

Master Thesis



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Localization and segmentation of in-vivo ultrasound carotid artery images

Martin Kostelanský

Supervisor: prof. Dr. Ing. Jan Kybic
Field of study: Open Informatics
Subfield: Artificial Intelligence
January 2021

I. Personal and study details

Student's name: **Kostelanský Martin** Personal ID number: **435373**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Computer Science**
Study program: **Open Informatics**
Specialisation: **Artificial Intelligence**

II. Master's thesis details

Master's thesis title in English:

Localization and segmentation of in-vivo ultrasound carotid artery images

Master's thesis title in Czech:

Lokalizace a segmentace in-vivo ultrazvukových obrazů karotidy

Guidelines:

Bibliography / sources:

Automatic multi-organ segmentation using learning-based segmentation and level set optimization. T Kohlberger, M Sofka, et al - International Conference on Medical Image Computing, 2011

Automatic detection and measurement of structures in fetal head ultrasound volumes using sequential estimation and integrated detection network (IDN) M Sofka, J Zhang, S Good, SK Zhou, D Comaniciu - IEEE Transactions on Medical Imaging, 2014

ŘÍHA, Kamil, Jan MAŠEK, Radim BURGET, Radek BENEŠ and Eva ZÁVODNÁ. Novel method for localization of common carotid artery transverse section in ultrasound images using modified viola-jones detector. Ultrasound in medicine & biology, New York: ELSEVIER SCIENCE INC, 2013, vol. 39, No 10, p. 1887-1902. ISSN 0301-5629.

doi:10.1016/j.ultrasmedbio.2013.04.013.

Sifakis, Golemati: Robust Carotid Artery Recognition in Longitudinal B-Mode Ultrasound Images IEEE Transactions on Image Processing (Volume: 23 , Issue: 9 , Sept. 2014)

Name and workplace of master's thesis supervisor:

prof. Dr. Ing. Jan Kybic, Biomedical imaging algorithms, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **06.10.2020** Deadline for master's thesis submission: **05.01.2021**

Assignment valid until: **30.09.2022**

prof. Dr. Ing. Jan Kybic
Supervisor's signature

Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Acknowledgements

I would like to thank Professor Kybic for his valuable guidance and answering my e-mails after 10 p.m. Dedicated to my family and friends for their endless support during my studies.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 5 January 2021

Abstract

This thesis is focused on the three separate image recognition tasks—classification, localization, and segmentation of the ultrasound images of the carotid artery with stenosis. The first problem was successfully solved by a ResNet50 CNN and a created dataset with 1,679 images. Such a model was able to categorize four classes of the ultrasound images (longitudinal, transverse, Doppler, conical) with a test accuracy of 99.22%. The region of interest, the carotid artery, was localized on the transverse and longitudinal images by the novel Faster R-CNN. The IoU between predicted and true bounding boxes was greater than 0.75 in 90% of the test cases for both, the transverse and longitudinal test images. Further, the area of an artery was segmented into an artery wall with plaque, a lumen, and surrounding tissue. The U-net trained only on 75 images achieved an average image accuracy of 86.53% on the test data for the transverse section and 84.23% for the longitudinal section.

Keywords: Carotid artery stenosis, Ultrasound, Medical imaging, Deep learning, Image classification, Object localization, Image segmentation

Supervisor: prof. Dr. Ing. Jan Kybic
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

Abstrakt

Táto práca je zameraná na tri samostatné problémy týkajúce sa spracovania obrazu – klasifikáciu, lokalizáciu a segmentáciu ultrazvukových snímok stenózy krčnej artérie. Prvý zo zmienených problémov bol úspešne vyriešený použitím neurónovej siete ResNet50 a vytvorením datasetu so 1679 snímkami. Tento model bol schopný klasifikovať štyri triedy ultrazvukových snímok (pozdĺžny, priečny, Dopplerovský, kónický) s testovacou presnosťou 99,22%. Oblasť záujmu, krčná artéria, bola pomocou Faster R-CNN lokalizovaná na priečnych a pozdĺžnych snímkoch. IoU medzi predpovedaným a skutočným ohraničujúcim boxom u oboch typov snímkov bola vyššia ako 0,75 u 90% testovacích prípadov. Následne bola segmentovaná oblasť artérie na stenu artérie s plakom, lumen a okolité tkanivo. U-net natrénovaná len na 75 snímkach dosiahla priemernú testovaciu presnosť segmentácie snímku 86,53% pre priečne a 84,23% pre pozdĺžne snímky.

Klíčová slova: Stenóza karotídy, Ultrazvuk, Lekárske zobrazovanie, Hlboké učenie, Klasifikácia obrazu, Lokalizácia objektu, Segmentácia obrazu

Překlad názvu: Lokalizace a segmentace in-vivo ultrazvukových obrazů karotídy

Contents

1 Introduction	1	7.2 Faster R-CNN.....	34
2 Goals	3	7.2.1 Training	34
3 Background	5	7.3 Experiments and results	35
3.1 Carotid Artery Stenosis	5	8 Segmentation of ultrasound	
3.1.1 Diagnosis	7	carotid artery images	41
4 Existing methods	9	8.1 Dataset.....	41
4.1 Image Classification.....	9	8.1.1 Data augmentation.....	42
4.1.1 VGG	10	8.2 U-net	43
4.1.2 ResNet	10	8.2.1 Training	43
4.2 Object Localization	12	8.3 Experiments and results	44
4.2.1 Faster R-CNN	14	9 Conclusion	51
4.3 Segmentation	15	A Convolutional neural net	53
4.3.1 U-net.....	16	A.1 Convolutional layer	53
5 Data	19	A.2 Pooling layer	55
5.1 ANTIQUE dataset.....	19	A.3 Fully-connected layer	55
5.2 SPLab dataset	20	A.4 Architecture	56
6 Classification of ultrasound		A.5 Training	56
carotid artery images	21	B Implementation details	59
6.1 Dataset.....	22	B.1 Project structure	59
6.1.1 Data augmentation.....	23	B.2 Examples	59
6.2 CNN Architectures	24	C List of Abbreviations	61
6.2.1 Small CNN.....	24	D Bibliography	63
6.2.2 VGG-16	25		
6.2.3 ResNet50	25		
6.2.4 Training	25		
6.3 Experiments and results	26		
7 Localization of CCA and ICA in			
ultrasound images	31		
7.1 Dataset.....	31		
7.1.1 Data augmentation.....	32		

Figures

3.1 Carotid and vertebral artery	5	7.5 Examples of bounding boxes predicted by the best longitudinal Faster R-CNN	40
3.2 Carotid artery angioplasty with stenting	6	8.1 Transformations used in segmentation	42
4.1 Example of classification	9	8.2 Accuracy of the U-net models on the test sets	45
4.2 The building block of residual learning	11	8.3 The least accurate test segmentation mask of the longitudinal U-net	46
4.3 Example of localization	13	8.4 The most accurate test segmentation mask of the longitudinal U-net	47
4.4 Region proposal network	13	8.5 The least accurate test segmentation mask of the transverse U-net	48
4.5 The architecture of Fast R-CNN	14	8.6 The most accurate test segmentation mask of the transverse U-net	49
4.6 The architecture of Faster R-CNN	15	A.1 Application of convolution	54
4.7 Example of segmentation	16	A.2 Types of pooling layers	55
4.8 Architecture of U-net	17	A.3 Example architecture of CNN . .	56
5.1 SPLab dataset example	20	B.1 Project structure	60
6.1 ANTIQUE dataset example	21		
6.2 Example of similar images across categories	22		
6.3 Transformations used in classification	23		
6.4 Examples of images mislabeled by ResNet50	28		
6.5 Examples of images labeled correctly by ResNet50	29		
7.1 Transformations used in localization	33		
7.2 Carotids predicted by transverse Faster R-CNN	34		
7.3 Image mislabeled by transverse Faster R-CNN	37		
7.4 Examples of bounding boxes predicted by the best transverse Faster R-CNN	38		

Tables

4.1 Comparison of VGG-16 and VGG-19 architectures	10	8.3 Convolutional block of U-net	43
4.2 Comparison of ResNet50 and ResNet101 architectures	12	8.4 Transverse U-net results	44
6.1 Classification dataset	22	8.5 Longitudinal U-net results	44
6.2 Data augmentation for classification	24	A.1 Convolutional kernels	54
6.3 Sizes of classification models	24		
6.4 The architecture of Small CNN	25		
6.5 Training evaluation of classification models	26		
6.6 Test evaluation of classification models	27		
6.7 Test errors of ResNet50	27		
7.1 Localization dataset	32		
7.2 Data augmentation for localization	32		
7.3 Results of transverse Faster R-CNN trained on the ANTIQUE dataset	35		
7.4 Results of transverse Faster R-CNN trained on the ANTIQUE+SPLab dataset	36		
7.5 Number of detected objects by Faster R-CNNs	36		
7.6 Results of transverse Faster R-CNNs trained on ANTIQUE and SPLab datasets	37		
7.7 Results of newly initialized longitudinal Faster R-CNN	39		
7.8 Results of pretrained longitudinal Faster R-CNN	39		
8.1 Segmentation dataset	41		
8.2 Data augmentation for segmentation	43		



Chapter 1

Introduction

Artificial intelligence is a scientific field that aims to build intelligent systems and understand the principles behind them [69]. Most of the researches assume that the ability to learn is a predisposition for intelligence [40]. Machine learning is a subfield of AI, which focuses on learning behavior from data. It has been applied in a wide range of applications, from natural language processing [53], finance [10], image processing [43] to medical diagnosis [50].

The use of electronic health records is increasing in the last decades, and an important part of patients' records consists of medical images [35]. Computed tomography (CT), magnetic resonance imaging (MRI), medical ultrasound, and positron emission tomography (PET) have become core tools in disease diagnostics. The digitization of medicine, combined with the successes of deep learning in image recognition [43, 74], led to its application in computer-aided diagnosis [81].

Carotid artery stenosis is a disease in which blood flow in an artery is reduced by atheromatous plaque. The symptoms of stenosis are hard to spot, and it might be unnoticed until the disease becomes severe enough to cause blood deprivation to the brain, transient ischemic attack, or even stroke [61]. In this work, the state-of-art image recognition deep learning models are applied to ultrasound carotid artery images, which will be later used in the research project "Evaluation of atherosclerotic plaque stability in carotids using digital image analysis of ultrasound images". This research aims to create a software tool for analyzing ultrasound images of carotid stenosis, and analyze visual differences in digital images of unstable (symptomatic) and stable (asymptomatic) plaques. Another goal is to verify the hypothesis that sonographic plaque characteristics can be associated with an increased risk of plaque progression and stroke risk [82].



Chapter 2

Goals

This thesis aims is to create a collection of machine learning methods for ultrasound carotid artery images. All of them are interconnected, nevertheless, each of them solves a different image processing task:

1. classification
2. localization
3. segmentation

The first goal is to propose and implement a model able to classify different categories of ultrasound images, namely transversal, longitudinal, conical, and Doppler ones. Later on, the project focuses on transversal and longitudinal classes only. The second task is to detect the area with the carotid artery in the image, which can be defined as a localization task. The last step is segmentation. The developed solution needs to segment the particular parts of an artery with stenosis—artery wall, plaque, lumen, and surrounding tissue.

Chapter 3

Background

3.1 Carotid Artery Stenosis

Blood to the head is transported by carotid and vertebral arteries (VA) (Figure 3.1). Both of them are in pairs, symmetrically on both sides of the neck. They later split into smaller arteries and arterioles, that together create a vascular loop supplying the brain with blood. The right common carotid artery (CCA) originates from the brachiocephalic artery and later splits into internal (ICA) and external carotid artery (ECA). Left common carotid artery branches of aorta directly and continues up the neck, where it is divided into ICA and ECA as well. ECA is the main blood supplier to the meninges, scalp, and face. ICAs and VAs deliver blood to the Central nervous system [24]. Branches of ICA also supply eyes, extraocular muscles, and adjacent structures (lacrimal gland, upper nose, and parts of the forehead) [5].

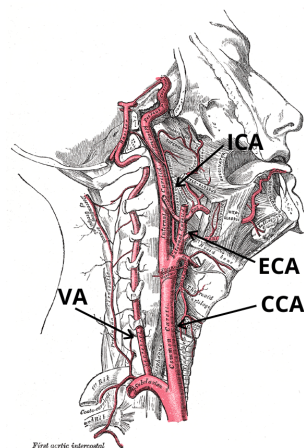


Figure 3.1: Anatomy of arteries in the neck and head—the right side [23] (edited).

Carotid artery stenosis is a disease that can be described as a narrowing of the carotid artery. This reduction is caused by locally collected plaque on the interior arterial wall. The atheromatous plaque may consist of fat, cholesterol, cellular waste products, calcium, and fibrin. As a result, the blood flow from the heart to the brain is reduced [8]. Thrombus or another part of the atherosclerotic plaque can break off and cause transient ischemic attack (TIA), which is the most common cause of stroke. Stenosis is common in the population. Some researchers suggest that more than 5% of the population older than 65 years have asymptomatic stenosis, with at least 50% of artery clogged by plaque [16]. This disease develops for years and might be unnoticed for a long time. The patients are often diagnosed with CAS after the first mini-stroke. The symptoms of stroke and TIA include numbness or weakness, trouble speaking, trouble seeing, dizziness, and severe headache. These problems occur suddenly since the freed parts of plaque travel quickly in the artery [8]. The risk factors that can contribute to the development of carotid atherosclerosis are older age, hyperlipidemia, hypertension, smoking, diabetes, obesity, and sedentary lifestyle [56]. For asymptomatic cases of stenosis, an intensive medicament treatment is most suitable. It includes lowering cholesterol in the blood, treating hypertension, and diabetes screening. This should be combined with healthy lifestyle choices as regular aerobic exercise, a low-fat diet, and smoking cessation [45]. In more severe cases, surgery is necessary. The less invasive option is angioplasty with stenting. During this procedure, a catheter is pushed through the narrowed area. Then a balloon is inflated, widening the space in the artery. Afterward, a stent is placed to keep the artery open. The stent is a plastic or steel tube; see Figure 3.2. During this procedure, some parts of the plaque might get free, so a small filter on the guidewire is placed in the artery [78]. If at least 70% of the artery is blocked, a more invasive method might be inevitable. During a carotid endarterectomy, the artery is opened, and the plaque is surgically removed. After the artery is stitched back together, the flow of the blood is restored. This procedure is done under general or local anesthesia [71].

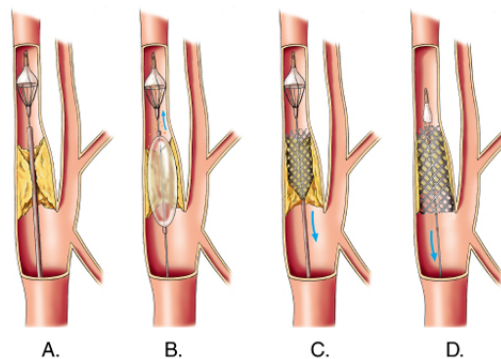


Figure 3.2: When performing carotid stenting, a catheter with a filter is deployed (A.), then the plaque is flattened by a balloon (B.). A stent is placed to keep the artery open (C. and D.) [30].

■ 3.1.1 Diagnosis

During a physical examination, the doctor might listen to the arteries by a stethoscope. Reduction of blood flow creates an abnormal whooshing sound. In medicine, this condition is called a bruit. A practitioner might suggest a test for carotid stenosis based on the patient's medical history, examination, or having some of the symptoms [86]. There are multiple techniques used in image diagnosis of CAS. The most common one is the ultrasound. It produces high-frequency sound waves above the threshold of human hearing. During the procedure, a probe is placed on the skin covered by gel. The probe not only emits the waves but also detect echoes reflected back. A special ultrasound technique is a Doppler ultrasound, which uses the Doppler effect to see and track the movement of blood cells in an artery. Medical ultrasound is noninvasive, safe, painless, and does not produce any ionizing radiation (which is produced by an x-ray) [57, 62]. Another method used is Carotid Angiography. It is an x-ray of arteries and veins. Before this procedure, a contrast dye needs to be injected [72, 77].

Chapter 4

Existing methods

4.1 Image Classification

Image classification is one of the primary tasks in the field of image processing. Its goal is to assign to an image one of the predefined categories. The neural networks have achieved a breakthrough in this field, namely the ones using convolutional layers. Later, as in many other domains, deep learning has become state of the art in this field. One of the benchmarks for this task is the ImageNet Large Scale Visual Recognition Challenge [68], which has begun in 2010. The task is to create a network able to classify over 1.4 million images into one thousand categories. The size of the annotated dataset with the reduction of training time achieved by using GPU led to deep architectures [43]. After the initial successes of deep convolutional neural networks, they have been widely used and applied in many fields, including medical and biological image processing. For example, to predict breast cancer based on histopathological images [76], to classify lung pattern for interstitial lung diseases [2], or to detect and classify abnormalities on frontal chest radiographs [84].

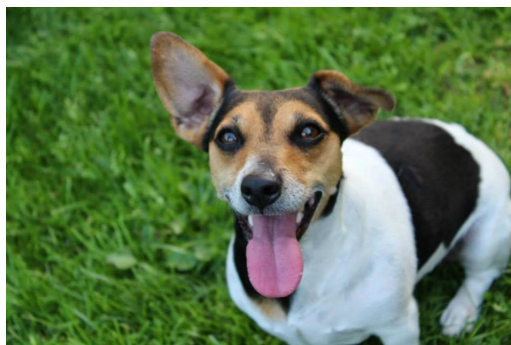


Figure 4.1: An image that would be labeled as a category “dog”.

4.1.1 VGG

Very deep convolutional neural networks for large-scale image recognition [74] were introduced in 2014. They achieved both, first and second places in the Classification tracks of ImageNet Challenge [79] in the same year.

Architecture

The original paper [74] proposed six different VGG architectures, each containing six blocks of convolutional layers separated by max-pooling ones. In the convolutional layers were used filters with size 3×3 (in one experiment were used filters with size 1×1 at the end of three convolutional blocks). The spatial dimensionality is preserved through the whole block by stride 1 and padding. The max-pooling layer reduces dimensionality by half. This is achieved by receptor field with size 2×2 and stride equal to 2. Finally, there are three fully connected layers; the first two with 4096 neurons and the last one with 1000 neurons, followed by a sigmoid activation function [74]. The two best performing models with 16, respectively 19 layers are described in Table 4.1.

VGG-16	VGG-19
Input: $224 \times 224 \times 3$	
$2 \times \text{convl3-64}$	
max-pooling2, stride 2	
$2 \times \text{convl3-128}$	
max-pooling2, stride 2	
$3 \times \text{convl3-256}$	$4 \times \text{convl3-256}$
max-pooling2, stride 2	
$3 \times \text{convl3-512}$	$4 \times \text{convl3-512}$
max-pooling2, stride 2	
$3 \times \text{convl3-512}$	$4 \times \text{convl3-512}$
max-pooling2, stride 2	
FC-4096	
FC-4096	
FC-1000	
soft-max	

Table 4.1: Comparison of 16 and 19 layers VGG architectures [74].

4.1.2 ResNet

ResNet [29], a deep residual convolutional network, was proposed in 2015. The depth of the network was pushed even further, up to 152 layers. This combination of residual learning and network's depth resulted in first place

in the Categorization track of ImageNet Challenge 2015 [85] (ResNet models can be found under MSRA team name).

Residual learning

Deep neural networks are generally harder to train [21]. ResNet targeted this problem by introducing skip-connections. The layers through the networks are not only connected with the preceding ones, but there are connections that skip the layers as well. These shortcuts help to train deep networks. They are based on the assumption that a network with these connections should be able to fit the data as well as the shallower network without them. Moreover, such a design solves the problem of the vanishing gradient. The connections forward the flow in the network, where it is added to the values transformed by multiple layers. This can be viewed in Figure 4.2, which can be written as $y = F(x, W_i) + W_s x$. In this equation F , denotes transformation by multiple layers, and W_s is either identity mapping or a linear projection if the dimension is reduced by F [29].

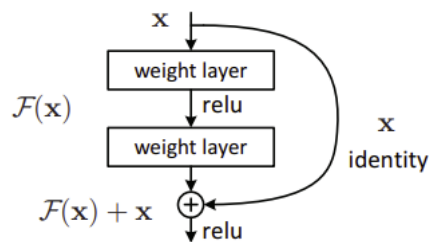


Figure 4.2: The building block of residual learning [29].

Architecture

The architecture of ResNet follows principles introduced in VGG and uses mostly convolutional layers with 3×3 filters, in some versions combined with 1×1 filters. ResNet takes an input of 224×224 pixels, which can be translated into $224 \times 224 \times 3$ matrix. This input is then processed by a convolutional layer with filter size 7×7 and stride 2, which results in the reduction of the dimension to half of the input size— 112×112 . The output of the first layer is fed into the max-pooling layer, with receptor filed 3×3 and stride 2. The following convolutional part is composed of four blocks of convolutional layers, which structure varies with the specific network’s version. The dimensionality between convolutional blocks is reduced by increasing stride to 2 in the first convolutional layer of each block, instead of using max-pooling, which is used in VGG. The result of convolutions is processed by a global average pooling layer, which computes the average of each feature map. The network contains only one fully connected layer, which is at the end, and it is followed by

the soft-max activation function, which translates the output of neurons to probabilities of the one thousand categories. Table 4.2 describes the two most successful architectures with 50, and 101 layers [29].

ResNet-50	ResNet-101
Input: $224 \times 224 \times 3$	
conv7-64, stride 2	
max-pooling3, stride 2	
$3 \times \begin{bmatrix} \text{conv1-64} \\ \text{conv3-64} \\ \text{conv1-256} \end{bmatrix}$	
$4 \times \begin{bmatrix} \text{conv1-128} \\ \text{conv3-128} \\ \text{conv1-512} \end{bmatrix}$	
$6 \times \begin{bmatrix} \text{conv1-256} \\ \text{conv3-256} \\ \text{conv1-1024} \end{bmatrix}$	$23 \times \begin{bmatrix} \text{conv1-256} \\ \text{conv3-256} \\ \text{conv1-1024} \end{bmatrix}$
$3 \times \begin{bmatrix} \text{conv1-512} \\ \text{conv3-512} \\ \text{conv1-2048} \end{bmatrix}$	
global average pooling	
FC-1000	
soft-max	

Table 4.2: Comparison of ResNet50 and ResNet101 architectures [29].

4.2 Object Localization

The goal of object localization is to select an area with a certain object in an image. Usually, by surrounding its borders with a rectangle (bounding box), see Figure 4.3 [60]. It is a simplification of a more complex task—object detection, whose goal is to detect all objects of proposed categories in an image. It has been applied in robot vision, security, autonomous driving, human-computer interaction, intelligent video surveillance, augmented reality, and more [48]. In the field of medical imaging, deep learning can be used to

localize and identify vertebrae in CT images [6], localize ventricle in cardiac MRI images [13], or detect lung nodules in CT scans [75].

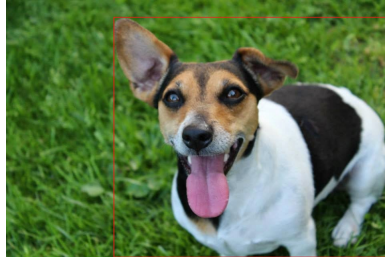


Figure 4.3: Bounding box localizing the object—a dog.

Region Proposal Networks

The objective of Region Proposal Network (RPN) is to generate object proposals, which could be processed by Fast R-CNN. An image is first processed by a set of convolutional and pooling layers, which results in a convolutional feature map. The RPN slides a small window with shape $n \times n$ over this feature map and reduces its dimensionality (convolutional layer with receptor field of size $n \times n$ and number of filters equal to reduced dimension). This is followed by two sibling 1×1 convolutional layers, one for classification and one for regression. At each position, multiple anchors are generated. RPN aims only to distinguish between object and background in the image, so it does not consider object categories in the classification. At every position, multiple (k) proposals are considered, so the classification layer has $2k$ neurons (two categories for each proposal), and the regression one computes $4k$ values (one bounding box per proposal). Each of these predictions is relative to an anchor—reference box with a fixed size. All anchors are centered in the center of the sliding window and the original version uses 3 size ratios in width and height, which creates 9 different anchors (Figure 4.4) [64].

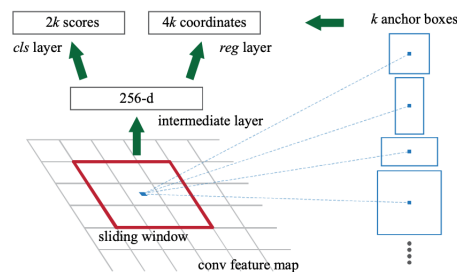


Figure 4.4: The architecture of the region proposal network [64].

Fast R-CNN

Fast R-CNN is a deep convolutional neural network designed to process regions of interest (RoI). As an input, it takes the whole image and processes it by a set of convolutional and pooling layers. This feature map is common for all proposals suggested for a given image, which speeds up the processing time. One region of interest is selected from the convolutional feature map and is then resized into a prespecified shape by max-pooling. The resized region can be easily fed as an input into fully connected layers. These are followed by two sibling branches. One is used to predict the probabilities of $k+1$ classes and, the second one to predict the bounding boxes of objects of k classes [18]. The whole architecture can be seen in Figure 4.5.

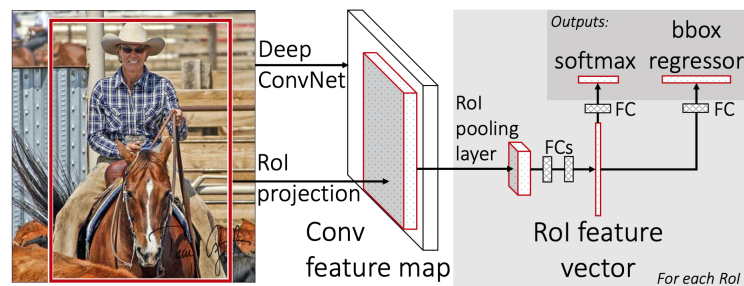


Figure 4.5: The architecture of Fast R-CNN [18].

4.2.1 Faster R-CNN

Faster R-CNN (R stands for “Region”) [64] was published in 2016, and it outclassed the best models at that time on Pascal 2007, Pascal 2012 [15] and, COCO dataset [46]. Object detection is a more complex problem than object localization or image classification, and thus it needs a more complex approach. Previous approaches were composed of multiple models that needed to be trained separately [28, 19, 18]. Faster R-CNN is based on its ancestor—Fast R-CNN [18] enriched by a Region Proposal Network (RPN), both of them trainable in a single stage. RPN proposes regions in an image with suggested positions of objects, and then the detection part (Fast R-CNN) locates an object in the region (Figure 4.6) [64].

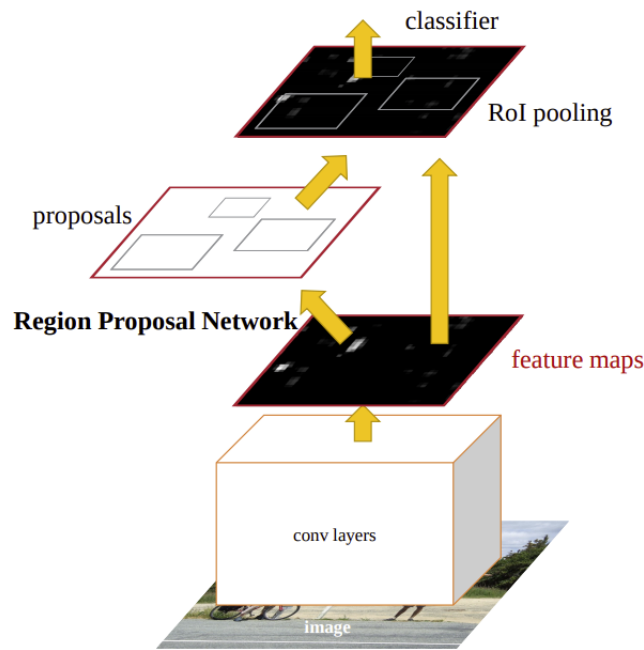


Figure 4.6: The architecture of Faster R-CNN [64].

4.3 Segmentation

Image segmentation is a task that assigns an object class label to each pixel of an image or can be viewed as a process of dividing an image into multiple regions. By segmentation, an object can be localized, and furthermore, we can detect its shape, borders, and relative size. The rise of deep learning brought many new approaches to this field [17]. The human body contains organs that have regular shapes that can be easily spotted. For example, the heart has an oval shape, which is wider at the top. However, there are structures and tissues with inhomogeneous shapes that can be hard to recognize even for an expert. Using image segmentation in computer-aided diagnosis, a medical practitioner may take advantage of automatically processed images, or it can help in massive screenings to process big amounts of collected data. Examples of image segmentation in medical imaging include lung segmentation of volumetric CT images [31], heart segmentation in 3D images [93], or segmentation of the brain in MRI scans [3].



Figure 4.7: Segmentation of the dog in the image.

■ 4.3.1 U-net

U-net [66] is a fully convolutional neural network, which was created in 2015. This new “U”-shaped net has achieved much success in the segmentation of biological images. The authors claim that U-Net is substantially faster and more accurate than competing methods—indeed, it outperformed the runner-up algorithm in the 2015 ISBI cell-tracking challenge [7]. This architecture has been a keystone for many new approaches in image segmentation [1, 54] and has been used even in areas outside biological imaging [91, 52].

■ Architecture

U-net is composed of two opposing arms, both of them built from four levels of convolutional blocks (Figure 4.8). Each block contains two convolutional layers. In the contracting part (the left arm), the number of filters is increased in every block, and the dimensionality between the levels is reduced by max-pooling. Symmetrically, in the expanding path (the right arm), the number of filters is decreasing, and the dimensionality is increased with the up-convolution. Moreover, the net contains residual connections between convolutional blocks on the same levels. The output from the left level is concatenated with the input of the right level. In the convolutional layers are used filters with size 3×3 and stride one. In the proposed version, padding is not used, thus the size is reduced by every convolutional layer by 1 for height and width. Due to this, the dimension of output (segmentation mask) is smaller than the input. The last layer contains k filters, where k represents a number of classes to segment. This is followed by a pixel wise soft-max activation function [66].

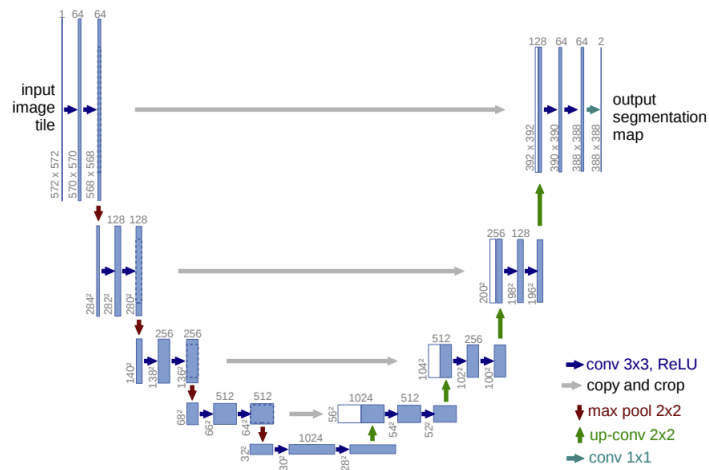


Figure 4.8: The architecture of U-net. There are two arms connected with residual connections. The left one reduces the dimensionality, and the opposite arm increases it almost to the input size [66].

Chapter 5

Data

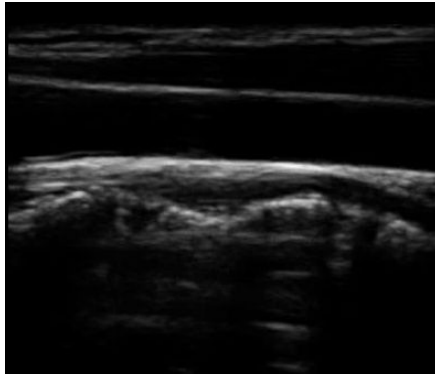
The data are an essential part of machine learning. Although they are present in almost every aspect of human lives, creating a dataset suitable for more complex tasks might still be difficult. In the field of medical imaging, a doctor with a specialized machine is needed in order to examine a patient. Such data themselves are not suitable for the image processing tasks directly; they need to be properly annotated and transformed into a dataset. The annotations vary in difficulty, and in many cases, experienced professionals are required. This chapter discusses two image databases used in this work. The primary one is the ANTIQUE dataset (Section 5.1), and the best from proposed neural networks will be used on these data. To improve the performance, a SPLab dataset (Section 5.2) was used in some of the experiments.

5.1 ANTIQUE dataset

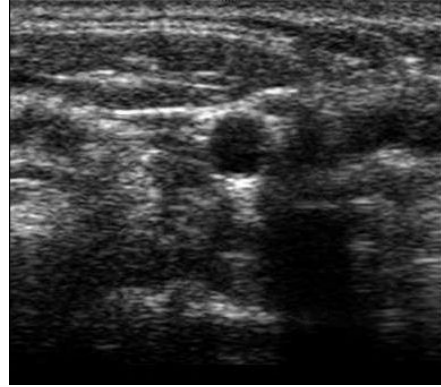
The ANTIQUE dataset was created during the study “Atherosclerotic Plaque Characteristics Associated With a Progression Rate of the Plaque in Carotids and a Risk of Stroke” [96], between 2015 and 2020. A group of 413 patients was selected and observed at the University Hospital Ostrava and Military University Hospital in Prague. The examined patients were between 30 and 90 years old, and all of them were diagnosed with stenosis $> 30\%$. The ultrasound scans of atherosclerotic plaque in the carotid bifurcation and ICA have sufficient image quality. Clinical examination was repeated every six months for three years, and it consisted of physical and neurological examinations, and examinations of carotid arteries by duplex sonography. The dataset in the raw form consists of the images taken in a single examination of a patient. Overall, there are 1,322 examinations available, together containing 28,178 ultrasound scans. There are no annotations regarding how the image was made (orientation of the ultrasound probe, Doppler ultrasound, etc.), nor the position of an artery or the severeness of the stenosis. A raw image does not contain only an ultrasound scan, but some additional information irrelevant for this work (Figure 6.1). Thus only the scan area is used.

5.2 SPLab dataset

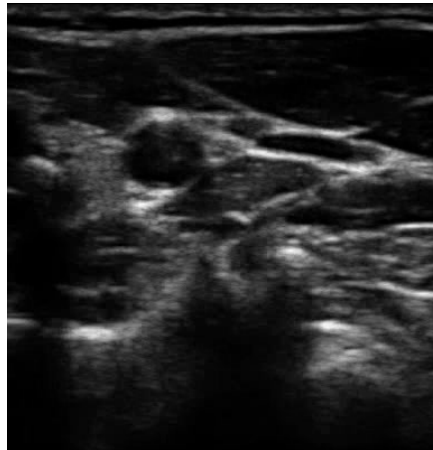
Two databases were used to enlarge the sizes of the annotated data, *the Artery database*, and *the Ultrasound image database* from the Signal processing laboratory at the Brno University of Technology [9]. *The Artery database* contains ultrasound images of the CCA transverse section. It is composed of two sets, each taken by a machine from a different ultrasound manufacturer. The first set was created by an Ultrasonic device, and it contains 849 images. The second set was taken by a Toshiba device, and it consists of 433 images, which are noisier [65]. Samples from both devices can be seen in Figure 5.1. *The Artery database* has been used exhaustively in the research at the BUT [70, 4, 95]. *The Ultrasound image database* contains 84 images of the CCA in the longitudinal section. This database was created by a Sonix OP ultrasound scanner [94].



(a) : SPLab longitudinal image



(b) : SPLab Toshiba transverse image



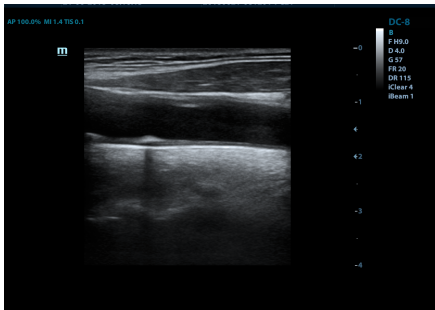
(c) : SPLab Ultrasonic transverse image

Figure 5.1: Examples of images from the SPLab dataset.

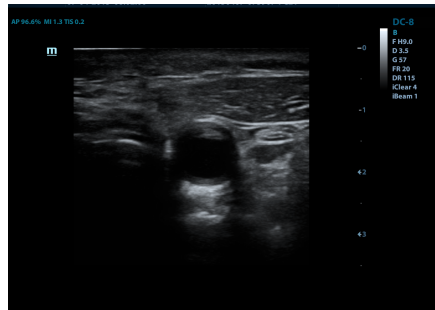
Chapter 6

Classification of ultrasound carotid artery images

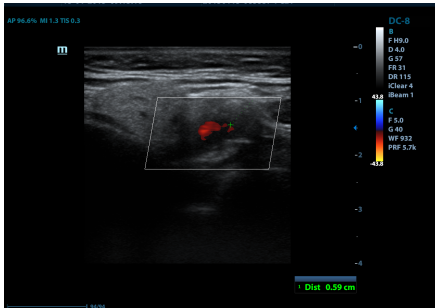
The target dataset contains patient's images from a single examination, and those need to be categorized to be processed further. The ultrasound images classify into four main categories—longitudinal, transverse, conical, and Doppler (Figure 6.1). For this, an annotated data set had to be created.



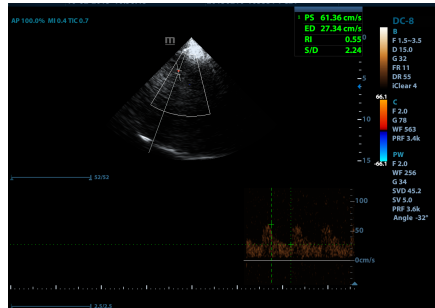
(a) : ANTIQUE longitudinal image



(b) : ANTIQUE transverse image



(c) : ANTIQUE Doppler image



(d) : ANTIQUE conical image

Figure 6.1: Examples of different categories in the ANTIQUE dataset.

6.1 Dataset

The annotations for the ANTIQUE dataset had to be created to train the neural network. The data was captured in sequences, and images from the same angle might appear similar. If such cases were present across the training, validation, or test set, it might have resulted in overfitting. Based on this assumption, files from one examination were sorted into either test, training, or validation group. The distribution of classes in each set follows the distribution of raw data. In some cases, the transverse images strongly remind the longitudinal one, especially when they show the part where CCA bifurcates into ECA and ICA, as can be seen in Figure 6.2. Thus the selection of examination records is not purely random but synthetically enlarged by such problematic samples. Overall, 1679 images from the ANTIQUE dataset were sorted into four categories (transverse, longitudinal, Doppler, conical) and three sets (training, validation, test), described in Table 6.1. In some of the experiments, the transverse and longitudinal classes in the training set were combined with SPLab data, which are already sorted. The training set without the SPLab database will be denoted as *Training set 1* and the training set with SPLab database as *Training set 2*.

Image class	<i>Training set 1</i>	<i>Training set 2</i>	Validation set	Test set
Longitudinal	263	347	100	119
Transverse	514	1728	144	306
Conical	80	80	30	54
Doppler	64	64	36	35

Table 6.1: The number of images in both training, validation, and test set.

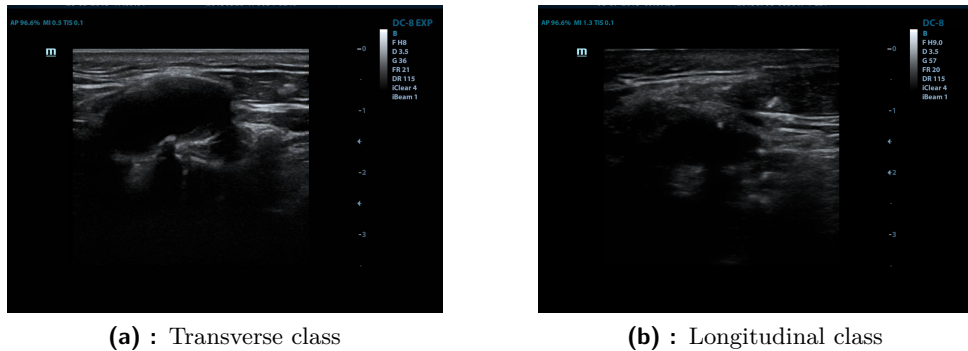
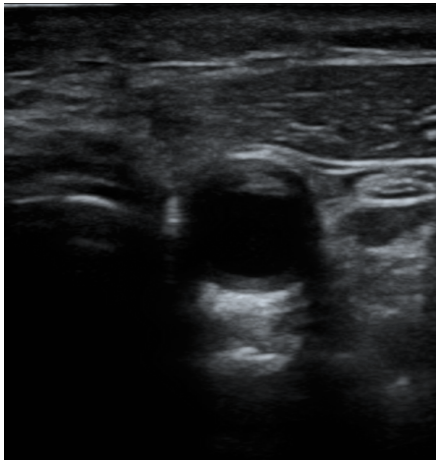


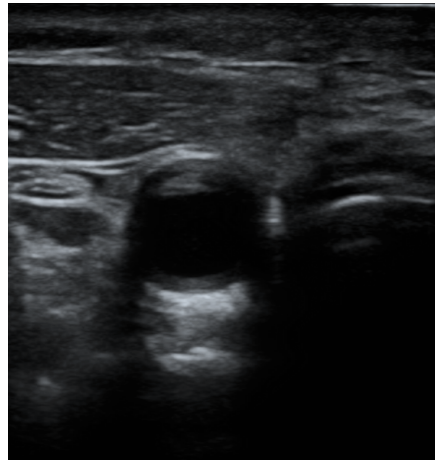
Figure 6.2: Left image shows a carotid bifurcation (transverse class) and the right one a longitudinal image.

6.1.1 Data augmentation

Every image needs to be processed when used in machine learning. The necessary set of training transformations consists of resizing to the size predefined by the particular architecture and normalizing values to the 0–1 range. This combination will be denoted as *Simple transformation*. Data augmentation is an easy way how to create robust models and artificially create bigger datasets. A simple example can be seen in Figure 6.3, where the transverse section image would be categorized the same, regardless of how flipped it is. Complex data transformation will be denoted as *Complex transformation*, and it is described in Table 6.2, together with *the Simple transformation*.



(a) : Original image



(b) : Horizontal flip



(c) : Vertical flip

Figure 6.3: Examples of image transformations used in classification.

<i>Simple transformation</i>	<i>Complex transformation</i>
Resize Normalize	Resize Normalize Random Horizontal flip, $p = 0.5$ Random Vertical flip, $p = 0.5$

Table 6.2: Transformations used to augment the training set.

6.2 CNN Architectures

Three different architectures were compared, from the relatively small one to the deep VGG-16 with over 130 million trainable parameters. The simplest from the proposed networks had 82,000 times fewer parameters than VGG-16 and 14,000 times less than ResNet50 (Table 6.3).

Model	Number of trainable parameters
Small CNN	1,628
VGG-16	134.2 millions
ResNet50	23.5 millions

Table 6.3: Comparison of the number of trainable parameters of the classification models.

6.2.1 Small CNN

A small convolutional net was created as a baseline. It consists of five layers—two convolutional, two max-pooling, and one fully connected. This network, with a relatively small number of learnable parameters, takes an input with a small resolution— 28×28 pixels. Afterward, a convolutional layer with 4 filters and 5×5 kernels is used. Dimensionality is halved by a max-pooling layer with receptor field 2×2 and stride 2. Followed by another block composed of the convolutional layer, with a number of filters increased to 8 and a max-pooling layer. Convolutional layers do not use padding, thus every application reduces the dimension by 2 from every side. The last, fully connected layer contains four neurons. The output of this layer can be translated into probabilities by a soft-max activation function.

Small CNN
Input: $28 \times 28 \times 3$
conv15-4
max-pooling2, stride 2
conv15-8
max-pooling2, stride 2
FC-4
soft-max

Table 6.4: The architecture of Small CNN.

6.2.2 VGG-16

Several VGG architectures were proposed. The sixteen layers version was selected; its performance was not significantly worse than VGG-19 on the ImageNet dataset, but contained 6 million fewer parameters than the deeper version [74]. The VGG-16 was used with weights pretrained on the ImageNet dataset, and only the last fully connected layer was removed and substituted with a newly initialized one containing 4 neurons. Since the goal is to train on ultrasound images, which are very different from those in the ImageNet, all layers are fine-tuned.

6.2.3 ResNet50

The following selected architecture is ResNet, which has surpassed VGG on multiple classification tasks with five times fewer parameters [29]. As in the previous case, the deepest architecture from the initially proposed ones was not used. In the tradeoff between performance and size, the ResNet50 was chosen. The model was pretrained on the ImageNet dataset, and the last and only fully connected layer was replaced with a new one containing 4 neurons.

6.2.4 Training

Transfer learning has shown to improve and speed up the training of deep neural networks [83]. The use of weights that are pretrained on a different dataset (for example, Image Net) has become a standard practice in computer vision [32, 73]. The weights that have not been pretrained are initialized by He initialization [26]. Since we were dealing with classification, a cross-entropy loss function was used. All of the models were trained by stochastic gradient descent with Nesterov momentum [20]. During this process, all the weights were adjusted. The momentum was set to 0.95, and the learning rate started at 10^{-4} . The learning rate was decayed by a multiplicative factor equal to 0.1, when the training loss did not significantly improve for 3 epochs. The

whole training lasted for 30 epochs, and the model with the lowest validation loss was selected.

6.3 Experiments and results

All of the proposed architectures were trained on both datasets, each time with a different set of transformations. Overall, each model was trained four times. Table 6.5 contains the lowest training and validation losses, as well as the percentage of accuracy, which gives a more straightforward description of how the model performs. As expected, the worst train and validation results had Small CNN. The combination which gave the best validation loss was *Train set 1* and *Complex transformation*. The Small CNN gave worse results when the SPLab data enlarged the ANTIQUE dataset. Such a small model was not able to generalize and learn from images taken by different machines. That changed when it came to deeper architectures, such as VGG-16. Both transformations achieved better results when using *Train set 2*. This training set combined with *Simple transformation* achieved the best validation results—0.08103 loss and accuracy 97.419%. The model expected to provide the best results was ResNet50. It was able to converge to train accuracy 100% in three out of four cases. Nevertheless, this did not reflect in validation metrics by overfitting. Validation losses overcame VGG-16 in every setting.

Small CNN					
Data	Transformations	Tr. loss	Tr. accuracy	Val. loss	Val. accuracy
<i>Train set 1</i>	<i>Simple tr.</i>	0.16071	95.005%	0.77715	79.355%
<i>Train set 1</i>	<i>Complex tr.</i>	0.36838	85.993%	0.77499	72.581%
<i>Train set 2</i>	<i>Simple tr.</i>	0.03718	98.828%	0.87961	72.903%
<i>Train set 2</i>	<i>Complex tr.</i>	0.14238	94.953%	0.82445	75.806%
VGG-16					
Data	Transformations	Tr. loss	Tr. accuracy	Val. loss	Val. accuracy
<i>Train set 1</i>	<i>Simple tr.</i>	0.00154	100%	0.08250	96.774%
<i>Train set 1</i>	<i>Complex tr.</i>	0.02608	99.240%	0.17101	93.548%
<i>Train set 2</i>	<i>Simple tr.</i>	0.00046	100%	0.08103	97.419%
<i>Train set 2</i>	<i>Complex tr.</i>	0.01316	99.504%	0.11784	94.839%
ResNet50					
Data	Transformations	Tr. loss	Tr. accuracy	Val. loss	Val. accuracy
<i>Train set 1</i>	<i>Simple tr.</i>	0.00064	100%	0.06046	98.710%
<i>Train set 1</i>	<i>Complex tr.</i>	0.00352	99.891%	0.05755	97.419%
<i>Train set 2</i>	<i>Simple tr.</i>	0.00053	100%	0.07633	97.097%
<i>Train set 2</i>	<i>Complex tr.</i>	0.00023	100%	0.04064	98.710%

Table 6.5: The best training and validation losses of classification models. The best validation loss for every architecture is highlighted.

Every model was evaluated on the test set in order to select the best one, see Table 6.6. These results mostly copied the validation one. The ResNet50 trained on *the Train set 2* with *Complex transformation* achieved the best test results from all of the experiments. The test loss of this net was 0.01342, with an accuracy of 99.222%. It made only four mistakes.

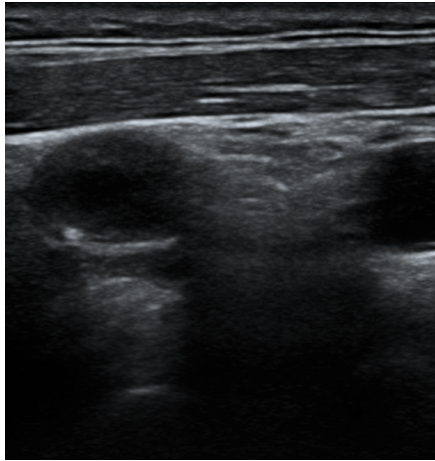
Small CNN			
Data	Transformations	Test loss	Test accuracy
<i>Train set 1</i>	<i>Simple tr.</i>	0.47997	82.101%
<i>Train set 1</i>	<i>Complex tr.</i>	0.52131	79.961%
<i>Train set 2</i>	<i>Simple tr.</i>	0.59333	77.626%
<i>Train set 2</i>	<i>Complex tr.</i>	0.40255	85.019%
VGG-16			
Data	Transformations	Test loss	Test accuracy
<i>Train set 1</i>	<i>Simple tr.</i>	0.06093	98.638%
<i>Train set 1</i>	<i>Complex tr.</i>	0.07469	96.693%
<i>Train set 2</i>	<i>Simple tr.</i>	0.03246	99.027%
<i>Train set 2</i>	<i>Complex tr.</i>	0.05183	98.638%
ResNet50			
Data	Transformations	Test loss	Test accuracy
<i>Train set 1</i>	<i>Simple tr.</i>	0.09188	98.444%
<i>Train set 1</i>	<i>Complex tr.</i>	0.05930	98.054%
<i>Train set 2</i>	<i>Simple tr.</i>	0.01699	99.222%
<i>Train set 2</i>	<i>Complex tr.</i>	0.01342	99.222%

Table 6.6: The test losses and accuracies of classification models. The lowest test loss for every architecture is highlighted.

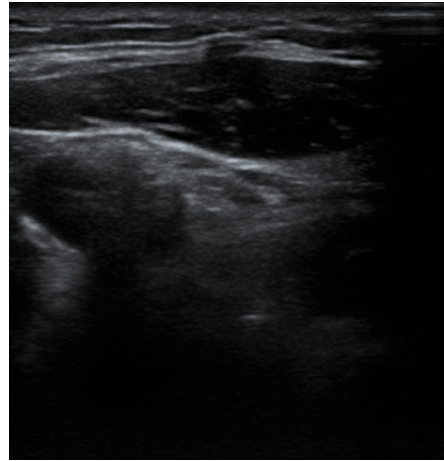
These are described in the confusion matrix shown in Table 6.7. Some of these mistakes were caused by switching transverse and longitudinal classes or vice versa. One time the conical image was classified as a Doppler one. Figure 6.4 shows examples of these mistakes, together with the probabilities predicted by the model for each class.

Predicted class / Ground truth	Longitudinal	Transverse	Conical	Doppler
Longitudinal	117	2	0	0
Transverse	1	305	0	0
Conical	0	0	53	1
Doppler	0	0	0	35

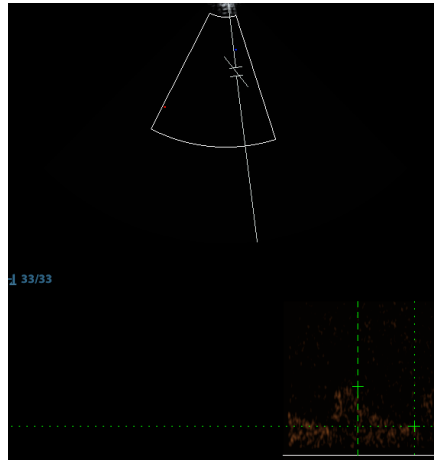
Table 6.7: Mistakes made by the best classification model on the test set.



(a) : Longitudinal image, predicted probabilities of classes: Long 42.3%, **Trans.** 57.7%, Conical 0.0%, Doppler 0.0%

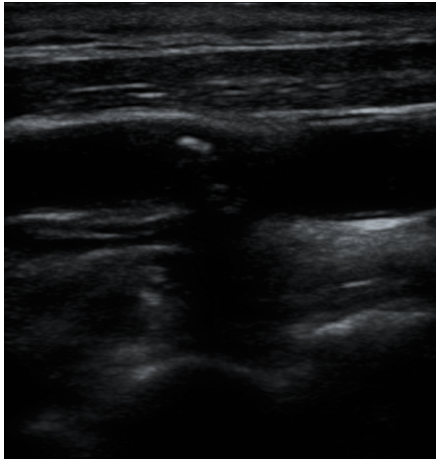


(b) : Transverse image, predicted probabilities of classes: **Long** 86.0%, Trans. 14.0%, Conical 0.0%, Doppler 0.0%

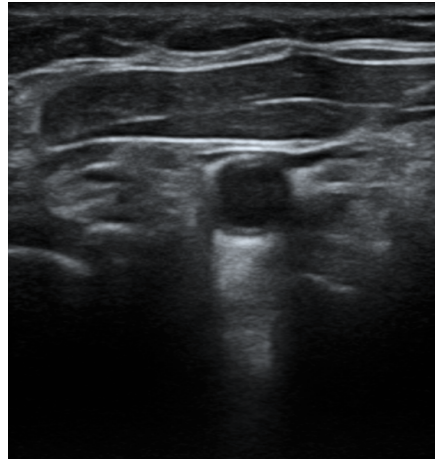


(c) : Conical image, predicted probabilities of classes: Long 0.0%, Trans. 0.0%, Conical 22.2%, **Doppler** 77.8%

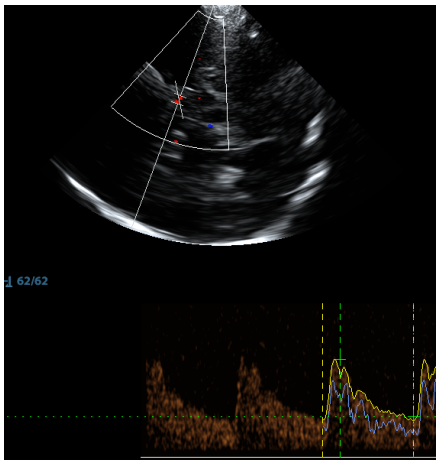
Figure 6.4: Three different mistakes made by the best classification neural network. The probabilities of classes predicted by the network are shown along with the true category.



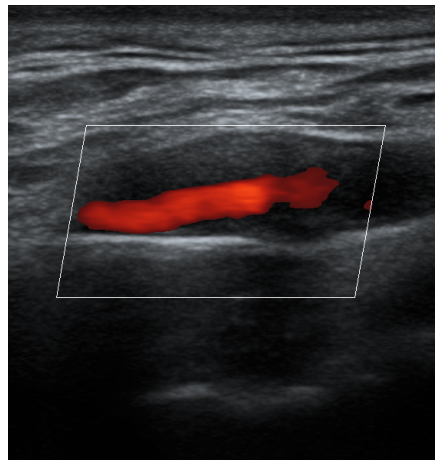
(a) : **Longitudinal image**, predicted probabilities of classes: **Long** 100.0%, Trans. 0.0%, Conical 0.0%, Doppler 0.0%



(b) : **Transverse image**, predicted probabilities of classes: Long 0.0%, **Trans.** 100.0%, Conical 0.0%, Doppler 0.0%



(c) : **Conical image**, predicted probabilities of classes: Long 0.0%, Trans. 0.0%, **Conical** 100.0%, Doppler 0.0%



(d) : **Doppler image**, predicted probabilities of classes: Long 0.0%, Trans. 0.0%, Conical 0.0%, **Doppler** 100.0%

Figure 6.5: Four different images classified correctly by the best classification neural network. The probabilities of classes predicted by the network are shown along with the true category.

Chapter 7

Localization of CCA and ICA in ultrasound images

The area scanned by ultrasound is bigger than the region of interest—the carotid artery. This can be solved by localization. In this work, the goal is to detect CCA or ICA if the image contains both ECA and ICA. ICA is chosen over ECA since stenosis in the external carotid artery may cause more severe damage. A bounding box should surround all parts of an artery—a lumen, a plaque, and a wall. For this purpose were created two annotated datasets (one for transverse and one for longitudinal images). Multiple experiments were proposed in order to maximize the performance of the Faster R-CNN.

7.1 Dataset

Since the original dataset did not contain any information about the location of a carotid, such references needed to be created. Precisely 150 representative examinations were selected from the stable and progressive group, 75 from each. From these was handpicked one transverse and one longitudinal image with good visibility of the artery per patient. As a result, two datasets were created. CCA or ICA was localized on every image by a bounding box (Figure 7.1a). Creating such labels might be particularly difficult, for example, to distinguish ECA from ICA on the transverse section images. These annotations were checked by medical students from the Faculty of Medicine and Dentistry of the Palacký University, who have the corresponding domain knowledge to distinguish the carotid arteries or to correctly recognize the border of an artery wall from the surrounding tissue. These data were divided into three groups—training, validation, and test one (Table 7.1). *Artery database* from the SPLab dataset already contains bounding boxes. Each one localizes a CCA in the transverse section ultrasound image. Two splits were created, training (80%) and validation (20%). The test group of SPLab images was not created because the target dataset was the ANTIQUE one.

Image class	Longitudinal	Transverse
Training set	75	75
<i>SPLab training set</i>	–	972
<i>SPLab validation set</i>	–	242
<i>Validation set</i>	25	25
<i>Test set</i>	50	50

Table 7.1: The number of images used in training and evaluation of the localization models.

7.1.1 Data augmentation

As well as in the previous chapter, all images were normalized to 0–1 range and then standardized with mean and standard deviation of ImageNet dataset (mean= (0.485, 0.456, 0.406), std= (0.229, 0.224, 0.225)). Creating an annotated dataset is not only time consuming, but in this case, it requires knowledge of human anatomy and medical ultrasound. To be maximally efficient with the data, multiple methods for data augmentation were created. In the localization, the bounding box needs to be transformed with the image. Horizontal and vertical flips were used again. Moreover, the Faster R-CNN takes an input of non-fixed shaped images, so a transformation was created that rescaled the image with the label. The lower and upper bound of the scaling ratio was set to 0.8 and 1.2. The main assumption behind this procedure is to make the model more robust to the carotids of different sizes since they can vary in the population. Another augmentation was random cropping. The tissue surrounding the carotid was randomly cropped, which influences the feature map produced by the RPN. Table 7.2 describes *Simple transformation* and *Complex transformation*, which are used during the training in the experiments. Figure 7.1 compares all the mentioned transformations.

<i>Simple transformation</i>	<i>Complex transformation</i>
Normalize Standardization	Normalize Standardization Random Horizontal flip, $p = 0.5$ Random Vertical flip, $p = 0.5$ Random Crop, $p = 0.1$ Random Reshape, $p = 0.25, l = 0.8, u = 1.2$

Table 7.2: The comparison of transformations used in the localization task.

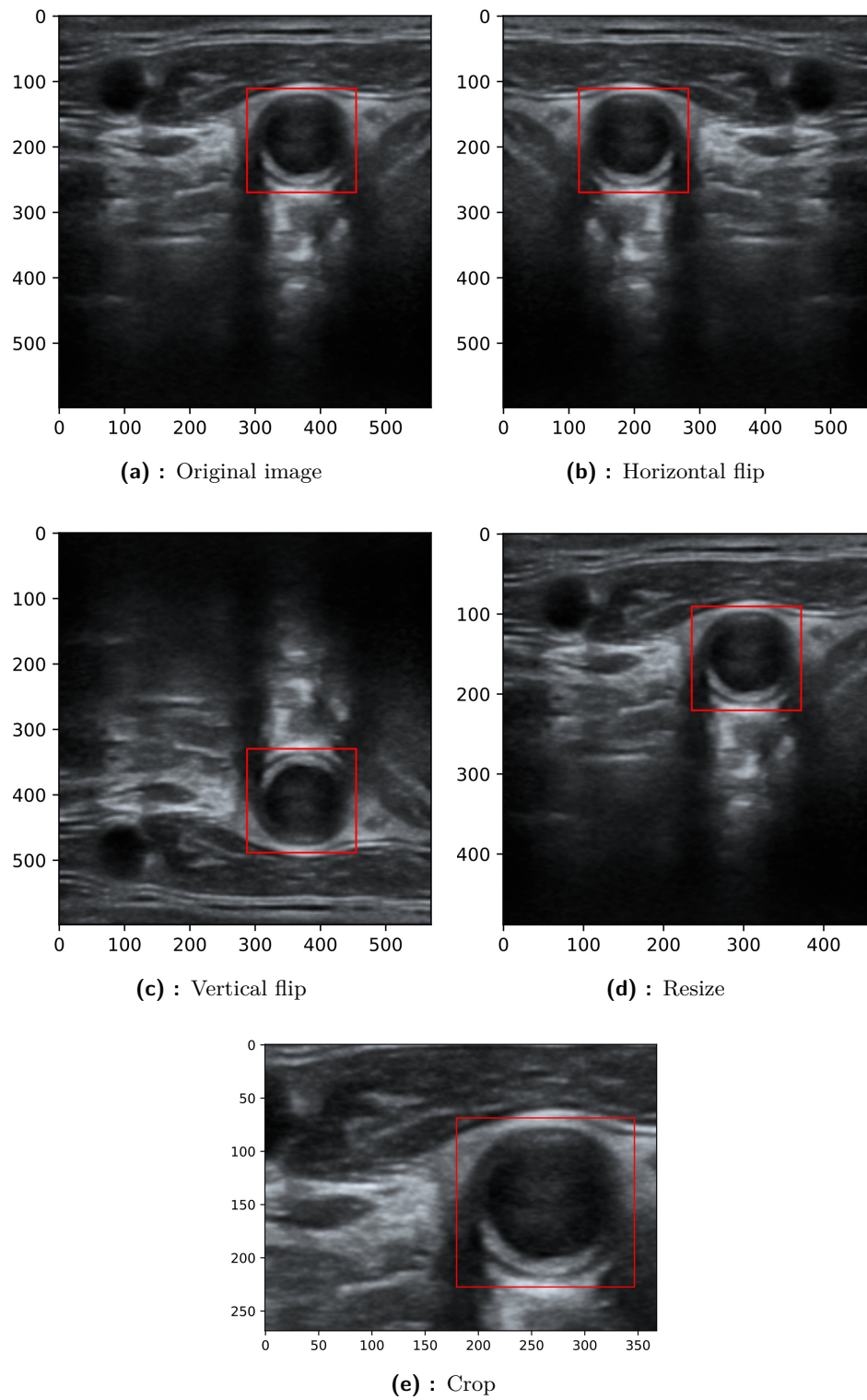


Figure 7.1: Transformations used in localization.

7.2 Faster R-CNN

There are only small adjustments in the originally proposed model. The ResNet architecture was selected as the backbone of the network. This part converts the input to the feature map by multiple convolutional layers. It consists of five convolutional blocks that were pretrained on the ImageNet. The head of the network was newly initialized, and its architecture stayed without a change. During the training, all of the parameters in the architecture were optimized.

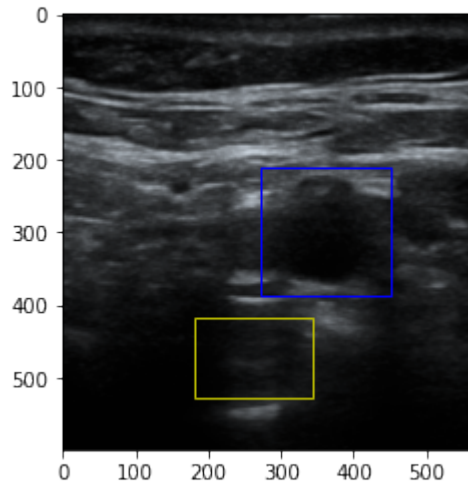


Figure 7.2: Multiple objects detected by transverse Faster R-CNN. The blue bounding box has $p_{carotid} = 0.9957$ and the yellow one $p_{carotid} = 0.0698$. The blue box correctly detects the carotid artery.

7.2.1 Training

In the case of Faster R-CNN, the objective function of the detection network is composed of two metrics—a classification loss and a localization loss. The classification loss (L_{cls}) computes the negative logarithm of the true class probability predicted by the model. The localization loss (L_{loc}) computes the difference between the bounding-box regression targets and the predicted coordinates [18]. The object localization can be evaluated not only in the term of losses, but also in the Intersection over Union (IoU). IoU computes the overlap between true and predicted bounding boxes divided by the union of these two boxes. The best possible score is 1.0, and the worst is 0.0 (Figures 7.4 and 7.5). Since the Faster R-CNN is a network designed for object detection, it can predict multiple boxes for a single category in an image. All of these boxes are paired with a class probability. This can be seen in Figure 7.2. The bounding box with the highest probability was selected, since in every image, there is only one CCA or ICA. To optimize the training

loss was used Adam [38]. The initial learning rate was 10^{-4} , and it was decayed 3 times after preselected epochs. The whole training of a single network took 40 epochs. The Faster R-CNN with the lowest validation loss on the ANTIQUE dataset was selected.

7.3 Experiments and results

A separate Faster R-CNN was developed for each image category. In the case of the transverse Faster R-CNN, the SPLab dataset was used in multiple ways in order to maximize the localization ability. As a baseline, only the ANTIQUE dataset was used during the training. There were no significant differences in the test losses between *Simple* and *Complex transformations*. When both networks were evaluated on the test set by IoU, the model trained with *the Complex transformations* was able to predict 60% of the bounding boxes with IoU bigger than 0.85 (Table 7.3).

ANTIQUÉ data		
Transformations	<i>Simple transformation</i>	<i>Complex transformation</i>
Training L_{cls}	0.00817	0.01288
Training L_{loc}	0.00943	0.02168
Validation L_{cls}	0.00817	0.01395
Validation L_{loc}	0.00943	0.02270
Test L_{cls}	0.02685	0.02331
Test L_{loc}	0.04502	0.04505
Test IoU ≥ 0.6	94%	92%
Test IoU ≥ 0.75	86%	84%
Test IoU ≥ 0.85	48%	60%

Table 7.3: The comparison of two transverse Faster R-CNNs trained on the ANTIQUÉ dataset. Each network was trained with different set of transformations.

SPLab training set later enlarged *the ANTIQUÉ training set*. This step improved test L_{loc} , but other metrics did not show rapid improvement, moreover many of them were even worse (Table 7.4). Taking into account the fact that to the training set was enlarged by 972 samples, this experiment was truly a disappointment.

ANTIQUÉ + SPLab data		
Transformations	<i>Simple transformations</i>	<i>Complex tr.</i>
Training L_{cls}	0.00194	0.00785
Training L_{loc}	0.00158	0.01820
SPLab validation L_{cls}	0.00819	0.00973
SPLab validation L_{loc}	0.02182	0.02960
ANTIQUÉ validation L_{cls}	0.00217	0.01223
ANTIQUÉ validation L_{loc}	0.00213	0.02427
Test L_{cls}	0.03074	0.02675
Test L_{loc}	0.03502	0.04183
Test IoU ≥ 0.6	90%	92%
Test IoU ≥ 0.75	88%	82%
Test IoU ≥ 0.85	64%	54%

Table 7.4: The losses of Faster R-CNNs trained on the combination of the ANTIQUÉ and the SPLab dataset.

The SPLab and the ANTIQUÉ data contain the same type of data, but the images themselves look different. When the datasets were combined, the network was trained to fit the SPLab data, although it will never be used on them. To use the information from the SPLab data, a network was firstly fitted on *the SPLab training set*. These models were able to detect 86% (*Simple transformation*) and 92% (*Complex transformation*) of the test bounding-boxes with IoU higher than 0.6 (Table 7.6), but as the IoU threshold got bigger, the percentage of correctly predicted bounding boxes decreased. Then, the Faster R-CNN with the lowest SPLab validation loss was fine-tuned on *the ANTIQUÉ training set*. Such an approach achieved the best results. The network trained using *the Complex transformation* was the best performing one. From the bounding boxes generated by this Faster R-CNN, 90% had IoU greater than 0.75 with the references. In one of the fifty training samples, the network did not predict any bounding box; this image is shown in Figure 7.3. Thus if an object was found on a test image, the IoU with the ground truth was at least 0.6. The network detected more than one carotid artery in seven cases, and only one object was found in the remaining 42 images (Table 7.5). Figure 7.4 shows four test images with the predicted bounding boxes.

Model	Zero	One	Many
<i>The best transverse Faster R-CNN</i>	1	42	7
<i>The best longitudinal Faster R-CNN</i>	0	33	17

Table 7.5: The number of detected arteries in the test images. The Faster R-CNN either found none, one or many objects classified as an artery.

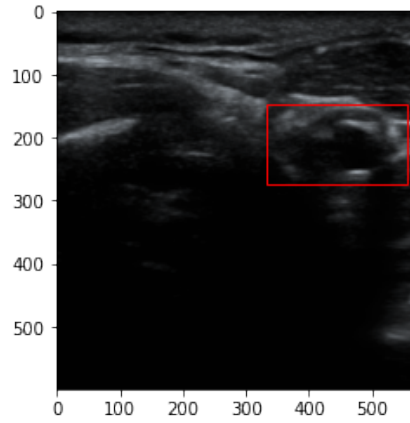


Figure 7.3: The only test sample in which the best transverse Faster R-CNN was not able to classify any region as an artery. The red bounding box shows the true position of the unnoticed artery.

SPLab data		
Transformations	<i>Simple tr.</i>	<i>Complex tr.</i>
SPLab training L_{cls}	0.00232	0.00713
SPLab training L_{loc}	0.00248	0.023814
SPLab validation L_{cls}	0.00589	0.00999
SPLab validation L_{loc}	0.02119	0.03048
Test L_{cls}	0.03346	0.03199
Test L_{loc}	0.03968	0.04961
Test IoU ≥ 0.6	86%	92%
Test IoU ≥ 0.75	64%	84%
Test IoU ≥ 0.85	34%	36%
ANTIQUÉ data		
Transformations	<i>Simple tr.</i>	<i>Complex tr.</i>
Training L_{cls}	0.00343	0.00626
Training L_{loc}	0.00245	0.01060
Validation L_{cls}	0.00328	0.00626
Validation L_{loc}	0.00270	0.01257
Test L_{cls}	0.02667	0.01873
Test L_{loc}	0.03533	0.03253
Test IoU ≥ 0.6	94%	98%
Test IoU ≥ 0.75	84%	90%
Test IoU ≥ 0.85	66%	68%

Table 7.6: The upper part of the Table describes training and evaluation of the Faster R-CNN trained on the SPLab dataset. The lower part holds the data from the fine-tuning on the ANTIQUÉ dataset.

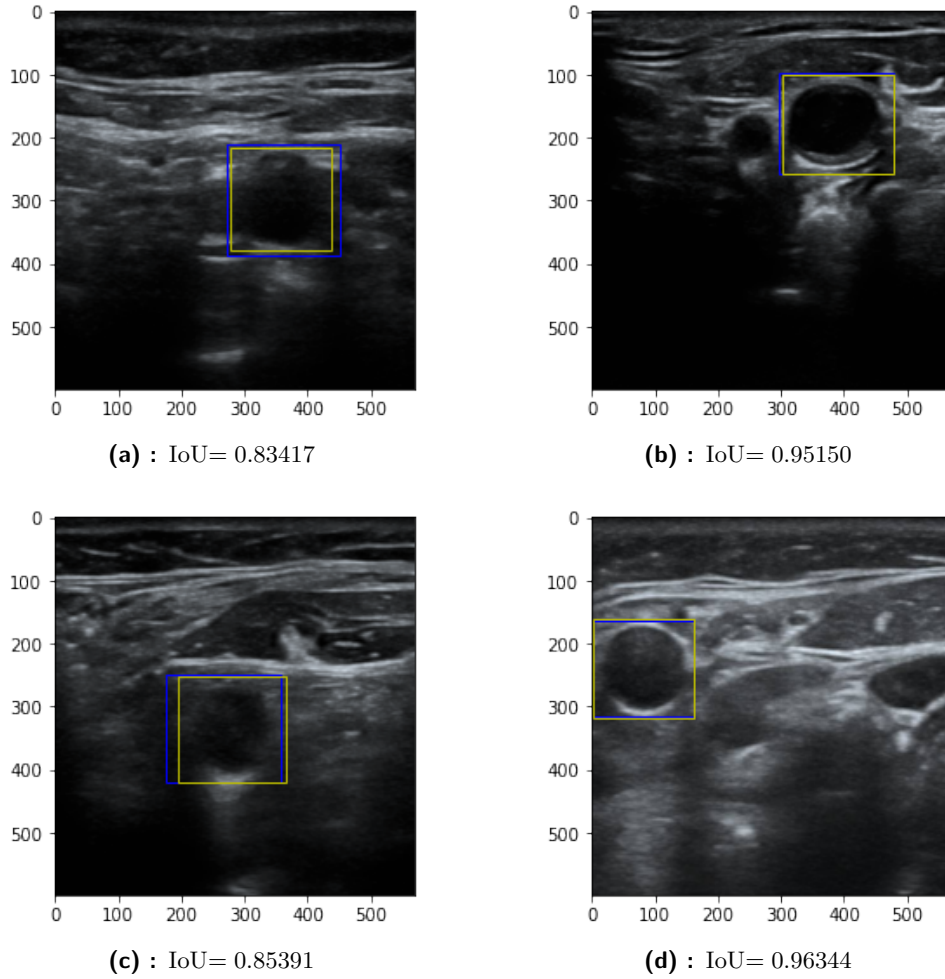


Figure 7.4: The blue boxes were generated by the best transverse Faster R-CNN from the experiments. The yellow bounding boxes are true positions of the carotid arteries.

Only the 150 annotated longitudinal images from the ANTIQUE dataset were available for the training and evaluation of longitudinal Faster R-CNN. Firstly, the newly initialized Faster R-CNN was trained to detect the carotid artery in an image. The network that trained using *Simple transformation* performed better than the one using data augmentation. The training L_{loc} of this neural network was half of the localization loss of the Faster R-CNN trained with *Complex transformation*, and 90% of predicted boxes had IoU greater than 0.75 with the true positions (Table 7.7). Since there are some similarities between the longitudinal and transverse images (both categories contain the same fibres, but from different angles), the best transverse Faster R-CNN was retrained for the localization of the carotid on the longitudinal images. Sadly, this approach did not bring the desired results (Table 7.8). This model achieved comparable results as the newly initialized Faster R-CNN but did not surpass them. The freshly initialized Faster R-CNN, trained with

Newly initialized Faster R-CNN		
Transformations	<i>Simple tr.</i>	<i>Complex tr.</i>
Training L_{cls}	0.00387	0.01138
Training L_{loc}	0.00261	0.01310
Validation L_{cls}	0.00371	0.01226
Validation L_{loc}	0.00291	0.01270
Test L_{cls}	0.01456	0.02370
Test L_{loc}	0.01927	0.03854
Test IoU ≥ 0.6	98%	100%
Test IoU ≥ 0.75	90%	90%
Test IoU ≥ 0.85	62%	60%

Table 7.7: The results of newly initialized Faster R-CNN trained to detect a carotid artery on the longitudinal images.

Pretrained Faster R-CNN		
Transformations	<i>Simple tr.</i>	<i>Complex tr.</i>
Training L_{cls}	0.00323	0.00844
Training L_{loc}	0.00331	0.01589
Validation L_{cls}	0.00354	0.00857
Validation L_{loc}	0.00337	0.01666
Test L_{cls}	0.02047	0.01752
Test L_{loc}	0.02808	0.03293
Test IoU ≥ 0.6	98%	100%
Test IoU ≥ 0.75	84%	88%
Test IoU ≥ 0.85	58%	52%

Table 7.8: The results of pretrained Faster R-CNN trained to detect a carotid artery on the longitudinal images.

Simple transformation, was selected as the best model for this task. Figure 7.5 shows sample predictions (blue bounding box) of this model on the test set. The model was able to detect an object in all of the test samples, but in 34% of the cases, more than one artery was detected (Table 7.5).

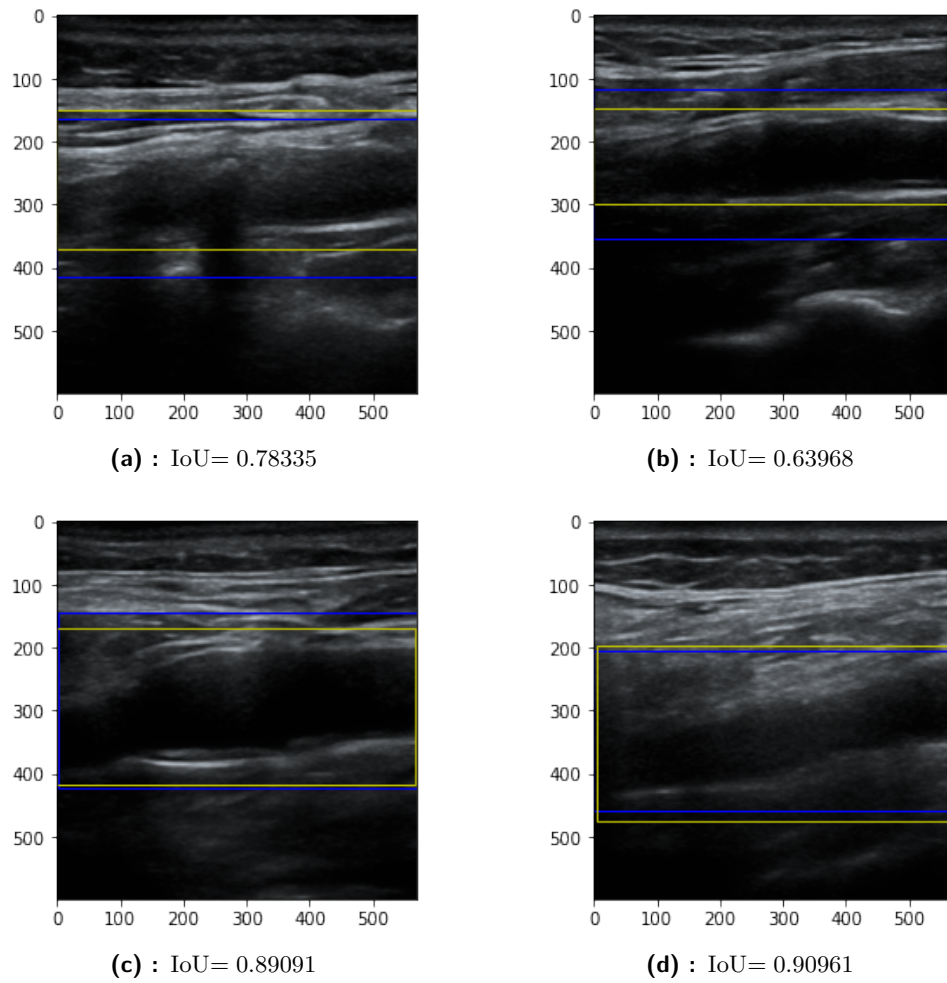


Figure 7.5: The blue boxes were generated by the best longitudinal Faster R-CNN from the experiments. The yellow bounding boxes are true positions of the carotid arteries.

Chapter 8

Segmentation of ultrasound carotid artery images

The image of the artery itself does not give any information about how much is the stenosis developed. For this purpose, the image needs to be segmented into its core parts. Initially, four parts were aimed to be recognized, namely a plaque, a lumen, an artery wall, and a surrounding tissue. By this, the severity of the disease could be measured by the percentage of the lumen blocked by a plaque. However, during the annotation process was found out that separating the artery wall from the plaque is not an easy task. Based on this, another approach was proposed, that the wall with the plaque might be combined into one category. Later the severeness of the carotid artery stenosis could be diagnosed by the thickness of the wall with stenosis. Two models based on the U-net architecture were trained, one for longitudinal and one for transverse images.

8.1 Dataset

Segmenting an image by hand is arguably the most complicated annotation one might face in image recognition. It not only takes much time but requires a high level of focus as well. The medical students annotated the same images as in the localization section. Even with medical education, it was still not easy to decide which parts of a carotid can be labeled as a plaque or specify a precise border of an artery wall. It required an iterative process of consulting and adjusting the annotations. From the original images was selected the rectangular area containing the segmentation of the carotid artery.

Image class	Training set	Validation set	Test set
Longitudinal	75	25	50
Transverse	75	25	50

Table 8.1: Number of samples in each set—training, validation and test one.

8.1.1 Data augmentation

In the case of segmentation, several transformations might be used to process the input. Two of them are horizontal and vertical flips, which were used in the previous sections. The input to the model can be cropped from an original ultrasound by a bounding box predicted by a Faster R-CNN. In the case that a localization model would create the input, it cannot be assumed that the bounding box would be as tight around the artery as the reference created by hand. In the better case, the area would contain surrounding tissue, not only an artery. For this purpose, a transformation was created that adds a random number of pixels from predefined interval $([0, max_add])$ to each side (8.1e). Normalization, Random adding of pixels, Random horizontal and vertical flips were composed to *the Complex transformation* (Table 8.2). In *the Simple transformation*, the input was only normalized to 0–1 range, and 15 pixels were added to each side (Figure 8.1b).

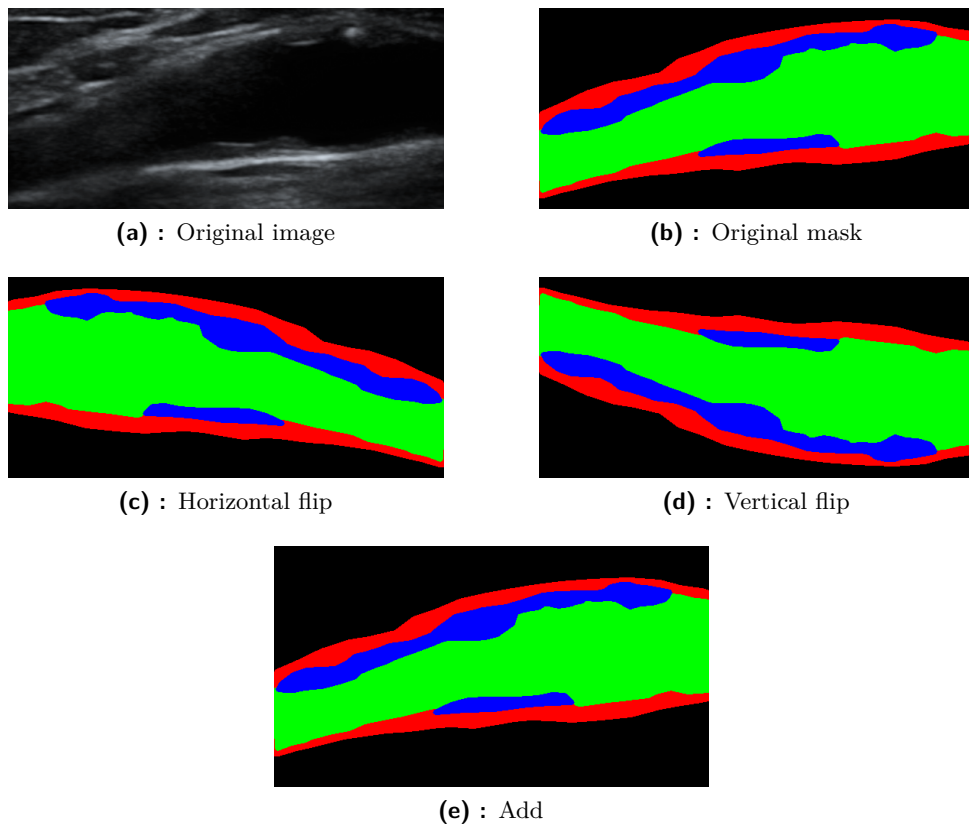


Figure 8.1: Examples of the image transformations used in segmentation. The artery wall is red, the plaque is blue, the lumen is green, and the rest tissue is black.

<i>Simple transformation</i>	<i>Complex transformation</i>
Normalize Add, 15	Normalize Random Add, $max_add = 30$ Random Horizontal flip, $p = 0.5$ Random Vertical flip, $p = 0.5$

Table 8.2: The comparison of transformations used in the segmentation task.

8.2 U-net

The model used in this chapter contain multiple changes compared to the original architecture of U-net [66]. In the original paper, the input and the output did not have the same size. This was caused by using convolutional layers without padding, which reduces dimension after every pass through. As a result, the forwarded outputs from the left arm of the network needed to be cropped. With the usage of padding, the input shape needs to be adjusted because of the pooling layers. The selected input shape was chosen 512×512 . Such shape can be nicely reduced by a factor of 2 by the max-pooling layers in the left part of the network, respectively, upsampled in the right arm. The upsampling is done by bilinear rescaling, followed by a convolutional layer that reduces the number of the features by half. Other adjustments were made in the convolutional blocks used through the network. ReLU activation functions were replaced by PReLU, which was shown to improve the fitting of a model [27]. Batch normalization was used after every second convolutional layer in order to stabilize the training process [33]. The convolutional block is described in detail in Table 8.3.

U-net Convolutional block
conv3
PReLU
conv3
BatchNorm
PReLU

Table 8.3: Convolutional layers have kernels with a shape 3×3 and a number of filters depending on the position in the net. As an activation function was used PReLU. During the training, batch normalization is used.

8.2.1 Training

The output of a U-net model can be treated as a pixel-wise classification, thus it can be trained with cross-entropy. RMSprop [67] with Momentum (set to 0.99) were selected as the optimizer. The learning rate started at 10^{-4}

and was lowered 10 times every time the model did not improve for 5 epochs. The training took 100 epochs, and the U-net with the lowest validation loss was selected.

8.3 Experiments and results

As well as in the previous Chapter, two separate models were created. In this task, no other data were available, so both models are trained only on the dataset described in Table 8.1. The transverse and the longitudinal U-nets were trained two times with a different set of transformations used. In both cases, *Simple transformation* achieved better results than *Complex transformation*. The complete evaluation can be found in Tables 8.4 and 8.5. The U-net fitted the not-augmented data easily, and it did result in the training losses.

Transverse U-net			
Transformations	Training loss	Validation loss	Test loss
<i>Simple tr.</i>	0.21543	0.33858	0.27408
<i>Complex tr.</i>	0.36391	0.38702	0.39736

Table 8.4: Results of the U-net network on the transverse data.

Longitudinal U-net			
Transformations	Training loss	Validation loss	Test loss
<i>Simple tr.</i>	0.10081	0.36706	0.34370
<i>Complex tr.</i>	0.33790	0.36282	0.35817

Table 8.5: Results of the U-net network on the longitudinal data.

Box-plot in Figure 8.4 describes the accuracies of the predicted segmentation masks on the test set. The mean of the test accuracies was 86.53% in the transverse case and 84.23% in the longitudinal case, although the longitudinal U-net had multiple outliers in the predicted masks. The worst test prediction made by the longitudinal model in terms of accuracy is shown in Figure 8.3. Opposingly, Figure 8.4 shows the best test segmentation mask. In the case of the transverse images, the transverse U-net model was able to find the artery easily. Although in some cases, the thickness of the artery wall with plaque is lower than in the segmentation references (Figure 8.5). The best-predicted test transverse mask had an accuracy of 94.96% and can be seen in Figure 8.6.

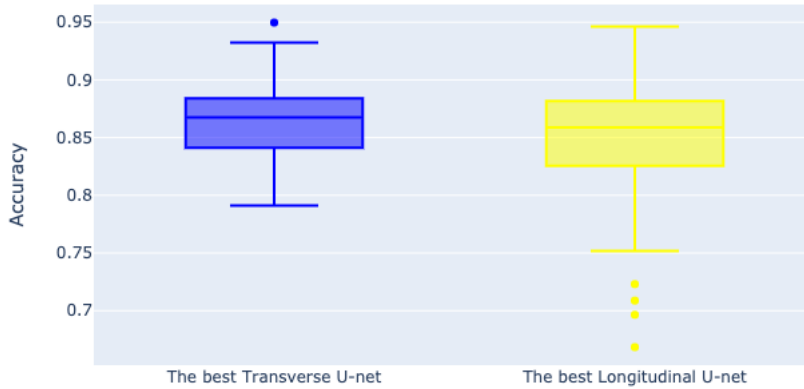


Figure 8.2: Box-plot describing accuracies of segmentation masks made by the best longitudinal and transverse U-nets on the test sets.

The U-net architecture was able to segment the ultrasound images. However, in order to create a network that could be used in the computer-aided diagnosis, the prediction ability needs to be increased, possibly by enlarging the labeled dataset. Despite the fact that U-net is able to be trained on small datasets [66], the training datasets in the available studies are usually bigger than the 75 images used in this project [51, 47, 59, 39]. The accuracy of the longitudinal U-net could be increased by selecting the area of an artery by a rotatable bounding box [55, 49]. The rectangular area selected by this box would contain less unrelated tissue, and the neural network would not be misled as in Figure 8.3. Another possible improvement could be achieved by remaking the borders in the existing references. In the current ground truth segmentations, the plaque is not always directly located on the artery wall—it is separated by a slim region classified as a lumen (Figure 8.3b). Because of this fact, the model is trained on the images, where the plaque is attached to the artery wall, and also on the images where the plaque is surrounded by the lumen. These two facts are contradictory since, from the physiological perspective, the plaque evolves from the artery wall.

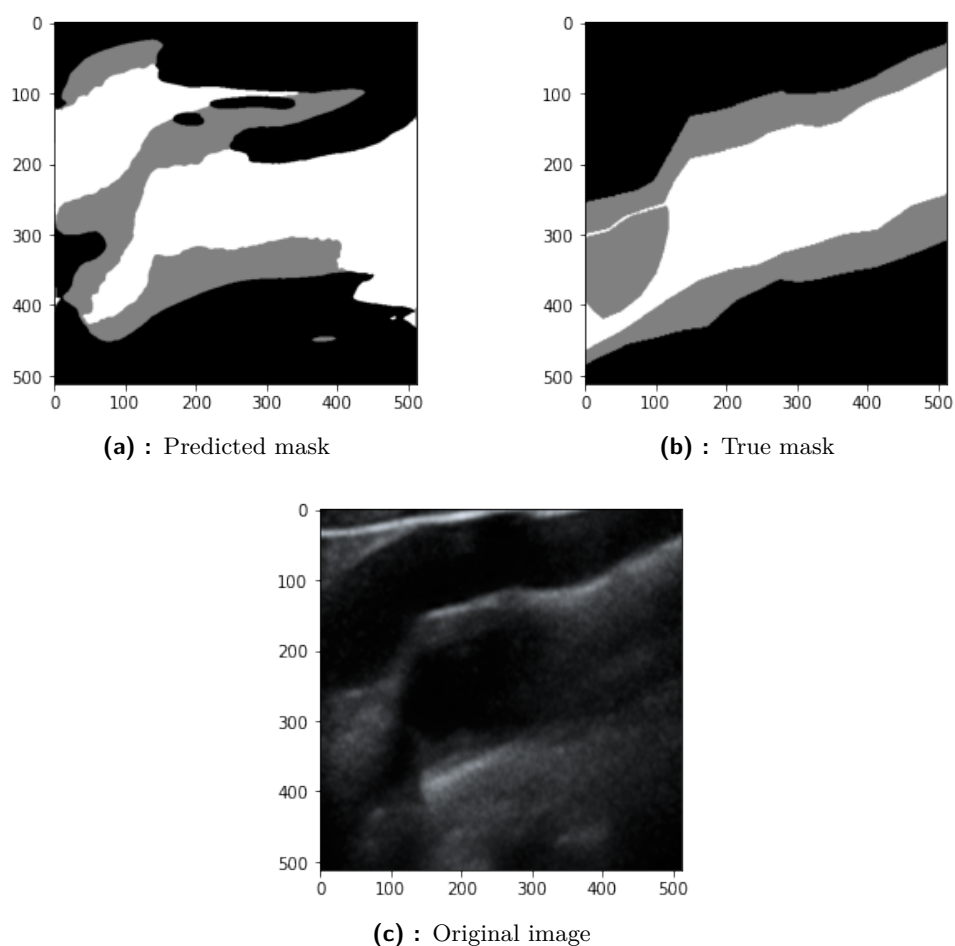


Figure 8.3: The least accurate test segmentation mask of the longitudinal U-net. The artery wall with plaque is grey, the lumen is white, and the rest tissue is black. In this case, the U-net evaluated as the artery not only the true one but the tissue above as well. Indeed, the tuboid shape in the upper part of the ultrasound image reminds an artery. This resulted in low accuracy of 66.84%. However, the model was able to recognize the narrowed lumen in the left part of the carotid artery.

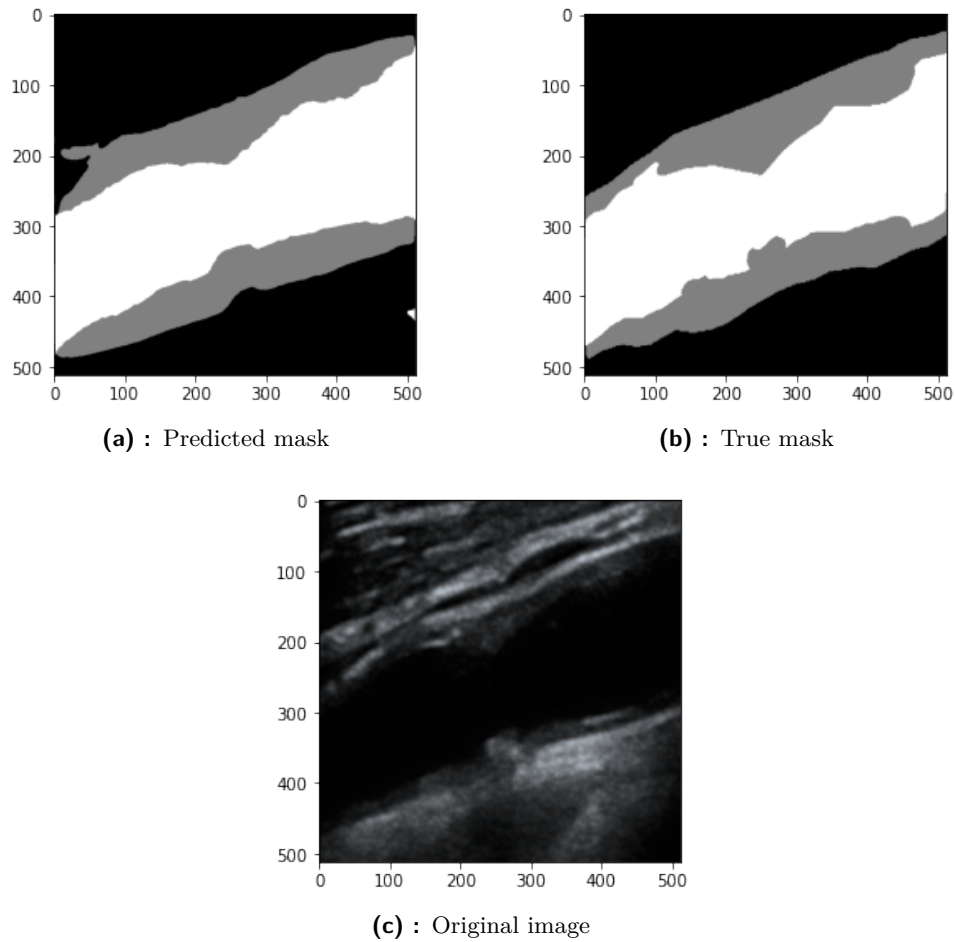


Figure 8.4: The most accurate test segmentation mask of the longitudinal U-net. The network was able to recognize the narrowed area in the middle part of the artery. The accuracy of this mask is 94.61%.

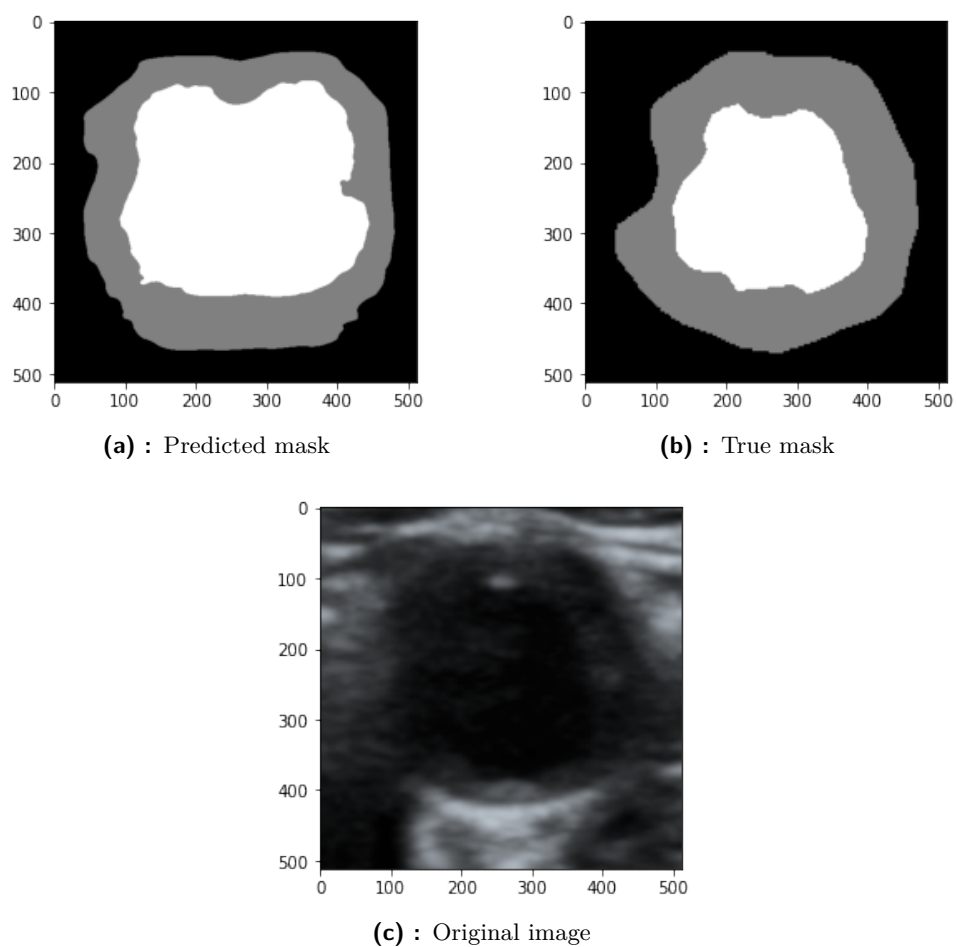


Figure 8.5: The least accurate test segmentation mask of the transverse U-net. It achieved an accuracy of 86.53%. The wall in the segmentation created by the network is thinner than the reference.

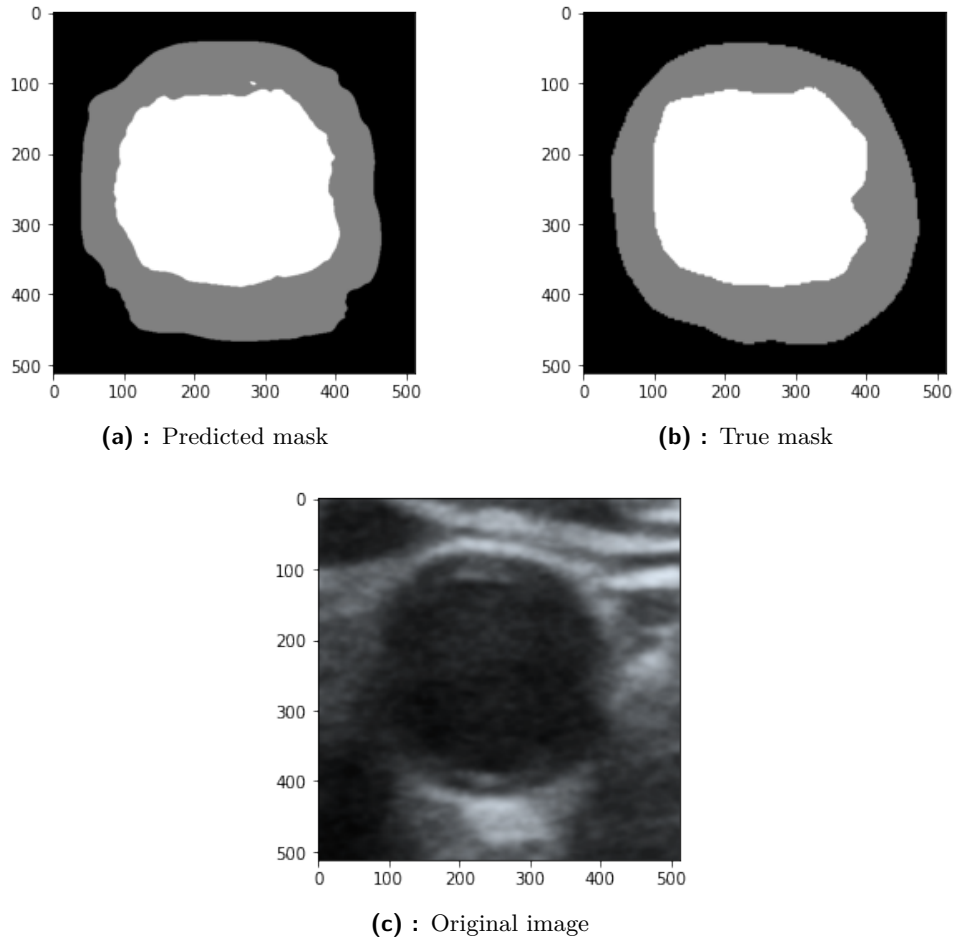


Figure 8.6: The most accurate test segmentation mask of the transverse U-net. The accuracy of the predicted mask was 94.96%. The thickness of the artery wall with plaque is similar to the one in the reference.

Chapter 9

Conclusion

The objective of this thesis was to propose and implement a set of neural networks for the database of ultrasound carotid artery images targeting three different image processing tasks—classification, localization, and segmentation. Over one thousand and five hundred images were classified by hand, on which three CNN architectures were trained and compared. The ResNet50 achieved almost 100% of accuracy on the test set, and easily distinguished the four categories of the ultrasound images. One hundred fifty representative examinations of the patients with carotid artery stenosis were selected, of which one longitudinal and one transverse image with good visibility of an artery was chosen. These data were annotated with bounding boxes localizing the CCA, respectively ICA. In collaboration with the Faculty of Medicine and Dentistry of the Palacký University, these annotations were checked by a group of medical students, who also created the segmentation masks of the carotid arteries. The bounding box annotations were used to train two separate Faster R-CNNs, one for each image type. Both models were able to predict 90% of the test bounding boxes with $\text{IoU} \geq 0.75$. For the segmentation task, the U-net model was selected. The experimental part showed that this architecture of the convolutional neural network is able to achieve solid performance with only 75 training images. The average accuracy of predicted segmentation masks was 86.53% on the transversal and 84.23% on the longitudinal test set. However, such results are not satisfactory in the field of medical image processing, where these segmentation masks would be used to diagnose the severeness of carotid artery stenosis. The U-net could be improved by enlarging the training set, improving the current segmentation references, or in the longitudinal case, by localizing the artery by a rotatable bounding box.

The results of this thesis will be used together with the neural networks and the code base in the research project “Evaluation of atherosclerotic plaque stability in carotids using digital image analysis of ultrasound images”. This project investigates the visual differences in digital images of unstable (symptomatic) and stable (asymptomatic) plaques and the connection between the ultrasound images and the increased risk of plaque progression and risk.

Appendix A

Convolutional neural net

Convolutional neural nets are a family of neural nets, which use convolutional layers. They usually take as an input grid structured data [22], typically images. For example, one may see images as a 3D tensor, one dimension for each primary color in RGB encoding. However, from their first practical use in reading check system [90], they have been applied in many domains, not necessarily in image processing only. They have been used in text processing, for example, in analyzing sentiment of a text [11], time series classification [92], or in the field of recommender systems [37]. Its main component is a convolutional layer, often combined with a pooling layer. One may find convolutional networks combined with fully connected or even LSTM layers [34]. This section will discuss the most used ones—convolutional, pooling, and fully connected layers.

A.1 Convolutional layer

The cornerstone of each convolutional neural net is a convolutional layer. In this layer, one or multiple convolutional filters are applied to the layer's input. In Table A.1, different convolutional filters are applied to an image. In the neural network, every trainable filter is relatively small and serves as a feature extractor. At each position, the filter's kernel takes input from its receptors field, computes its weighted sum, adds bias, and applies a non-linear activation function [88]. Equation A.1 describes this transformation.

$$Y_{i,j} = \sigma\left(b + \sum_{k=0}^x \sum_{l=0}^y w_{k,l} a_{i+k,j+l}\right) \quad (\text{A.1})$$

The kernel is continuously applied over an input, and the distance between two such operations is called a stride. Since convolution reduces an image's dimension, the input is usually padded with some constant value (for example, 0) to preserve it [12]. Figure A.1 shows the convolution on an input of size

A. Convolutional neural net

5×5 , and the size of the kernel is 3×3 . Zero padding is used to keep the spatial dimension.




Convolutional kernel	Image
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
$\frac{1}{9} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

Table A.1: Example of different convolutional kernels. The first one is an identity kernel, which does not change an image. The second one blurs the input, and the last one sharpens it.

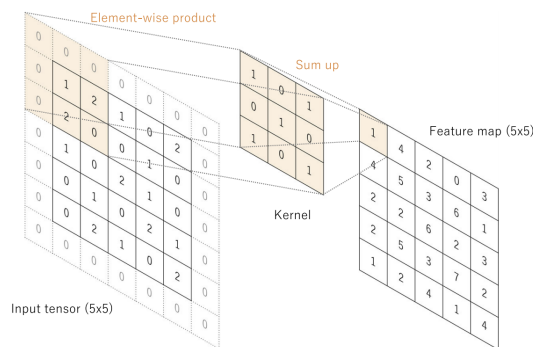


Figure A.1: Example of a convolution of input with shape 5×5 , using 3×3 kernel, zero padding, and stride equal to one. [89]

A.2 Pooling layer

Pooling layers often follow the convolutional ones, and their purpose is to reduce the dimensionality. Firstly, the average pooling was used. It computes the average value of the receptor field, shown in Equation A.2. Max-pooling was introduced in the last years. Such a layer propagates the maximum value at each position (Equation A.3) [63]. For example, a pooling layer with a receptor field of size 2×2 and stride 2 reduces the dimension to half. Such an example can be seen in Figure A.2.

$$Y_{i,j} = \frac{1}{xy} \sum_{k=0}^x \sum_{l=0}^y w_{k,l} a_{i+k,j+l} \quad (\text{A.2})$$

$$Y_{i,j} = \max_{(p,q) \in \mathfrak{R}_{i,j}} (x_{p,q}) \quad (\text{A.3})$$

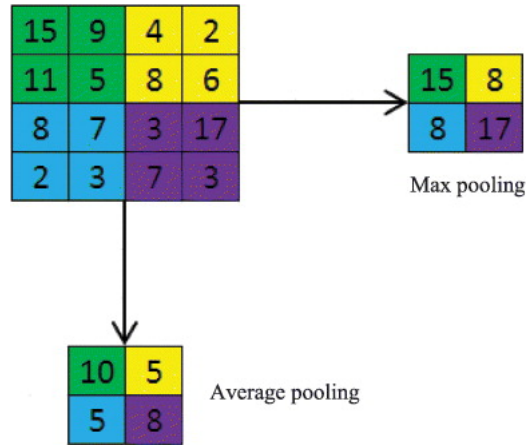


Figure A.2: Comparison of average and max-pooling with receptor field of size 2×2 and stride 2.[63]

A.3 Fully-connected layer

A fully connected layer comprises one or multiple neurons, where each neuron is connected to every unit in the previous layer by trainable weight. With added bias and transformation with a non-linear function, the output is forwarded to the next layer (Equation A.4) [36].

$$Y_{i,j} = \sigma(b_{i,j} + \sum_{k=0}^n w_{i,j,k} a_{j-1,k}) \quad (\text{A.4})$$

A.4 Architecture

Typically, CNN's input size is fixed, so the image needs to be preprocessed accordingly. Firstly, the input is processed by a series of convolutional layers grouped in blocks, where every convolutional layer has the same setting (number of filters, kernel size, stride, padding). Pooling layers reduce the dimension between the blocks, and the number of filters in the next convolutional block is increased. The last part is composed of one or multiple fully connected layers. The number of neurons depends on the application. If the network should behave like a binary classifier, we can use one neuron with a sigmoid activation function. If the aim is to classify n categories, the layer should contain n neurons followed by a soft-max function. In the case of localization of a single object, four neurons can predict two corners of a bounding box surrounding the target. Figure A.3 displays the whole Architecture [74].

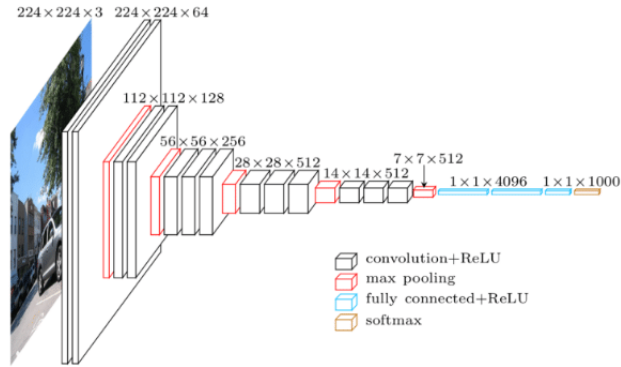


Figure A.3: Example architecture of convolutional neural network (VGG-16) [80].

A.5 Training

During the training of a network, its free parameters, weights, and biases are being changed to values, making the model perform well on the training dataset. A loss function measures the performance of a model. In the case of regression, sum-of-squared errors can be used to compute the fit (Equation A.5).

$$L(\theta) = \sum_{i=0}^N (y_i - f(x_i)) \quad (\text{A.5})$$

Cross-entropy can be used to evaluate the classifier (Equation A.6).

$$L(\theta) = - \sum_{i=0}^N \sum_{k=0}^K y_{ik} \log f_k(x_i). \quad (\text{A.6})$$

The loss of a model $L(\theta)$ is typically reduced by gradient descent, in the neural network case, also called back-propagation. Thanks to the fact that the neural network can be seen as a composition of functions, the gradient can be easily computed by the chain rule. During the gradient descent, free parameters of a model iteratively update. Update of a model parameter w_k in the iteration $e + 1$ has form:

$$w_k^{e+1} = w_k^e - \alpha \sum_{i=0}^N \frac{\delta L_i}{\delta w_k^e}, \quad (\text{A.7})$$

with learning rate α [25].



Appendix B

Implementation details

The project was implemented in the programming language `Python` [87], version 3.7. Deep learning library `PyTorch 1.6` [58] is utilized to use, create, and train neural networks. Code documentation is following `Numpy docstring` guide [14]. Code is formatted by `Black` code formatter [44] and can be found and downloaded from `GitHub` [42]. The models can be downloaded from `Google Drive` [41].



B.1 Project structure

The project contains three main parts: `classification`, `localization`, and `segmentation`. Each one contains an implementation of particular models, datasets, training, and additional functionality used in the project. The code shared between them is stored in the `carotids` directory. Figure B.1 shows the complete structure.



B.2 Examples

For every part, there is an example script that trains the sample model. Training scripts are named `classification_train.py`, `localization_train.py`, and `segmentation_train.py`. Each of them contains a training procedure equivalent to the one which produced the best network for the given task. There are three programs that show how to load and use such models. These scripts can be found under the names `classification_use.py`, `localization_use.py`, and `segmentation_use.py`.

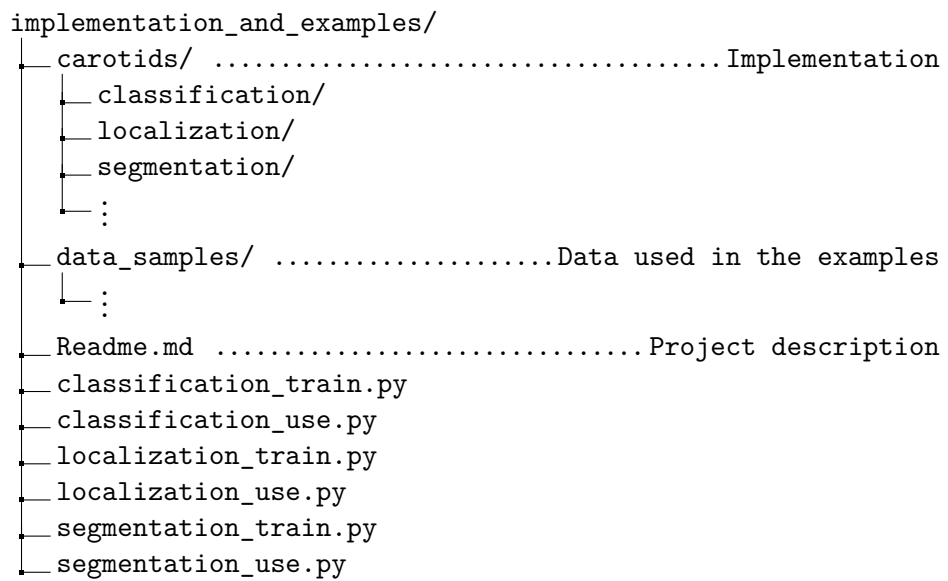


Figure B.1: The implementation of the project is located in the subfolder `carotids`. It contains three different folders, each containing implementation of a different task—`classification`, `localization`, and `segmentation`. In the root directory are a `Readme` file, `License`, and examples. A small number of sample images is present in `data_samples` folder, which are used in the examples.



Appendix C

List of Abbreviations

AI	Artificial intelligence
BUT	Brno University of Technology
CA	Carotid artery
CAD	Computer-aided diagnosis
CCA	Common carotid artery
CNN	Convolutional neural network
CONVL(s)	Convolutional layer(s)
CT	Computed tomography
ECA	External carotid artery
FC(s)	Fully connected layer(s)
ICA	Internal carotid artery
IoU	Intersection over Union
MI	Medical imaging
MRI	Magnetic resonance imaging
RPN	Region proposal network
TIA	Transient ischemic attack
VA	Vertebral artery

Appendix D

Bibliography

- [1] M. Z. Alom, M. Hasan, C. Yakopcic, T. Taha, and V. Asari. Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation. Feb. 2018.
- [2] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE Transactions on Medical Imaging*, 35(5):1207–1216, 2016.
- [3] M. S. Atkins and B. T. Mackiewich. Fully automatic segmentation of the brain in MRI. *IEEE Transactions on Medical Imaging*, 17(1):98–107, 1998.
- [4] R. Benes, J. Karasek, R. Burget, and K. Riha. Automatically designed machine vision system for the localization of CCA transverse section in ultrasound images. In *Computer Methods and Programs in Biomedicine*, volume 109, pages 92 – 103, 2013.
- [5] M. Charlick and J. M Das. Anatomy, Head and Neck, Internal Carotid Arteries. In *StatPearls*. StatPearls Publishing, Treasure Island (FL), 2020.
- [6] H. Chen, C. Shen, J. Qin, D. Ni, L. Shi, J. C. Y. Cheng, and P.-A. Heng. Automatic Localization and Identification of Vertebrae in Spine CT via a Joint Learning Model with Deep Neural Networks. In N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 515–522, Cham, 2015. Springer International Publishing.
- [7] Computer Vision Group, University of Fribourg. U-net wins two Challenges at ISBI 2015. <https://lmb.informatik.uni-freiburg.de/people/ronneber/isbi2015/>, 2015. Accessed: 2020-11-18.
- [8] Department of Neurology, Columbia University. Carotid artery disease. <https://www.columbianeurology.org/neurology/staywell/carotid-artery-disease>, Aug 2020. Accessed: 2020-11-25.

- [9] Department of Telecommunications, Brno university of technology. Signal processing laboratory. <http://splab.cz/en/>. Accessed: 2020-09-12.
- [10] X. Ding, Y. Zhang, T. Liu, and J. Duan. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2327–2333. AAAI Press, 2015.
- [11] C. dos Santos and M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland, Aug. 2014. Dublin City University and Association for Computational Linguistics.
- [12] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. 03 2016.
- [13] O. Emad, I. A. Yassine, and A. S. Fahmy. Automatic localization of the left ventricle in cardiac MRI images using deep learning. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 683–686, 2015.
- [14] J. N. Eric Larson and R. Gommers. numpydoc docstring guide. <https://numpydoc.readthedocs.io/en/latest/format.html>, Jan 2020. Accessed: 2020-12-15.
- [15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.
- [16] L. Gates and J. Indes. Evaluation and treatment of carotid artery stenosis. In *Carotid Artery Disease - From Bench to Bedside and Beyond*. InTech, Jan. 2014.
- [17] S. Ghosh, N. Das, I. Das, and U. Maulik. Understanding deep learning techniques for image segmentation. *ACM Comput. Surv.*, 52(4), Aug. 2019.
- [18] R. Girshick. Fast R-CNN. pages 1440–1448, 2015.
- [19] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.
- [20] I. Gitman, H. Lang, P. Zhang, and L. Xiao. Understanding the role of momentum in stochastic gradient methods. 2019.
- [21] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256, 01 2010.

- [22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [23] H. Gray and W. Lewis. *Anatomy of the Human Body*. Bartleby.com, 2000.
- [24] T. F. Hasan, O. O. Akinduro, N. Haranhalli, and R. G. Tawk. Chapter 3 - neurovascular anatomy in relation to intracranial neoplasms. In K. Chaichana and A. Quiñones-Hinojosa, editors, *Comprehensive Overview of Modern Surgical Approaches to Intrinsic Brain Tumors*, pages 37–38. Academic Press, 2019.
- [25] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009.
- [26] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.
- [27] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.
- [28] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.
- [29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016.
- [30] R. Hiscott. Mortality Risk Following Carotid Artery Stenting, Neurology today. <https://journals.lww.com/neurotodayonline/blog/breakingnews/pages/post.aspx?PostID=442>, Jan 2019. Accessed: 2020-09-15.
- [31] S. Hu, E. A. Hoffman, and J. M. Reinhardt. Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images. *IEEE Transactions on Medical Imaging*, 20(6):490–498, 2001.
- [32] M. Huh, P. Agrawal, and A. A. Efros. What makes ImageNet good for transfer learning?, 2016.
- [33] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning*, 37:448–456, 07–09 Jul 2015.

- [34] F. Karim, S. Majumdar, H. Darabi, and S. Chen. LSTM fully convolutional networks for time series classification. *IEEE Access*, 6:1662–1669, 2018.
- [35] J. Ker, L. Wang, J. Rao, and T. Lim. Deep learning applications in medical image analysis. *IEEE Access*, 6:9375–9389, 2018.
- [36] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516, Dec 2020.
- [37] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16*, pages 233–240, New York, NY, USA, 2016. Association for Computing Machinery.
- [38] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 12 2014.
- [39] M. Kolařík, R. Burget, V. Uher, and M. K. Dutta. 3D Dense-U-Net for MRI brain tissue segmentation. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–4, 2018.
- [40] I. Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, 2001.
- [41] M. Kostelanský. Carotids models. https://drive.google.com/drive/folders/1gRT2sJv0F5efB3eZsnWPdG_CpzvjUcYS?usp=sharing, 2021. Accessed: 2021-01-03.
- [42] M. Kostelanský. Carotids project implementation. <https://github.com/kostelansky17/carotids>, 2021. Accessed: 2021-01-03.
- [43] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.
- [44] L. Langa. Black - The Uncompromising Code Formatter. <https://pypi.org/project/black/>, 2020. Accessed: 2020-12-29.
- [45] G. Lanzino, A. A. Rabinstein, and R. D. Brown Jr. Treatment of carotid artery stenosis: medical therapy, surgery, or stenting? *Mayo Clinic proceedings*, 84(4):362–368, Apr 2009.
- [46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. *Computer Vision – ECCV 2014*, pages 740–755, 2014.

- [47] L. Liu, L. Mou, X. X. Zhu, and M. Mandal. Skin lesion segmentation based on improved U-net. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–4, 2019.
- [48] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128(2):261–318, Feb 2020.
- [49] L. Liu, Z. Pan, and B. Lei. Learning a rotation invariant detector with rotatable bounding box. 11 2017.
- [50] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng. Early diagnosis of Alzheimer’s disease with deep learning. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 1015–1018, 2014.
- [51] Y. Liu, N. Qi, Q. Zhu, and W. Li. CR-U-Net: Cascaded U-Net with residual mapping for liver segmentation in CT images. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2019.
- [52] K. Ma, Z. Shu, X. Bai, J. Wang, and D. Samaras. DocUNet: Document Image Unwarping via a Stacked U-Net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [53] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.
- [54] F. Milletari, N. Navab, and S. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016.
- [55] A. Mohammed, C. Liu, and A. Waheeb. An improved rotation invariant CNN-based detector with rotatable bounding boxes for aerial image detection. In *2019 International Conference on Electronic Engineering and Informatics (EEI)*, pages 251–255, 2019.
- [56] R. Mortimer, S. Nachiappan, and D. C. Howlett. Carotid artery stenosis screening: where are we now? *The British journal of radiology*, 91(1090):20170380–20170380, Oct 2018.
- [57] National Institute of Biomedical Imaging and Bioengineering. Ultrasound. <https://www.nibib.nih.gov/science-education/science-topics/ultrasound>, Jul 2016. Accessed: 2020-11-01.

- [58] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [59] G. Patel, H. Tekchandani, and S. Verma. Cellular segmentation of bright-field absorbance images using Residual U-Net. In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pages 1–5, 2019.
- [60] A. R. Pathak, M. Pandey, and S. Rautaray. Application of deep learning for object detection. *Procedia Computer Science*, 132:1706 – 1717, 2018. International Conference on Computational Intelligence and Data Science.
- [61] K. Prasad. Pathophysiology and medical treatment of carotid artery stenosis. *The International journal of angiology : official publication of the International College of Angiology, Inc*, 24(3):158–172, Sep 2015.
- [62] RadiologyInfo.org. Carotid ultrasound imaging. <https://www.radiologyinfo.org/en/info.cfm?pg=us-carotid>, Feb 2019. Accessed: 2020-11-13.
- [63] W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. PMID: 28599112.
- [64] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 06 2015.
- [65] K. Riha. Artery databases, SPLab. <http://splab.cz/en/research/zpracovani-medicinskych-signalu/databaze/artery>. Accessed: 2020-08-17.
- [66] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015.
- [67] S. Ruder. An overview of gradient descent optimization algorithms. 09 2016.
- [68] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.

- [69] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edition, 2009.
- [70] K. Ríha, J. Mašek, R. Burget, R. Beneš, and E. Závodná. Novel method for localization of common carotid artery transverse section in ultrasound images using modified viola-jones detector. In *Ultrasound in Medicine and Biology*, volume 39, pages 1887 – 1902, 2013.
- [71] S. P. Saha, S. Saha, and K. S. Vyas. Carotid Endarterectomy: Current Concepts and Practice Patterns. *The International journal of angiology : official publication of the International College of Angiology, Inc.*, 24(3):223–235, Sep 2015. 26417192[pmid].
- [72] A. Saxena, E. Y. K. Ng, and S. T. Lim. Imaging modalities to diagnose carotid artery stenosis: progress and prospect. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6537161/>, May 2019. Accessed: 2020-11-11.
- [73] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Molura, and R. 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.
- [74] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [75] Sindhu Ramachandran S. and Jose George and Shibon Skaria and Varun V. V. Using YOLO based deep learning network for real time detection and localization of lung nodules from low dose CT scans. In N. Petrick and K. Mori, editors, *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, pages 347 – 355. International Society for Optics and Photonics, SPIE, 2018.
- [76] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. Breast cancer histopathological image classification using convolutional neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2560–2567, 2016.
- [77] Stanford Health Care (SHC) - Stanford Medical Center. Angiogram for Carotid Artery Disease. <https://stanfordhealthcare.org/medical-conditions/blood-heart-circulation/carotid-artery-disease/diagnosis/angiogram.html>, Sep 2017. Accessed: 2020-10-27.
- [78] Stanford Medical Center. Carotid artery angioplasty. <https://stanfordhealthcare.org/medical-treatments/a/angioplasty/>

- types/carotid-artery-angioplasty.html, Aug 2018. Accessed: 2020-12-31.
- [79] Stanford Vision and Learning Lab. Large Scale Visual Recognition Challenge 2014 (ILSVRC2014). <http://www.image-net.org/challenges/LSVRC/2014/results>, 2014. Accessed: 2020-11-02.
- [80] T. Sugata and C. Yang. Leaf app: Leaf recognition with deep convolutional neural networks. *IOP Conference Series: Materials Science and Engineering*, 273:012004, 11 2017.
- [81] K. Suzuki. Overview of deep learning in medical imaging. *Radiological Physics and Technology*, 10, 07 2017.
- [82] TA ČR Starfos. Evaluation of atherosclerotic plaque stability in carotids using digital image analysis of ultrasound images. <https://starfos.tacr.cz/en/project/NV19-08-00362>, 2019. Accessed: 2020-11-02.
- [83] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A Survey on Deep Transfer Learning. In V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham, 2018. Springer International Publishing.
- [84] Y.-X. Tang, Y.-B. Tang, Y. Peng, K. Yan, M. Bagheri, B. A. Redd, C. J. Brandon, Z. Lu, M. Han, J. Xiao, and R. M. Summers. Automated abnormality classification of chest radiographs using deep convolutional neural networks. *npj Digital Medicine*, 3(1):70, May 2020.
- [85] UNC Computer Vision Recognition Group. ImageNet Large Scale Visual Recognition Competition 2015 (ILSVRC2015). <http://image-net.org/challenges/LSVRC/2015/results>, 2015. Accessed: 2020-11-04.
- [86] U.S. Department of Health and Human Services. Carotid artery disease. <https://www.nhlbi.nih.gov/health-topics/carotid-artery-disease>. Accessed: 2020-11-11.
- [87] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.
- [88] J. Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5:23, 2017.
- [89] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, Aug 2018.
- [90] Yann Le Cun, L. Bottou, and Y. Bengio. Reading checks with multilayer graph transformer networks. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 151–154, 1997.

- [91] Z. Zhang, Q. Liu, and Y. Wang. Road extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.
- [92] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, 2017.
- [93] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Fast Automatic Heart Chamber Segmentation from 3D CT Data Using Marginal Space Learning and Steerable Features. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8.
- [94] M. Zúkal, R. Benes, P. Cíka, and K. Říha. Ultrasound image database | SPLab. <http://splab.cz/en/download/databaze/ultrasound>. Accessed: 2020-12-03.
- [95] K. Říha and R. Beneš. Circle detection in pulsative medical video sequence. In *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS*, pages 674–677, 2010.
- [96] D. Školoudík. Atherosclerotic Plaque Characteristics Associated With a Progression Rate of the Plaque in Carotids and a Risk of Stroke. <https://clinicaltrials.gov/ct2/show/NCT02360137>, Feb 2015. Accessed: 2010-09-30.