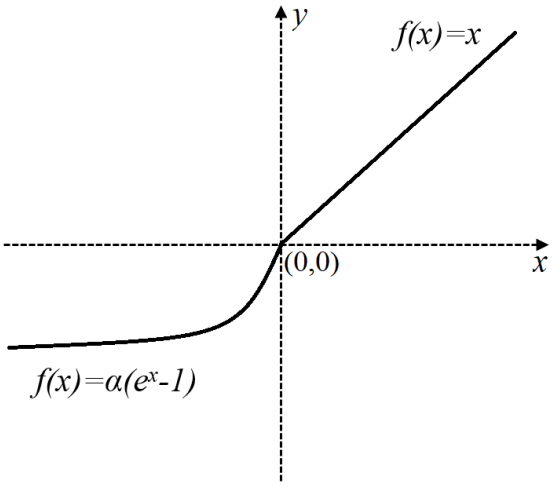


Figure B.4. The ELU activation function. Inspired by [Sharma et al., 2017]

The mathematical expression of ELU with a bias of zero is given ín equation B.2 [Sharma et al., 2017].

$$f(x_i) = \begin{cases} x_i & if \;\; x_i \geq 0 \\ \alpha(e^{x_i} - 1) & if \;\; x_i < 0, \end{cases} \tag{B.2}$$

B.3.3 Sigmoid activation function

The sigmoid activation function is a logistic function, that outputs values in the range 0 to 1, as shown in figure B.5. The larger input, the closer the output value was to 1 whereas the smaller the input was, the closer the output was to 0. [Nwankpa et al., 2018]

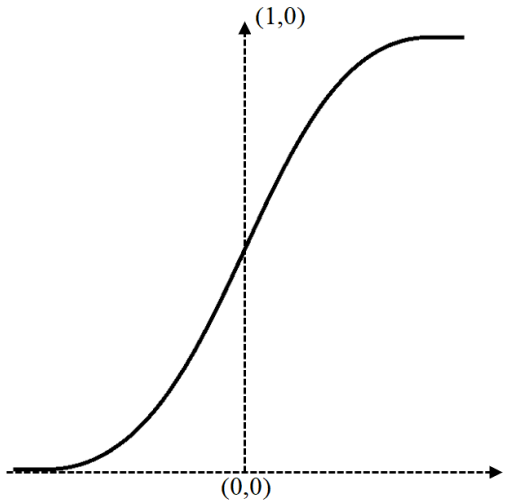


Figure B.5. The sigmoid activation function. The sigmoid activation function scales its input to a range between 0 and 1. Inspired by Sharma et al. [2017]; Nwankpa et al. [2018]

The sigmoid activation function is defined by equation B.3 with a bias of zero. [Nwankpa et al., 2018]

$$f(x) = \frac{1}{1 + e^{-x_i}} \tag{B.3}$$

B.3.4 Softmax activation function

The softmax activation function was used as the output activation function in the classifier. The softmax was a combination of multiple sigmoid functions, returning a probability of both the CAD label and the non-CAD label. Thus, the predicted label of each CCTA volume was the label with the highest probability. The sum of the probabilities for the CAD and non-CAD labels of a CCTA volume equaled 1. This was achieved by applying a standard exponential function to each element of the input vector and then normalizing the input element by dividing with the sum of all the exponentials. The following equation shows the mathematical expression for the softmax activation function:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{B.4}$$

$\sigma(z)_j$ is the probability for either of the labels, K refers to the total number of labels, and z_k is the total of the elements in the input vector. [Sharma et al., 2017]

B.3.5 Tangent Hyperbolic activation function

The Tangent Hyperbolic (tanh) activation function normalized the input to be in a range between -1 and 1 [Chollet et al., 2018]. Equation B.5 shows the mathematical expression of a tanh activation function with a bias of zero.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{B.5}$$

The tanh activation function has the property of being zero-centered as shown in figure B.6. [Nwankpa et al., 2018]

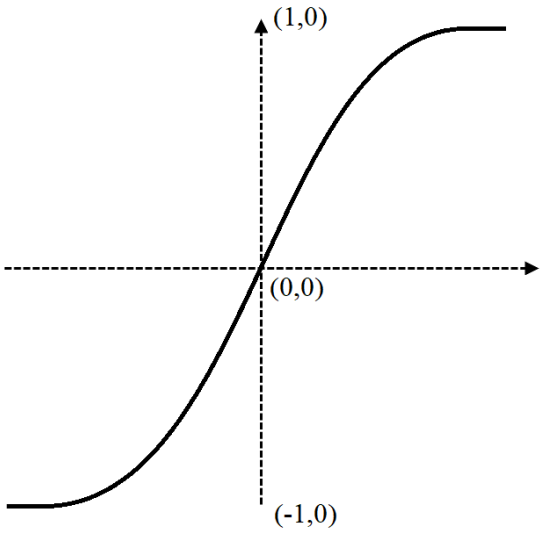


Figure B.6. The tanh activation function. The tanh activation function is zero-centered and scales the input in the range between -1 and 1. The bias value of the tanh activation function is zero. Inspired by Nwankpa et al. [2018].

B.4 Max pooling

The max pooling operation operated by sliding a window with a size of 2x2 pixels across the output feature map from the ReLU activation function and outputted the maximum value of the pixels in the window. The window was moved according to the stride of 2 pixels. Thus, the window was moved two pixels each time. Thereby, the input feature map was down sampled. [Chollet et al., 2018] The max pooling operation is illustrated in figure B.7.

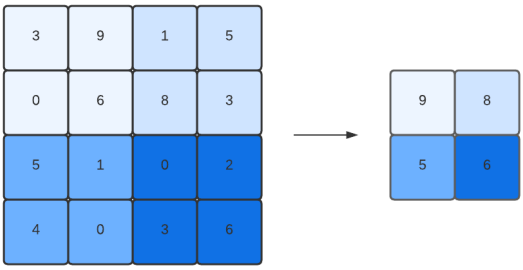


Figure B.7. The concept of the max pooling operation. The maximum value of the pixels in the window with a size of 2x2 is outputted from the max pooling operation. Inspired by Bonaccorso [2018].

B.5 Dropout

Dropout is a regularization technique used in neural networks preventing them from overfitting to the training data. The dropout operation randomly drops out or set some of the output hidden units to zero. The concept of dropout is illustrated in figure B.8. The amount of zeroed output hidden units was determined by the dropout rate, which was the fraction of zeroed hidden units. Usually, the dropout rate varies between 0.2 and 0.5,

corresponding to a range of 20% to 50% zeroed hidden units, where a dropout rate of 0.5 is shown in figure B.8. [Chollet et al., 2018]

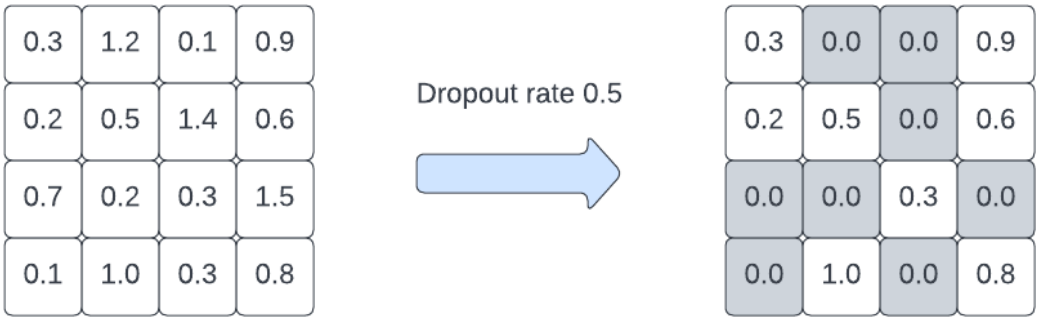


Figure B.8. Concept of random dropout with a dropout rate of 0.5. Inspired by Chollet et al. [2018]

B.6 Fully connected neural network

The FCNN consisted of an input layer and dense layers. The input layer distributed each element of the input to each node in the first dense layer. Hereafter, the dense layers connected each previous node to each following node. Thus, the output of one node in the previous layer was the input to each node in the sequential layer. When calculating the output of a node, the inputs to the node were summed and passed through a non-linear activation function, which determined the output of the node. The concept of the FCNN is illustrated in figure B.9 [Khan et al., 2018]

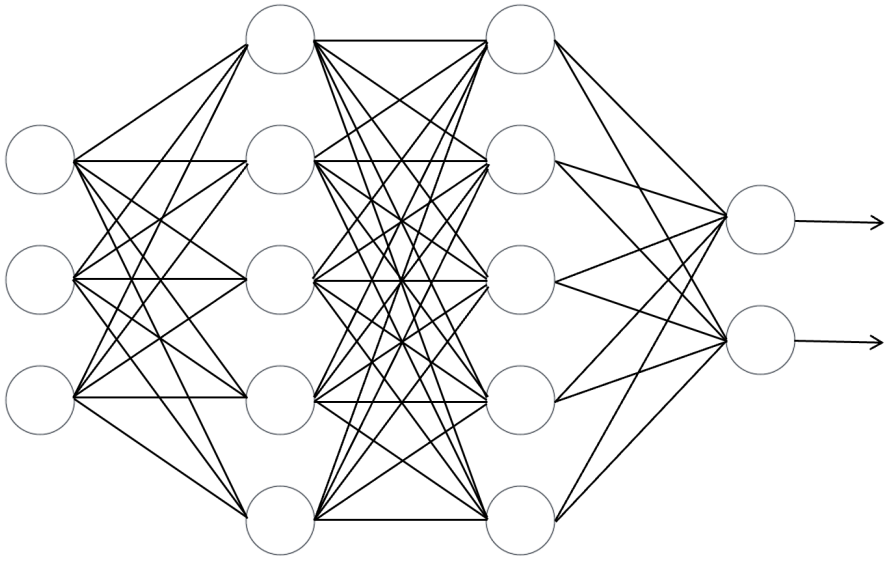


Figure B.9. Concept of a fully connected neural network. Inspired by Khan et al. [2018].

Training of the CAD network

C

C.1 Gradient descent algorithm

The gradient descent optimization algorithm aimed to minimize the loss function. The concept of gradient descent is illustrated in figure C.1. The gradient of the loss function gives the direction of the steepest descent and by calculating the negative of the gradient, the direction decreasing the loss function was found, indicating how much the parameters must be adjusted for a better prediction. By using gradient descent, the CAD network iterative moved closer to the minimum value of the loss function by taking small steps in the direction given by the gradient. How much the parameters were adjusted according to the loss function was controlled by the optimizer's learning rate. To find an optimal learning rate toward the loss function's minimum, an stochastic gradient descent-based optimizer was used. [Chollet et al., 2018]

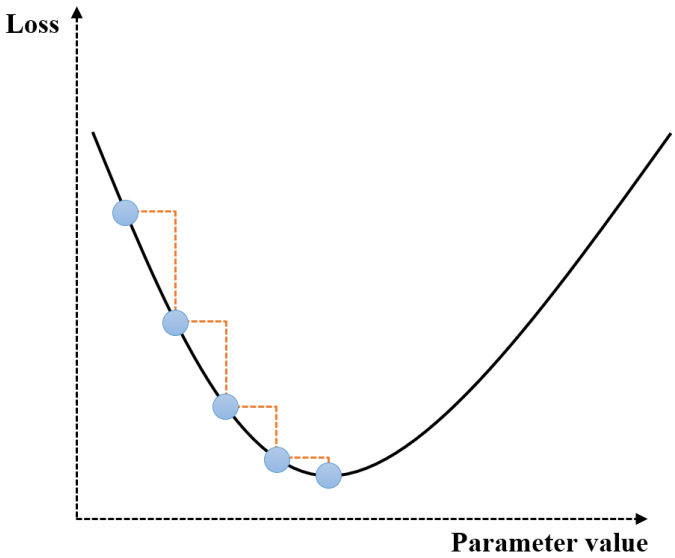


Figure C.1. Concept of gradient descent optimization algorithm. Inspired by Chollet et al. [2018].

The equation for the gradient descent optimization algorithm is presented in equation C.1:

$$w_{t+1} = w_t - \alpha \cdot \frac{\partial L}{\partial x} \tag{C.1}$$

where w_{t+1} is the computed parameter value, w_t is the prior parameter value, α is the learning rate, and $\frac{\partial L}{\partial x}$ is the gradient for the direction of the descent. [Chollet et al., 2018]

BPTT and backpropagation used the chain rule to calculate the gradient backward through the layers of the CAD network to adjust the parameters. The chain rule can be seen in equation C.2:

$$\frac{\partial h}{\partial x_i} = \sum_j \frac{\partial h}{\partial u_j} \cdot \frac{\partial u_j}{\partial x_i} \tag{C.2}$$

Here $x \in \mathrm{R}^m$ and $u \in \mathrm{R}^n$ indicating that the inner function maps m inputs to n outputs and the outer function receives n inputs to produce an output, h. Further, $i = 1, ..., m$ and $j = 1, ..., n$. [Chollet et al., 2018; Guo, 2013]

C.1.1 Adam optimizer

The Adam algorithm was an adaptive learning rate optimization algorithm. It is a combination of an adaptive gradient algorithm and root mean square propagation. The concept of the Adam optimizer is presented in figure C.2. [Kingma and Ba, 2014]

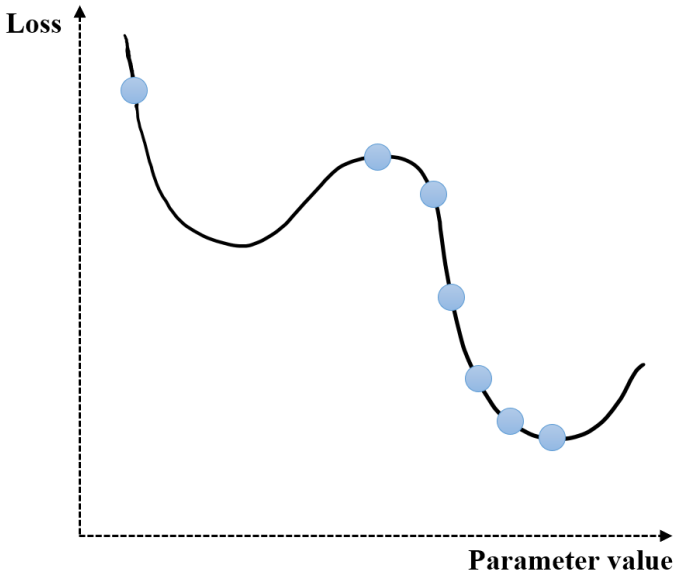


Figure C.2. The concept of the Adam optimizer. The adaptive learning rate varies the step size of the update of the parameter reaching the global minimum during training. Inspired by Kingma and Ba [2014].

The calculation of the adaptive learning rate was done by applying estimations of the first and second moments of the gradient, where the first moment was mean and the second moment was uncentered variance. To estimate the moments, an exponential moving average calculated on the gradient was used. The adaptive learning rate was calculated by the following equation:

$$w_{t+1} = w_t - \widehat{m_t} \left(\frac{\alpha}{\sqrt{\widehat{v_t}} + \epsilon} \right) \tag{C.3}$$

where w_{t+1} is the computed parameter value, w_t is the prior parameter value, $\widehat{m_t}$ is the bias-corrected first moment, α is the learning rate, $\widehat{v_t}$ is the bias-corrected second moment, and ϵ is a small positive constant. [Kingma and Ba, 2014]

Hyper-parameter optimization of the CAD network

D

A manual hyper-parameter optimization was performed on the CAD network to find which hyper-parameters to use for the identification of CCTA volumes containing either CAD or not. This was done as the baseline architecture could not identify CCTA volumes with CAD or not. Each hyper-parameter was tested three times to avoid noisy fitness evaluation, which is when a neural network produces different results because of different initialization conditions and random batch training [Stathakis, 2009]. The method of early stopping was implemented to avoid overfitting to the training data by stopping the training if the performance of the CAD network validation loss did not decrease over 10 epochs. The 10 epoch was chosen to ensure that the decrease in validation loss was not a coincidence, thus, that the training of the CAD network was not stopped before the losses converge. Afterwards, the hyper-parameter results were evaluated according to the selection criteria in table D.1 to find the hyper-parameter which had the best performance according to the selection criteria. If the performance of the CAD network improved with a given hyper-parameter, the architecture was updated to ensure the best performance of the CAD network.

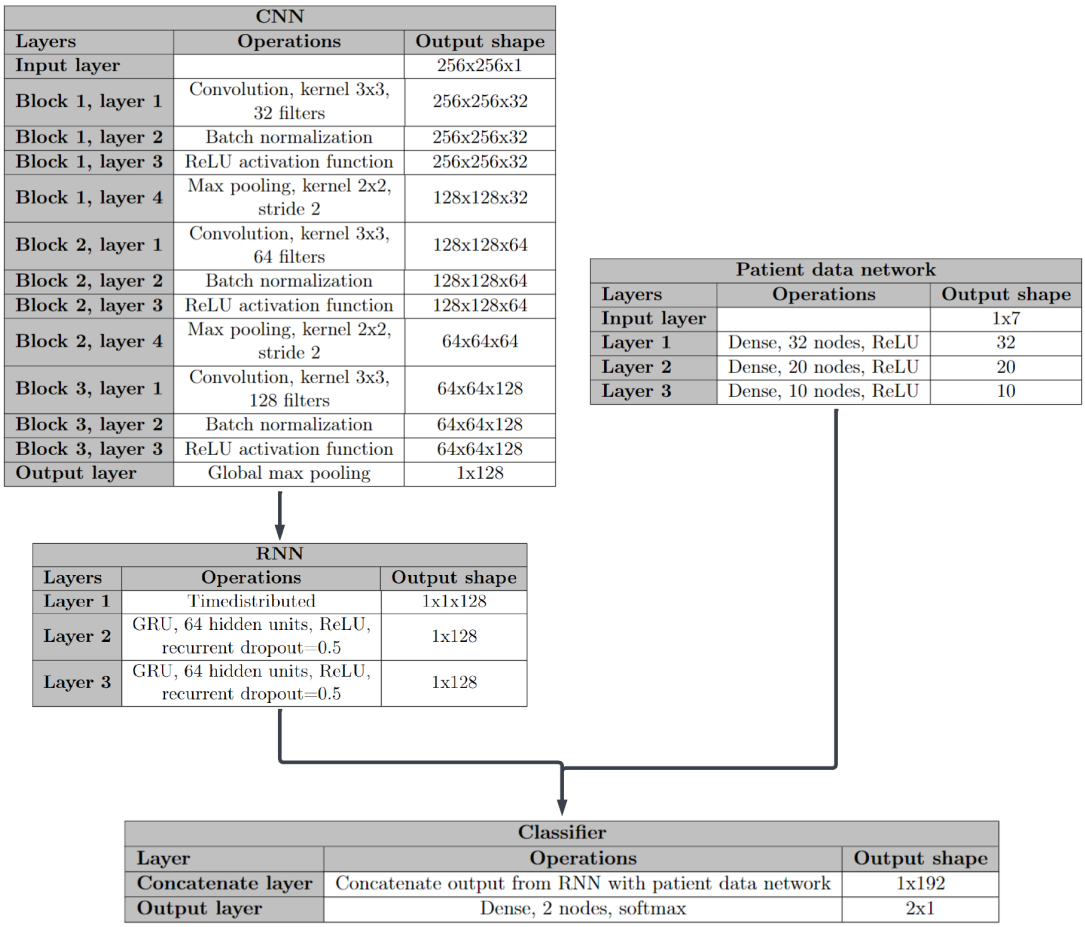
When choosing hyper-parameters, three criteria were used, shown in table D.1. The first criterion was assessed through a visual inspection of the loss curves of the training- and validation losses. Moreover, it must be the first criterion to be fulfilled as a descending tendency in losses through the training indicated that the network was getting better at learning and generalizing. The second criterion was that the loss difference between the training- and validation loss in the last epoch had to be the lowest. It must be fulfilled to ensure that the CAD network did not overfit to the training data as this indicated that the CAD network could not generalize to unseen data.

Criteria for selecting hyper-parameters	
1.	The training- and validation losses must have a descending tendency
2.	The average difference between training- and validation losses must be as low as possible

Table D.1. Criteria used to selected hyper-parameters.

The starting point of the hyper-parameter optimization was inspired by the architecture by Zreik et al. [2018b] and Spasov et al. [2018]. The baseline architecture prior to the hyper-

parameter optimization is shown in figure D.1. The learning rate used in the baseline architecture was 0.001. The baseline architecture was tested three times to create a starting reference to compare the following hyper-parameter configurations. The results from the baseline architecture can be seen in table D.2 and in figure D.2a, D.2b, and D.2b. The hyper-parameters was tested sequentially. First hyper-parameters such as number of recurrent layers and GRUs, and the use of dropout in the RNN was tested. Afterwards, the activation function in the classifier was tested and lastly, the learning rate was tested.



CNN		
Layers	**Operations**	**Output shape**
Input layer		256x256x1
Block 1, layer 1	Convolution, kernel 3x3, 32 filters	256x256x32
Block 1, layer 2	Batch normalization	256x256x32
Block 1, layer 3	ReLU activation function	256x256x32
Block 1, layer 4	Max pooling, kernel 2x2, stride 2	128x128x32
Block 2, layer 1	Convolution, kernel 3x3, 64 filters	128x128x64
Block 2, layer 2	Batch normalization	128x128x64
Block 2, layer 3	ReLU activation function	128x128x64
Block 2, layer 4	Max pooling, kernel 2x2, stride 2	64x64x64
Block 3, layer 1	Convolution, kernel 3x3, 128 filters	64x64x128
Block 3, layer 2	Batch normalization	64x64x128
Block 3, layer 3	ReLU activation function	64x64x128
Output layer	Global max pooling	1x128

Patient data network		
Layers	**Operations**	**Output shape**
Input layer		1x7
Layer 1	Dense, 32 nodes, ReLU	32
Layer 2	Dense, 20 nodes, ReLU	20
Layer 3	Dense, 10 nodes, ReLU	10

RNN		
Layers	**Operations**	**Output shape**
Layer 1	Timedistributed	1x1x128
Layer 2	GRU, 64 hidden units, ReLU, recurrent dropout=0.5	1x128
Layer 3	GRU, 64 hidden units, ReLU, recurrent dropout=0.5	1x128

Classifier		
Layer	**Operations**	**Output shape**
Concatenate layer	Concatenate output from RNN with patient data network	1x192
Output layer	Dense, 2 nodes, softmax	2x1

Figure D.1. An overview of the baseline architecture for the hyper-parameter optimization.

The loss curves of the training- and validation loss for each hyper-parameter configuration were plotted, according to the epochs.

D.1 Number of recurrent layers

The hyper-parameter optimization of the number of recurrent layers was tested as the addition of more layers gives more tunable parameters, thus, more learning power. The hyper-parameter optimization of the number of recurrent layers consisted of two tests in addition to the test of the baseline architecture. The first test consisted of testing three recurrent layers with 64 GRUs. The second test was performed on an RNN architecture with four recurrent layers with 64 GRUs. The results of the hyper-parameter

optimization of the number of recurrent layers are presented in table D.2. The hyper-parameter configuration of three recurrent layers obtained the lowest mean difference between training- and validation loss.

Baseline architecture Two recurrent layers		**Three recurrent layers**		**Four recurrent layers**	
Mean	***Range***	***Mean***	***Range***	***Mean***	***Range***
0.8908	0.5013 - 1.2656	**0.1906**	**0.035 - 0.3893**	0.1926	0.0644 - 0.2851

Table D.2. Results from the hyper-parameter optimization of the number of recurrent layers, where the baseline architecture, three, and four recurrent layers were tested three times each. ***Mean*** is the mean difference between the training- and validation losses in the last epoch of the training in the three tests. ***Range*** is the minimum and maximum difference between the training- and validation losses. The ***Mean*** and ***Range*** of the chosen hyper-parameter configuration are highlighted in blue.

The loss curves of the training- and validation loss for the hyper-parameter optimization of the number of recurrent layers are plotted in figure D.2a to D.4c. In the three tests of the baseline architecture, a general descending tendency for the training loss were observed along with a overall flat tendency in the validation loss. Furthermore, intersections between training- and validation loss in all three tests were observed. In the first test presented in figure D.2a, a descending tendency for both the losses was observed from epoch 1 to 5 hereafter the validation loss increased and an intersection was seen in epoch 4. The second test presented in figure D.2b illustrated an increasing validation loss, which indicated overfitting to the training data. Moreover, intersections were seen in epoch 4, 5, and 6. In the last test presented in figure D.2c, a fluctuating tendency in the validation loss was observed along with intersections between training- and validation loss in epoch 4, 6, and 17.

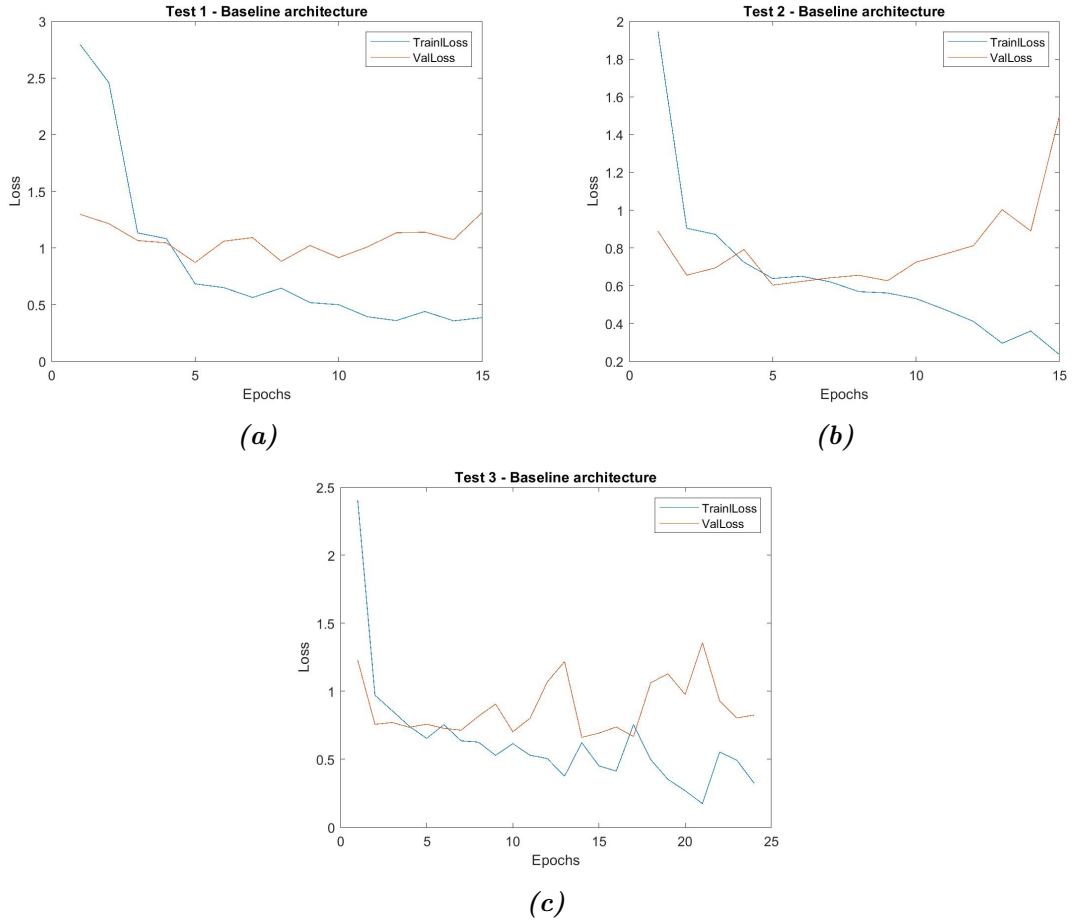


Figure D.2. Loss curves for the CAD network with the baseline architecture.

The loss curves in figure D.3 for testing three recurrent layers showed the same general descending tendency in the training losses and a flat tendency in the validation as the loss curves from the baseline architecture. The first test, presented in figure D.3a, had one intersection in epoch 8 observed. Moreover, the validation loss had an ascending tendency, indicating overfitting. In the second test, presented in figure D.3b, four intersections were observed in epochs 3, 5, 7, and 9. The last test, presented in figure D.3c, illustrated four intersections in epochs 4, 7, 9, and 10. Furthermore, the validation loss showed fluctuations. In both tests two and three, a flat tendency in the validation loss was observed.

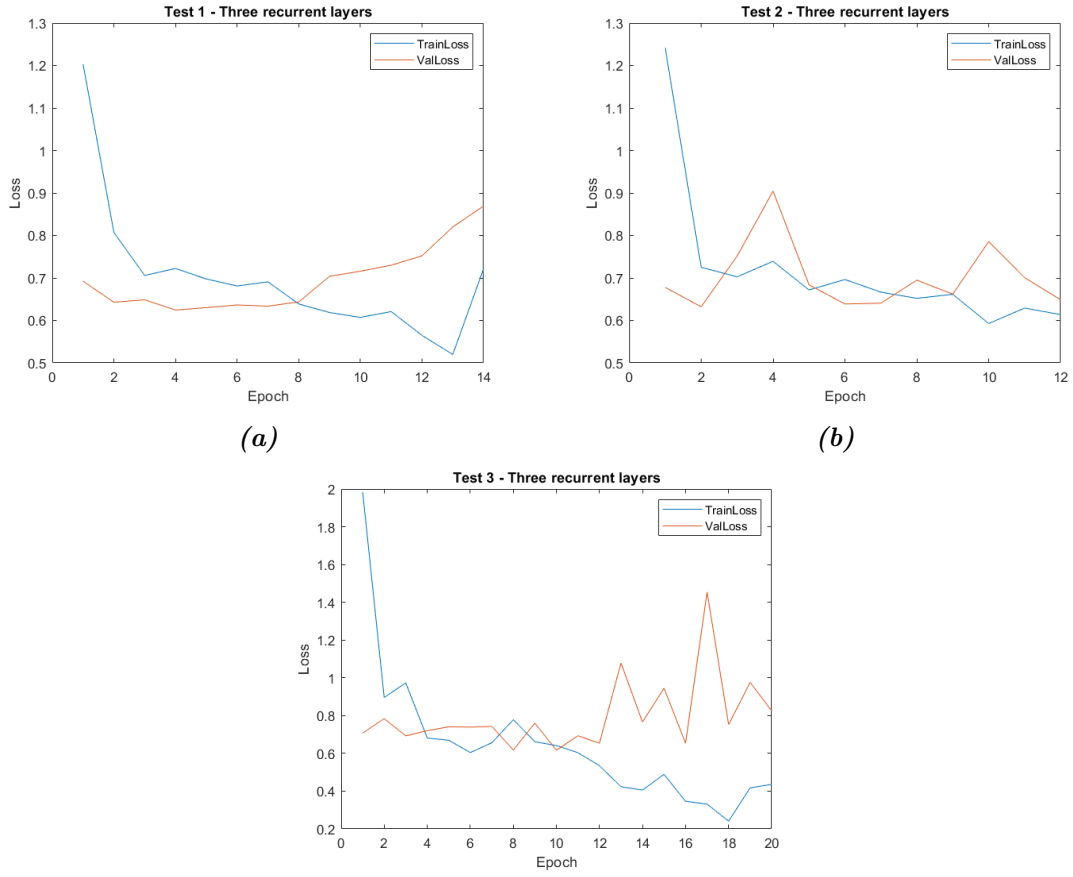


Figure D.3. Loss curves for the CAD network with three recurrent layers containing 64 GRUs.

The same general tendency as in the two previous tests of testing baseline architecture and three recurrent layers were observed for testing four recurrent layers. In the first and second tests for testing four recurrent layers, presented in figure D.4a and D.4b, an ascending tendency in the validation loss was observed, which indicated overfitting. Moreover, one intersection in epoch 3 was observed in both tests. In the last test, presented in figure D.4c, one intersection in epoch 2 was observed. Moreover, the validation loss had ascending tendency from epoch 1 to 9, but a descending tendency from epoch 9 to 11.

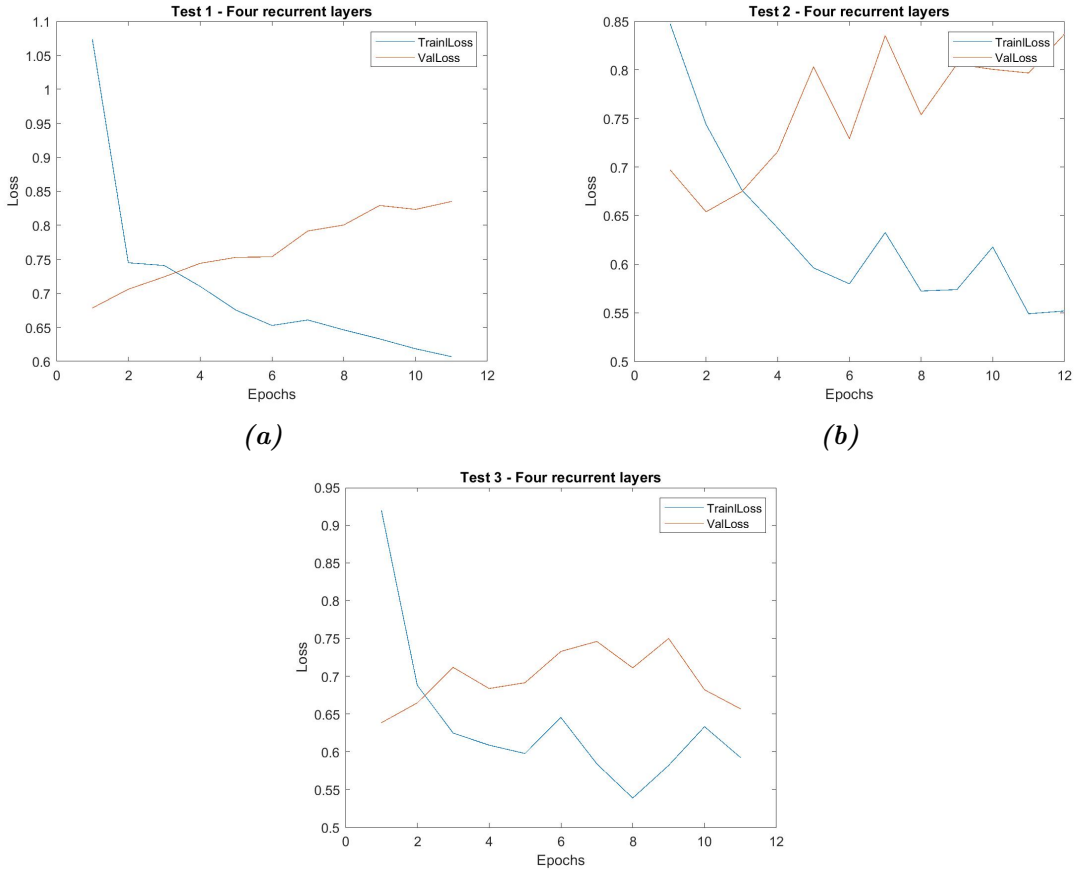


Figure D.4. Loss curves for the CAD network with four recurrent layers containing 64 GRUs.

The results from the hyper-parameter test for the number of recurrent layers showed that all three architectures had a descending tendency in training loss and an ascending or flat validation loss. Thus, none of the three tests had a descending tendency in both losses. However, the architecture with three recurrent layers had the lowest mean difference of 0.1906 between the training- and validation loss, thus, it was chosen to use three recurrent layers.

D.2 Number of GRUs

The hyper-parameter optimization of the number of GRUs in the recurrent layers was tested as the more GRUs in the recurrent layers, the more complex information could be learned. The hyper-parameter optimization of the number of GRUs consisted of one test. The loss curves can be seen in figure D.5 and the mean difference of the losses is shown in table D.3. The architecture containing 128 GRUs in the three recurrent layers was compared to the architecture containing 64 GRUs in the three recurrent layers, found through the previous hyper-parameter optimization in table D.3. When comparing the two architectures, the lowest mean difference between training- and validation loss was obtained with the hyper-parameter configuration of three recurrent layers with 128 GRUs.

64 GRUs		128 GRUs	
Mean	***Range***	***Mean***	***Range***
0.1906	0.035 - 0.3893	**0.1241**	**0.0939 - 0.2036**

Table D.3. Results from the hyper-parameter optimization of the number of GRUs, where 64 and 128 GRUs were tested three times each. ***Mean*** is the mean difference between the training- and validation losses in the last epoch of the training in the three tests. ***Range*** is the minimum and maximum difference between the training- and validation losses. The ***Mean*** and ***Range*** of the chosen configuration are highlighted in blue.

From figure D.5, a general descending tendency in the training loss and intersections between the training- and validation loss was observed. In the first and last tests, presented in figure D.5a and D.5c, a flat tendency in the validation loss was observed. Moreover, seven intersections in epoch 4, 7, 9, 14, 16, 18, and 19 was observed in figure D.5a, whereas one intersection in epoch 7 was present in figure D.5c. In the second test, presented in figure D.5b, a descending tendency in the validation loss from epoch 1 to 31, was observed, and from epoch 31 to 35 the validation loss increased. Furthermore, two intersections in epochs 5 and 7 were observed.

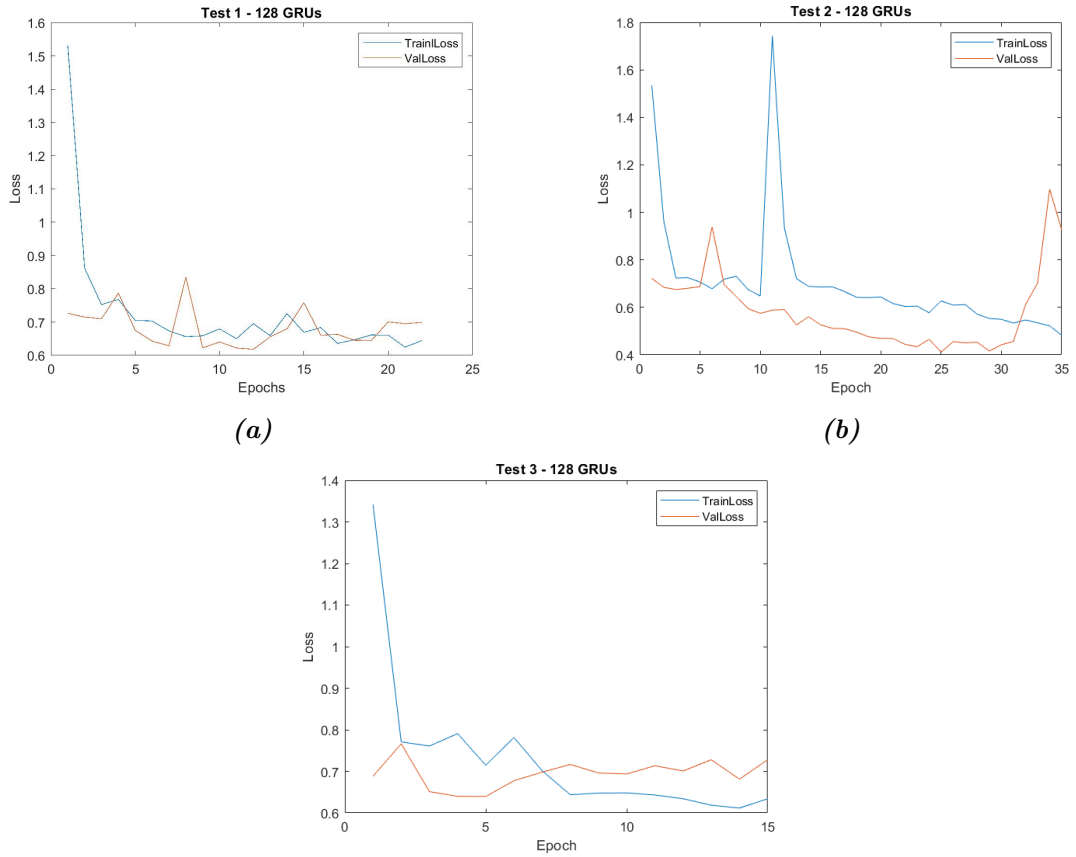


Figure D.5. Loss curves for the CAD network obtained through hyper-parameter optimization testing 128 GRUs.

The results from the hyper-parameter test for the number of GRUs showed that every test of the architecture with 128 GRUs in the three recurrent layers had a descending tendency

in training loss. The only test with a descending validation loss was the second test, seen in figure D.5b. The architecture with 128 GRUs had the lowest mean difference of 0.1241 between the training- and validation loss, thus, the architecture with 128 GRUs in the three recurrent layers fulfilled the selection criterion.

D.3 Dropout

To review the use of dropout in the RNN, the hyper-parameter optimization of dropout in the recurrent layers consisted of testing the architecture without dropout. The test of dropout was performed to see if it prevented the CAD network from overfitting to the training data as dropout is a regularization technique for preventing overfitting. A dropout rate of 0.5 was used as the baseline architecture applied a dropout rate of 0.5. Since the previous architecture consisting of three recurrent layers with 128 GRUs contained dropout with the rate of 0.5, this architecture was used for comparison. The loss curves of the previous architecture are shown in figure D.5. Thus, an architecture containing three recurrent layers with 128 GRUs without a dropout rate was tested. The loss curves from a training of the architecture without dropout are shown in figure D.6. Furthermore, the mean difference and range between the training- and validation losses are presented in table D.4, where the hyper-parameter configuration with dropout obtained the lowest mean difference between the training- and validation loss.

Dropout		**Without dropout**	
Mean	***Range***	***Mean***	***Range***
0.1241	**0.0939 - 0.2036**	0.2945	0.0569 - 0.6294

Table D.4. Results from the hyper-parameter optimization of applying dropout in the recurrent layers were tested three times each. ***Mean*** is the mean difference between the training- and validation losses in the last epoch of the training in the three tests. ***Range*** is the minimum and maximum difference between the training- and validation losses. The ***Mean*** and ***Range*** of the chosen configuration are highlighted in blue.

The three tests of the CAD network without dropout showed either flat loss curves with a spike, as seen in figure D.6a and D.6b, or overfitting to the training data set, seen in figure D.6c. The flat tendency in figure D.6a and D.6b indicated that the learning of the CAD network did not improve during training. Further, a spike approximately in epoch 7 and 5 in figure D.6a and D.6b, respectively, was present. Moreover, the training- and validation losses intersected with each other in epoch 5, 6, 9, 11, 13, and 14 in figure D.6a, in epoch 3 and 7 in figure D.6b, and in epoch 4 and 6 in figure D.6c.

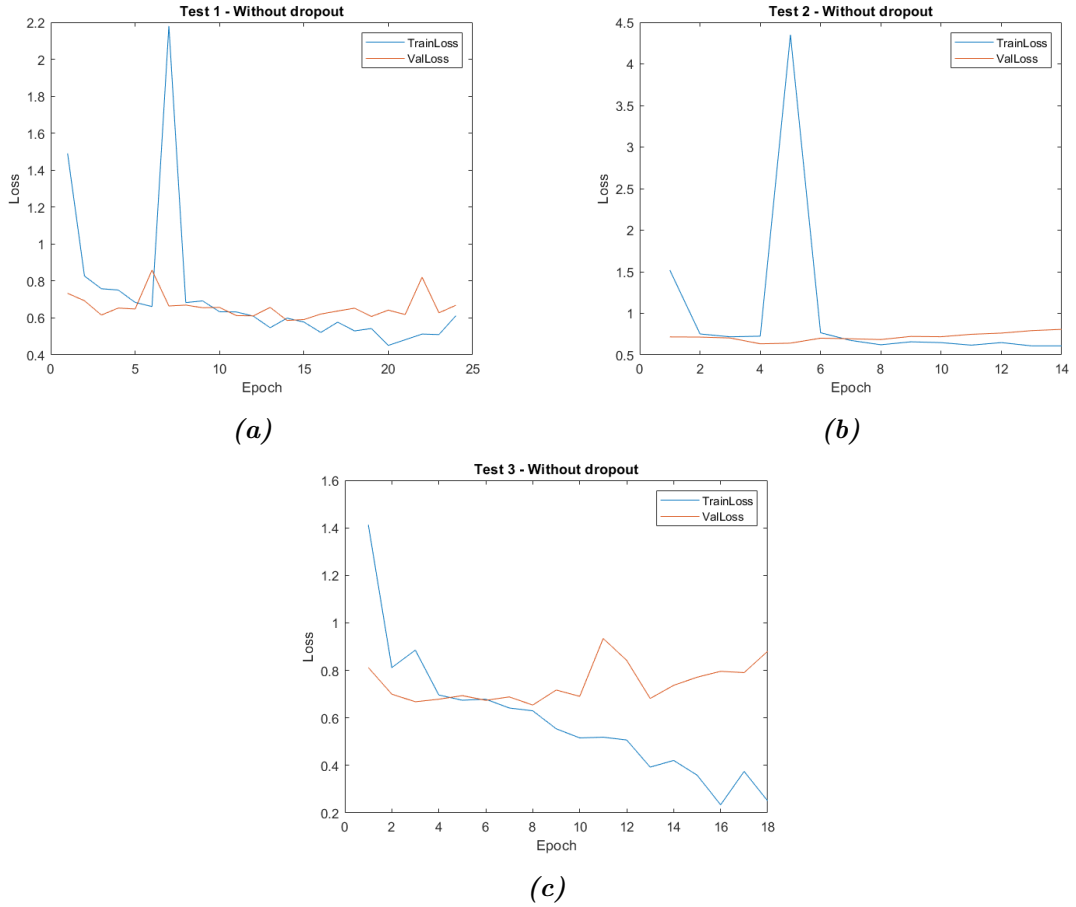


Figure D.6. Loss curves for the CAD network without dropout.

The results from the hyper-parameter configuration without dropout showed a descending tendency in training loss. However, the validation loss was flat or slightly ascending. Compared to the loss curves of a hyper-parameter configuration with dropout, a descending tendency in the validation loss was observed in figure D.5b. When comparing the architecture with dropout, the lowest mean difference of 0.1241 between the training- and validation loss was seen, thus, the architecture with dropout was chosen.

D.4 Activation function

The hyper-parameter optimization of the activation function in the CAD network's classifier was performed, since the classification was a binary problem, but the starting hyper-parameter configuration contained a softmax activation function for multi-class problems. Thus, a test of a hyper-parameter configuration containing the sigmoid activation function was performed, where the loss curves can be seen in figure D.7. The hyper-parameter configuration of the three recurrent layers with 128 GRUs contained the activation function softmax in the classifier. Therefore, the test for applying the activation function softmax equaled the results from the test with 128 GRUs in the three recurrent layers. Thus, the loss curves for using the softmax activation function corresponds to figure D.5a, D.5b, and D.5c. The results of the hyper-parameter optimization of the activation function in the classifier are presented in table D.5. The hyper-parameter configuration

with the activation function softmax obtained the lowest mean difference between training- and validation loss.

Softmax		Sigmoid	
Mean	***Range***	***Mean***	***Range***
0.1241	**0.0939 - 0.2036**	0.3620	0.3158 - 1.2251

Table D.5. Results from the hyper-parameter optimization of activation function, which consisted of testing softmax and sigmoid, were each tested three times. ***Mean*** is the mean difference between the training- and validation losses in the last epoch of the training in the three tests. ***Range*** is the minimum and maximum difference between the training- and validation losses. The ***Mean*** and ***Range*** of the chosen configuration are highlighted in blue.

In the first and last test, presented in figure D.7a and D.7c, a flat tendency in both the training- and validation loss were present. However, a spike in the training- and validation loss in figure D.7a and a spike in training loss in figure D.7c were observed. Furthermore, five intersections between the training- and validation loss in epoch 7, 8, 15, 23, and 24 in the first test was observed. In the last test, three intersections in epoch 2, 3, and 5, was observed. In the second test, illustrated in figure D.7b, a descending tendency in the training loss and an ascending tendency in the validation loss were observed along with intersections in epoch 6, 8, and 11.

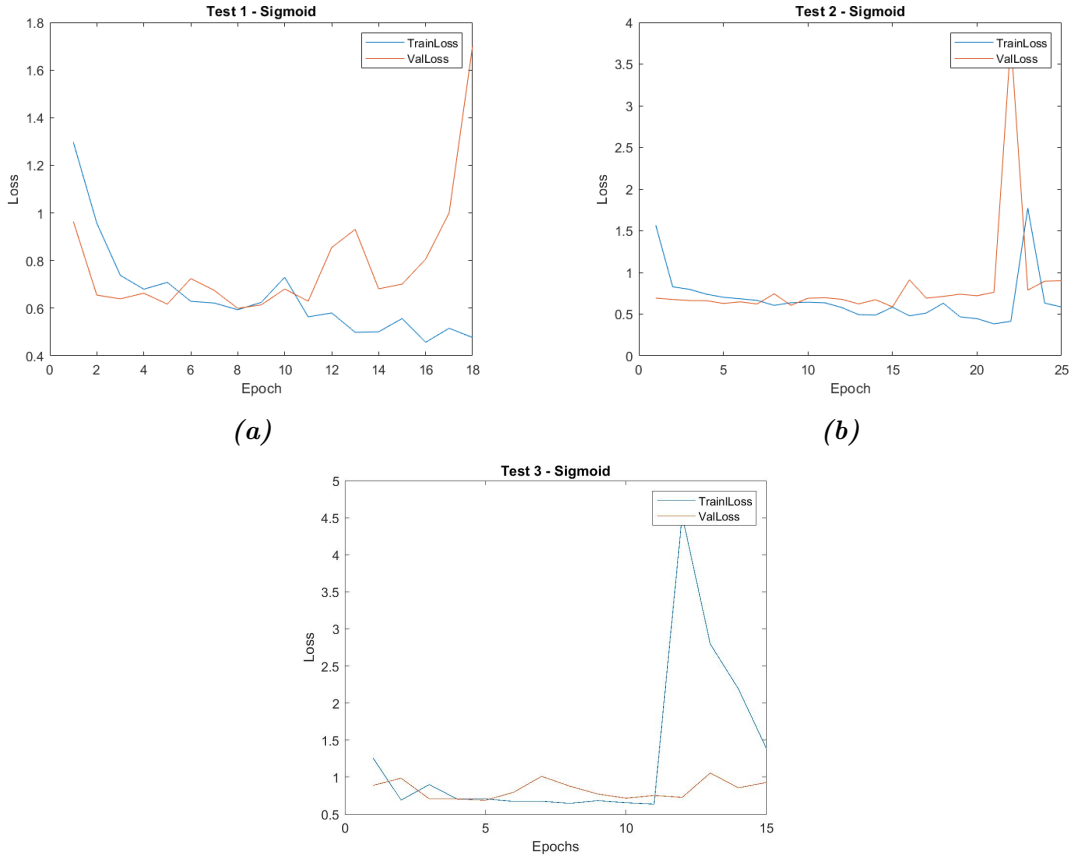


Figure D.7. Loss curves for the CAD network with the activation function sigmoid.

The results from the hyper-parameter test for applying softmax as an activation function in the classifier showed a descending tendency in the training loss. The architecture with softmax had the lowest mean difference of 0.1241 between the training- and validation loss, thus, the architecture with softmax in the classifier was chosen.

D.5 Learning rate

The hyper-parameter optimization of learning rate was tested to find the learning rate that is low enough that the CAD network converges without needing to many iterations but not as high that the optimal solution will be skipped [Chollet et al., 2018]. The hyper-parameter optimization consisted of three tests. The results from the test of the learning rate were compared to the results from the hyper-parameter configuration of three recurrent layers with 128 GRUs, presented in table D.3 and figure D.5, where a learning rate of 0.001 was used. Therefore, the test for applying an learning rate of 0.001 equaled the results from the test with three recurrent layers with 128 GRUs. The three tests consisted of testing a learning rate varying from 0.0001 to 0.000001. The result of the hyper-parameter optimization of learning rate are presented in table D.6. The hyper-parameter configuration with the learning rate of 0.00001 obtained the lowest mean difference between training- and validation loss.

Learning rate of 0.001		**Learning rate of 0.0001**	
Mean	***Range***	***Mean***	***Range***
0.1241	0.0939 - 0.2036	0.7473	0.5782 - 0.8862
Learning rate of 0.00001		**Learning rate of 0.000001**	
Mean	***Range***	***Mean***	***Range***
0.0055	**0.0032 - 0.0099**	0.1225	0.1102 - 0.1422

Table D.6. Results from the hyper-parameter of learning rate, which consisted of four different learning rates, where each were tested three times.
textbf*Mean* is the mean difference between the training- and validation losses in the last epoch of the training in the three tests. ***Range*** is the minimum and maximum difference between training- and validation loss. The ***Mean*** and ***Range*** of the chosen learning rate are highlighted in blue.

The loss curves for learning rate 0.0001 showed a general descending tendency in the training loss and an ascending tendency in the validation loss, indicating overfitting. Intersections between the training- and validation loss were also presented in the loss curves. In the first test illustrated in figure D.8a, two intersections in epoch 10 and 12 were observed. In the second test illustrated in figure D.8b, four intersections in epoch 5, 6, 7 and 12 was observed. In the last test illustrated in figure D.8c, one intersection in epoch 7 was observed.

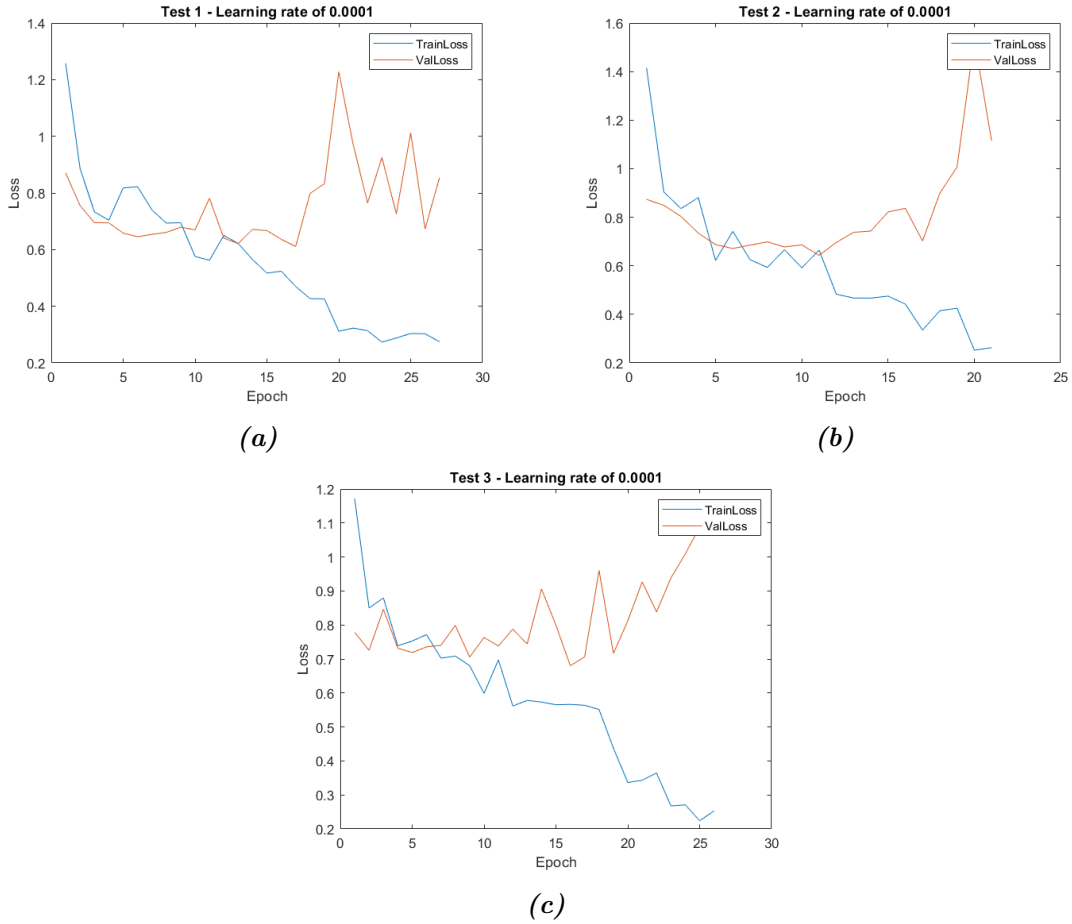


Figure D.8. Loss curves for the CAD network with a learning rate of 0.0001.

The test of a learning rate of 0.00001 in the CAD network shown a fluctuating, but descending tendency for both the training- and validation curves in figure D.9a, D.9b, and D.9a. Moreover, the training- and validation losses from the three tests intersected with each other during training. For figure D.9a the training- and validation loss intersected in epoch 6, 9, and 11 and in figure D.9b the losses intersected in epoch 2, 5, 7, 9, 13, 14, and in epoch 18. In figure D.9c, the training- and validation loss intersected in nearly every epoch for the first 20, approximately, and then again frequently in the following epochs.

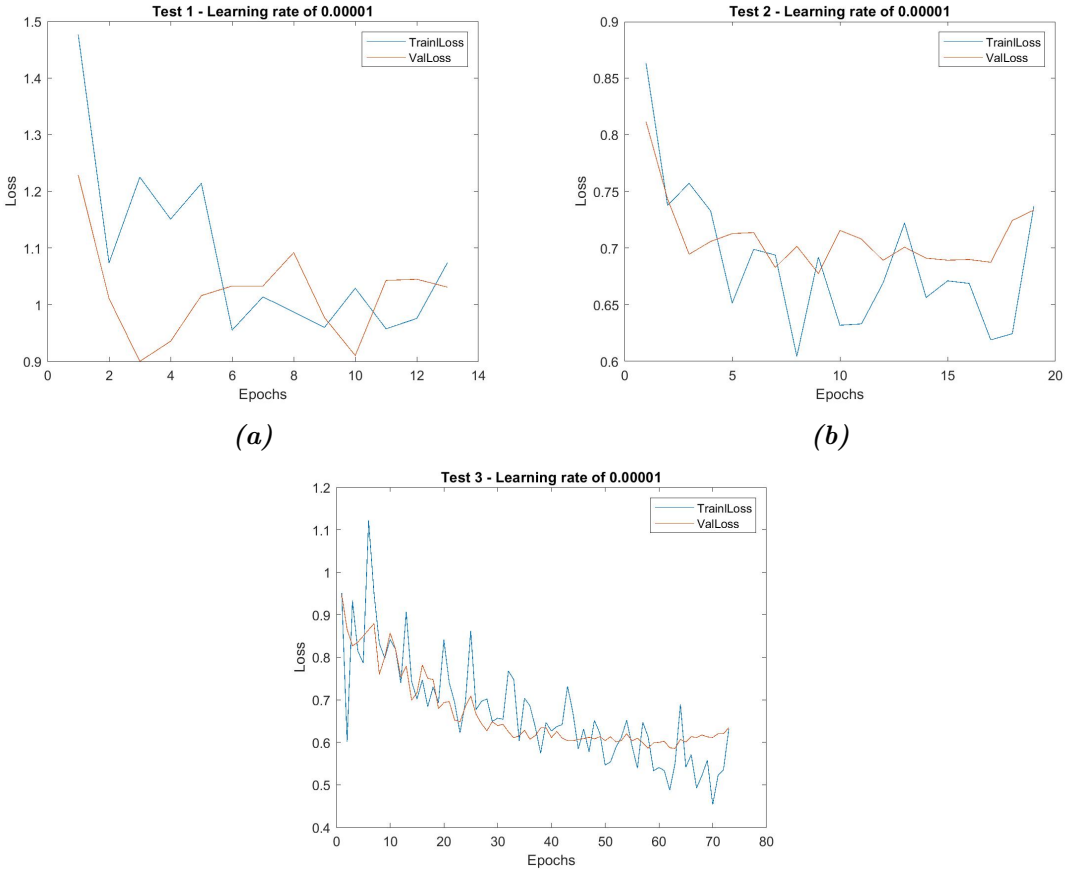


Figure D.9. Loss curves for the CAD network with a learning rate of 0.00001.

From the hyper-parameter optimization with a decreased learning rate of 0.000001, an over overall tendency of fluctuating, but descending training loss curves was seen in figure D.10a, D.10b, and D.10a. For the validation loss curves in D.10b and D.10c, a relative flat tendency was observed, although the validation curve in figure D.10c was slightly descending at the first 13 epochs. However, validation loss was descending in figure D.10a. Further, intersections between training- and validation loss was present at epoch 4, 6, 8, 9, 11, 13, 15, 17, 18, 20, 22, 27, 31, 33, and 35 in figure D.10a.

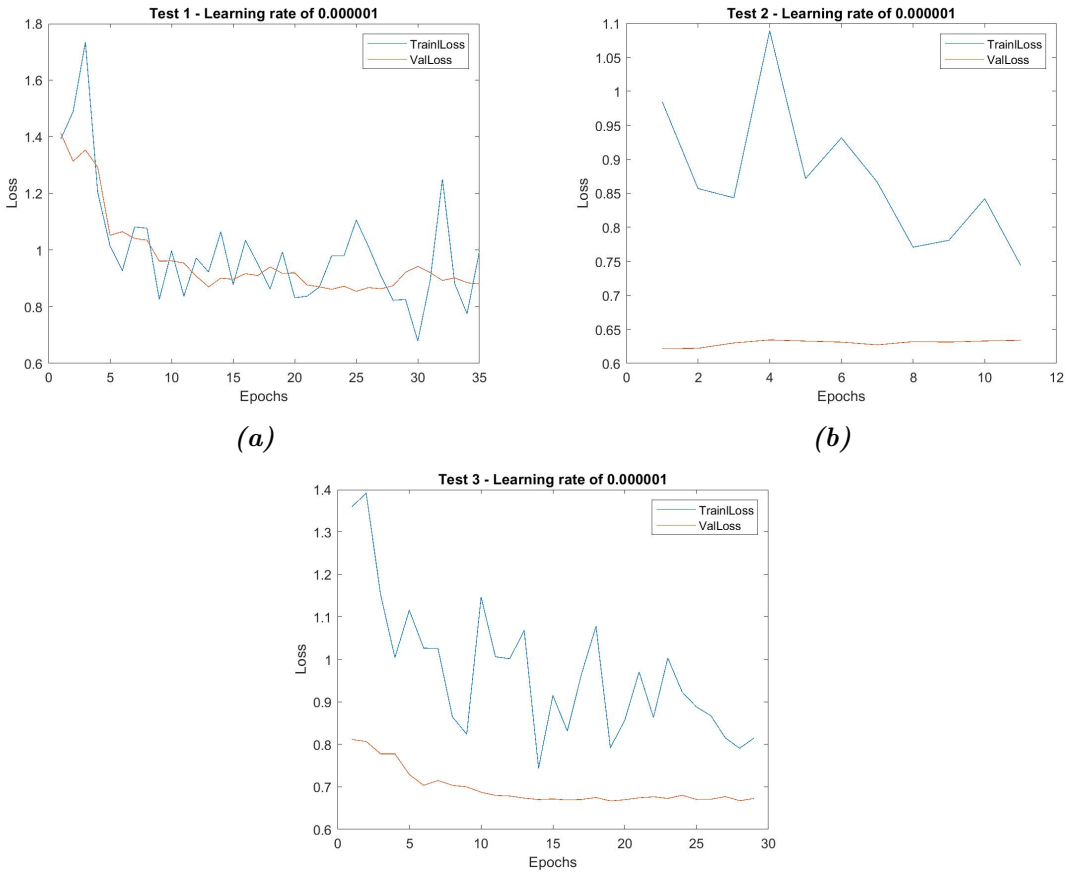


Figure D.10. Loss curves for the CAD network with a learning rate of 0.000001.

The results from the hyper-parameter optimization of the learning rate indicated that the learning rate of 0.00001 fulfilled the criteria by obtaining the lowest mean difference between training- and validation loss of 0.0055. Moreover, the loss curves for the learning rate of 0.00001 had a descending tendency for both the training- and validation loss, thereby, a learning rate of 0.00001 was chosen.