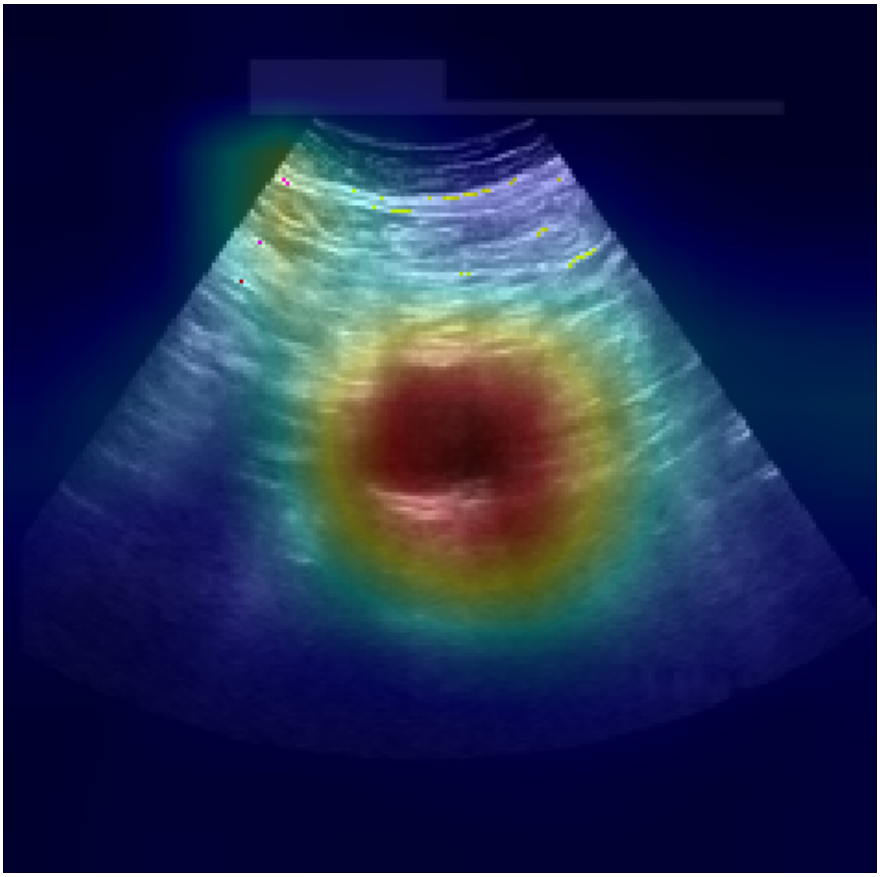
CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG



Multimodal deep learning for diagnosing sub-aneurysmal aortic dilatation

DATX05

SARA FINATI
ELIN LJUNGGREN

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY AND UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2019

Master's thesis 2019

Multimodal deep learning for diagnosing sub-aneurysmal aortic dilatation

SARA FINATI
ELIN LJUNGGREN

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer science and engineering
Software engineering
Chalmers University of Technology and University of Gothenburg
Gothenburg, Sweden 2019

Multimodal deep learning for diagnosing sub-aneurysmal aortic dilatation
SARA FINATI & ELIN LJUNGGREN

Supervisors: Charlotta Aguirre Nilsson and Medina Velic, QRTECH AB
Examiner: Richard Torkar, Department of CSE

Master's Thesis 2019
Department of Computer science and engineering
Software engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Heatmap visualization of a correctly classified aneurysm.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2019

Multimodal deep learning for diagnosing sub-aneurysmal aortic dilatation
SARA FINATI & ELIN LJUNGGREN
Department of Computer science and engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Abdominal Aortic Aneurysm (AAA) is a localized enlargement of the abdominal aorta that can progress to a rupture, which will cause an internal bleeding that is fatal in the majority of the cases. To save more lives, a nationwide screening program invites all men at 65 to measure their largest aortic diameter during an ultrasound examination. The diagnosis is based solely on this diameter and if it is below 30 mm the patient is declared healthy and discarded from any follow-up monitoring. However, recent studies have shown that patients with a diameter within 25–29 mm, a sub-aneurysm, run an elevated risk of developing a full-size aneurysm and hence might need further surveillance.

This thesis is a collaboration between the product development company QRTECH AB and Västra Götalandsregionen (VGR). It proposes a novel solution for predicting which sub-aneurysms that might grow into a full-size aneurysm with the help of an ultrasound image complemented by patient data including aortic diameter, number of years of smoking and snus, blood pressure and medications. The solution consists of a multimodal deep learning algorithm that classifies the sub-aneurysms as either healthy or sick and thereby suggests patients that should be kept under surveillance.

Due to lack of any follow-up data for the men with sub-aneurysms a comparison with meta-studies, examining how many sub-aneurysms that progressed into a full-size aneurysm, was carried out. The results from those studies did not agree with the results obtained from classifying the sub-group in this project. A feasible explanation is the limited data set which most likely affected the learning. However, the evaluation of the model's performance was still promising and indicates the potential of using neural networks for diagnosing AAA.

Keywords: multimodal deep learning, abdominal aortic aneurysm (AAA), sub-aneurysmal aortic dilatation, VGG19, Keras, heatmaps, permutation importance.

Acknowledgements

The first persons that deserve a huge thanks for helping us develop this thesis are our fantastic supervisors at QRTECH, Charlotta Aguirre Nilsson and Medina Velic. We cannot thank you enough for all the help you have given us and for being supportive and encouraging along the way. Also a big thank you to Gunnar Snorri Ragnarsson and Sankar Sathyamoorthy at QRTECH for all guidance regarding deep learning.

We want to show our appreciation for our examiner Richard Torkar, for accepting the role with such short notice and providing valuable feedback on the report.

Thanks to Marcus Langenskiöld, consultant vascular surgeon at the Department of Vascular Surgery at Sahlgrenska University Hospital, for sharing your medical expertise on the subject and helping us establish the purpose of the project. Another important person during this work was Lars Lindsköld who provided us with the data and always believed in us and encouraged our work, thank you!

We would also like to express our biggest gratitude to our family and friends for your support through our ups and downs, we could not have done it without you.

Finally, a big shout-out to everyone at QRTECH for always giving us a good laugh and for making our days at the office pure joy! Go QRTEAM!

Thank you!

Sara Finati & Elin Ljunggren
Gothenburg, June 2019

Contents

List of Figures

List of Tables

List of acronyms

AAA Abdominal Aortic Aneurysm

AP Average Precision

ANN Artificial Neural Network

AUC Area Under the ROC Curve

CAM Class Activation Map

CLAHE Contrast Limited Adaptive Histogram Equalization

CNN Convolutional Neural Network

ECM Extra Cellular Matrix

EVAR EndoVascular Aortic Repair

FN False Negative

FP False Positive

FPR False Positive Rate

MLP Multiple Layer Perceptron

OCR Optical Character Recognition

ReLU Rectified Linear Units

ROC Receiver Operating Characteristic

ROS Reactive Oxygen Species

SGD Stochastic Gradient Descent

TN True Negative

TNR True Negative Rate

TP True Positive

TPR True Positive Rate

VGR Västra Götalandsregionen

1

Introduction

This chapter starts by introducing the background of this thesis, followed by a clarification of the purpose and what it aims to accomplish. Lastly, a section about related research is covered.

1.1 Background

Abdominal Aortic Aneurysm (AAA) is a localized enlargement of the abdominal aorta and considered life-threatening. A rupture of such aneurysm will cause an internal bleeding that is fatal in the majority of the cases [1]. The condition is not reversible but change of lifestyle could halt the progression of the aneurysm and consequently reduce the mortality risk [2].

To minimize the death rate, a screening program was initiated in Sweden in 2006 and reached nationwide coverage in 2015. It has been very successful since it is a cost-efficient way for early detection of AAA [3]. The program focuses on the high risk population, hence all men at 65 years of age are invited to do an ultrasound examination of the abdominal aorta to measure its largest diameter [1]. If the diameter is 30 mm or larger, the man is diagnosed with AAA. To this day, a total of $302,957$ men have been called for examination, 84% have attended, and the prevalence of detecting AAA is currently 1.5%. Though, this does not reflect the whole society since the compliance is lower in areas with low socioeconomic status [2]. The screening program is predicted to annually prevent 90 premature deaths from aortic aneurysms and to gain 577 quality-adjusted life years [3].

Today, the diagnosis of AAA is solely based on the diameter of the aorta as mentioned above. Although this way of diagnosing has high specificity and sensitivity, there is room for improvement. One disadvantage is the case of overdiagnosis. This can for example mean that a dilatation $\geq$ 30 mm is diagnosed as an aneurysm even though it will never grow to a critical size (usually $\geq$ 55 mm) where it might rupture. Obtaining such a diagnosis can lead to psychosocial deterioration due to unnecessary anxiety [1]. Another disadvantage is the strict limit of 30 mm since all men with an aortic diameter less than this are discarded from further examinations as they are perceived to be low-risk subjects. But recent studies have shown that

the sub-group with a sub-aneurysmal aortic dilatation of 25–29 mm runs an elevated risk to develop a full-size aneurysm [2]. In addition, the increasing life expectancy makes the risk even higher [4].

There is still an uncertainty regarding how cost-efficiency, and the life quality of the men in the sub-group, will be affected if they would be incorporated into the follow-up monitoring provided by the screening program [2]. Since it is desirable to keep both the cost and cases of overdiagnosis down, it would be beneficial to try to find which of the men in the sub-group that run a risk of actually developing an aneurysm. It is therefore of interest to investigate if there are any underlying causes that might conclude which sub-aneurysms would progress further.

A well suited machine learning algorithm for this type of purpose is Artificial Neural Network (ANN). This algorithm has recently found many uses within applications regarding medical diagnosis since it in many ways enhances doctors' ability to analyze medical data [5, 6]. From the screening program, patient data such as age, smoking habits, diameter of the aorta and blood pressure can be obtained together with the images from the ultrasound examinations. The patient data and the images belong to two different modalities. By analyzing these parameters in a neural network, so called multimodal deep learning, features indicating AAA could potentially be found that otherwise would have been left unseen. These features may be a first step to develop a more individualized diagnosis and treatment of AAA among the men having sub-aneurysms.

1.2 Aim

The aim of this project is to investigate if a neural network can help determine which of the patients in the sub-aneurysm group are most likely to develop a full-size aneurysm and therefore need further surveillance.

1.3 Specification of issue under investigation

This project is a collaboration between the engineering company QRTECH and Västra Götalandsregionen (VGR), with the main focus to investigate if the ultrasound images together with patient data can indicate which of the men with a sub-aneurysmal aortic dilatation that should be kept under surveillance. Since the provided data set does not contain any information whether these patients actually will develop a full-size aneurysm or not, the obtained results cannot be fully verified. Instead a comparison with meta-analyses, which studies how many of the sub-aneurysms that progressed to an aneurysm within five and ten years, will be carried out. Due to the lack of monitoring data, it is also essential to analyze what the neural network will base its decisions on.

The main questions covered in this thesis are:

- *How well do the results agree with the meta-analyses? Will the percentage of how many sub-aneurysms that progressed into a full-size-aneurysm coincide with how many of the men in the sub-group that are classified as sick?*
- *What image features will the network find most important?*
- *Which of the input parameters yield the highest significance?*

To answer these questions a multimodal deep learning algorithm will be implemented to classify the sub-group as either sick or healthy. The structure will consists of two branches, one analyzing the image data and the other analyzing the patient data before being combined to a joint decision.

Regarding the second question, saliency maps and heatmaps will be employed to visualize what image features the network will find important. These maps will highlight the pixels or regions that contribute the most to the classification.

Finally, a technique called permutation importance will be applied to identify how significant each input parameter is. This is done by shuffling one input column leaving the others unaffected, all inputs are then fed to the network to be classified as either sick or healthy. How much the accuracy is degraded will determine the significance of the shuffled parameter.

1.4 Limitations

Due to the limited time frame of this thesis the neural network will not be built from scratch. Instead, an existing network architecture will be used as foundation and adjusted in order to fit the problem covered in this project. Further, the algorithm will be developed to only perform binary classification, meaning that the patients having sub-aneurysms will only be classified as either sick or healthy. A sick classification means that the patient will develop an aneurysm over 30 mm and a healthy will not. Hence, the neural network will not be able to predict which of the sub-aneurysms that will progress to a critical size where it might rupture.

The provided data originates from VGR and is part of the aforementioned screening program. It will therefore be limited in both age and gender but also geographically. Since socioeconomic factors affect the compliance and prevalence, the data set might not represent the true population and the obtained results might differ from the meta-analyses.

1.5 Related work

Following sections will first introduce a successful implementation of multimodal deep learning combining images and non-image data for improving a medical diagnosis. Secondly, the meta-analyses will be presented, which will be compared to the results obtained from this project.

1.5.1 Multimodal deep learning for cervical dysplasia diagnosis

An application for multimodal deep learning in the medical domain is discussed in the paper by Xu et al. [7] from 2016 regarding diagnosis of cervical dysplasia. During the patient's screening visit, a digital image of the cervix is collected together with clinical test results. Previous research showed that these modalities could provide complementary information and consequently improve the diagnostic accuracy [8, 9]. However, the image data and the clinical data were trained separately and then fused together to yield the final decision [7]. This approach does not fully exploit the implicit correlations across the modalities, which is why the authors proposed an improved framework. A Convolutional Neural Network (CNN) was implemented to convert the image data to a feature vector which can be fused with non-image data. The network was then concatenated with joint fully connected layers to learn the non-linear multimodal correlations. The final network was able to predict cervical dysplasia with an accuracy of 88.91%, sensitivity of 87.83%, and specificity of 90%. These results significantly outperform other methods that only uses single modality information.

1.5.2 Sub-aneurysmal aortic dilatation

In most screening programs patients diagnosed with a sub-aneurysm are left with no further examinations. But sub-aneurysmal dilatation do not represent a normal aortic diameter and it has been discovered that these persons are likely to develop a full-size aneurysm. Therefore, investigations have been made to explore the benefits of follow-up surveillance of patients with sub-aneurysms and to study how sub-aneurysms grow. Two studies that have investigated this are meta-analytical in their character, i.e., the studies by *et al.* [10] 2013 and *et al.* [11] 2018.

et al. [10] collected data from eight screening programs in England, Denmark and Finland which all had made long term follow-up of their participants. The meta-analysis ended up with $N = 1,696$ patients with sub-aneurysms, having a median age of 66 years at the first examination and 66 of them being females. These patients were followed up with a median time of every 4 years and $1,011$ of the $1,696$ subjects (59.6%) developed aneurysms (mean time of 4.7 years). At 5 years of surveillance even more of the patients had developed aneurysms (67.7%). Reaching the point

after 10 years where 96% of the patients had developed an aneurysm [10].

How many of the patients in the screening program that developed an aneurysm > 50 mm and were suited for surgical repair was also investigated by *et al.*. But in this sample only 11.5% developed an aneurysm of this size (mean time of approximately 11 years). Seven of the screening programs provided additional data for how many of patients' aortas that progressed to a rupture. There were 14 reported ruptures and the mean time from the first examination to the rupture was 18.7 years. Although there could be a risk that this number is under-reported since the patients were under surveillance and would have had preventive surgical repair before [10].

In the study by *et al.* [11] collection of data from different screening programs and studies was also performed. This resulted in 37 studies performed on males with participant size ranging from 3 to $52,690$. In these studies the prevalence of sub-aneurysms ranged from 1.14% to 8.53% and by the 5-year follow-up 55% to 88% of these progressed to an aneurysm [11]. Four of the studies also reported the number of participants with sub-aneurysms that proceeded to surgical repair. Out of these, 10% had elective surgery and 1% had emergency surgery after rupture [11].

The conclusion from both of these investigation whether or not to include patients with sub-aneurysms in follow-up programs was ambiguous. *et al.* [11] could not confidently say that the patients should be kept in surveillance while *et al.* [10] recommended to include them.

1.6 Thesis outline

Chapter 2 will present the theoretical foundation this thesis relies on. It will provide an overview of the disease AAA as well as introducing the key concepts of ANN.

In **Chapter 3** the methods used for implementing and evaluating the multimodal deep learning algorithm are described thoroughly, while the obtained results will be displayed in **Chapter 4**. **Chapter 5** presents an analysis of the results and **Chapter 6** discusses the questions stated in Section 1.3. The threats to this project's validity is covered in **Chapter 7**. Lastly, **Chapter 8** will provide concluding remarks and ideas for future work.

2

Theory

This chapter will introduce the theoretical foundation that will be necessary in order to fully understand the upcoming work. A further explanation of the concept of Abdominal Aortic Aneurysm will be given, followed by a presentation of the key concepts of Artificial Neural Network.

2.1 Abdominal aortic aneurysm

Abdominal Aortic Aneurysm (AAA) is an enlargement of the aorta usually located below the vessels leading to the kidneys, see Figure 2.1 [2].

There are usually no symptoms characterizing AAA. The causes of AAA are believed to be degradations of connective tissue proteins like elastin and collagen, or chronic inflammation in the middle layer of the vessel wall. This will reduce the strength of the wall and in the end cause it to erupt [2]. Risk factors causing AAA are mainly smoking, but also male sex, high blood pressure (hypertension), age, high BMI and high cholesterol. Heredity and family history can also be contributing factors [2, 12]. The formation of small aneurysms have been associated with diabetes but these usually grow slower than aneurysms in a non-diabetic patient [13].

The number of diagnosed AAA cases has been decreasing the past 10–20 years and today the occurrence in Europe and USA are 2–3% for men between 65–70 years. In

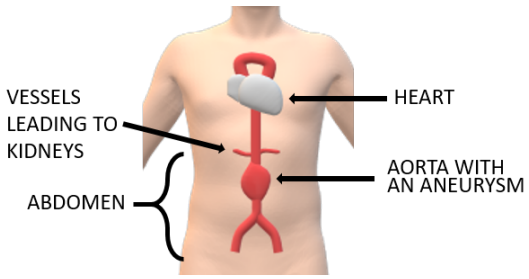


Figure 2.1: Schematic visualization of the location of an AAA. As the figure shows the aneurysm is usually located in the abdomen beneath the vessels leading to the kidneys.

Sweden the corresponding number is 2%, women are not included in these numbers since AAA is 4–6 times more common among men. The decrease of AAA cases is believed mainly to be the result of a decreased amount of smokers in the general population [2].

An aneurysm grows slowly but the speed increases as the aneurysm gets larger. The annual growth is estimated to 5–10% of the diameter, though there is a high variability and the aneurysm does not grow continuously. Its way of growing is instead characterized by growth periods. The same risk factors for causing the aneurysm are also believed to increase its growth speed, with the exception of diabetes [2]. Previous studies have also shown that a thrombus located in the infrarenal aorta, in the aorta below the kidneys, can promote a faster growth. A large thrombus stimulates inflammation and the production of Reactive Oxygen Species (ROS) and Extra Cellular Matrix (ECM) degradating enzymes. These are all molecular processes that are involved in the development of AAA [14].

There are no specific treatments for preventing the aneurysm from growing or slowing down its growth. Trials have been made with several different medications and compounds, though the sample sizes in each trial have been quite small and there has been no significant effects on the primary outcomes [14]. Today the only recommendations for slowing down the growth of aneurysms is to quit smoking or in some suitable cases treat the patient for hypertension [2].

If the aneurysm keeps growing it will eventually erupt and to prevent this a surgical intervention is made when the aneurysm reaches 50–55 mm in diameter. These aneurysms are usually detected by an ultrasound examination as part of a screening program. The patient then gets a remittance to a vascular surgeon and a CT-scan is made to validate the aneurysm's diameter from the ultrasound examination. If the diameter is still $>$ 55 mm, a surgical intervention is made [2]. This surgical intervention is the same as when the aneurysm ruptures, either open surgery or EndoVascular Aortic Repair (EVAR). Both of these interventions replace the aneurysm with a vascular prosthesis, but they differ when it comes to type of prosthesis inserted and the procedure. Open surgery is an anaesthetic procedure where the abdomen is cut open and a synthetic graft is sewn to the aorta. When instead using EVAR, a catheter is entered from the groin arteries into the aorta and through this a stent graft is inserted and hooked to the wall of the aorta. EVAR is the currently most used method and has the best short term effect due to its low risk of immediate complications of the surgery. But complications from the stent graft itself is more common than when using a synthetic graft and thereby requires more surveillance. This implies that the long term effect between the two methods is negligible [2].

The risk of rupture increases as the aneurysm grows and is believed to be around 10% for aneurysms $>$ 60 mm. Due to the lack of symptoms the most common case is to discover it during screening or, in worst case, when the aneurysm ruptures. If it ruptures, the patient can experience pain beaming from the abdomen to the back and the sides and then quickly lose consciousness. The only way to treat a ruptured

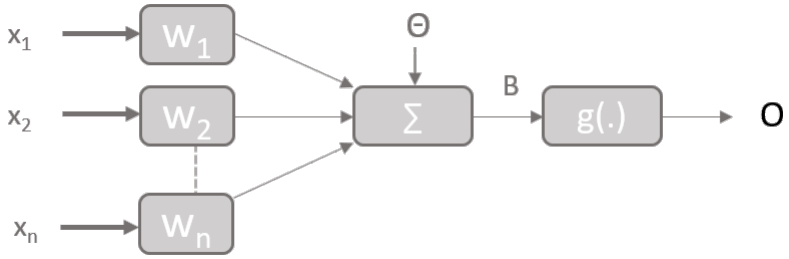


Figure 2.2: A graphical representation of a perceptron corresponding to equation $O = g(B) = g\left(\sum_{i=1}^{n} w_i x_i + \Theta\right)$.

aneurysm is by surgical intervention either by open surgery or EVAR. But even though the patient gets to the hospital and receives treatment the total mortality rate is 70–80% [2].

2.2 Artificial neural networks

Diseases, like AAA, usually consist of complex relations that need to be analyzed in order to explain why they occurred and how they will progress. One way of analyzing these relations is by Artificial Neural Networks (ANNs) which have emerged more and more lately. This method has been used in industry and research for many years and has now also entered the healthcare business [15, 16]. The concept of ANN is inspired by how the neurons in the human body store input data and learn from it. Artificial neurons have the same main features as the biological neuron which are parallelism and high connectivity and are called perceptrons. They are fed with an input signal, x_i, that is weighted using the synaptic weights, w_i, and summed together. A bias, Θ, is introduced that pushes the sum in the direction of the neuron output. This gives the local field, B, which is inserted into an activation function, $g(B)$, and equals the final output, O [17]. The corresponding equation of the model can be seen below in equation (2.1)

$$O = g(B) = g\left(\sum_{i=1}^{n} w_i x_i + \Theta\right), \tag{2.1}$$

while a graphical representation is illustrated in Figure 2.2.

ANNs consist of several processing units that represent the perceptrons and are linked together by many forward directed interconnections, i.e., artificial synapses. The interconnections are weighted using synaptic weights that determine the importance of the different synapses. As the network learns, these weights are adjusted according to the input data. The structure of ANNs is what makes them suited for classification and prediction tasks since they can extract relationships between several variables in the targeted application [15].

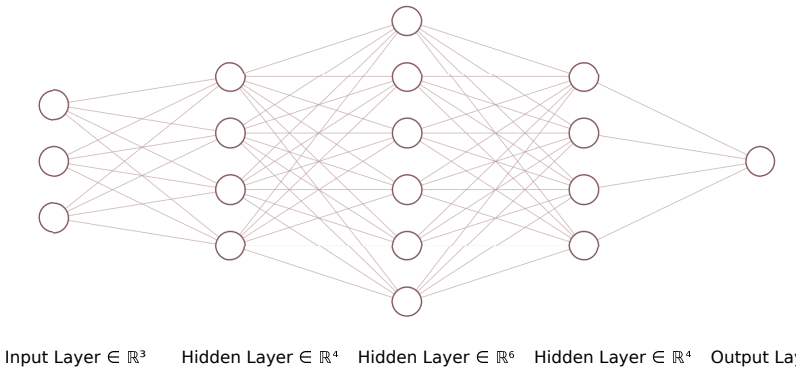


Figure 2.3: Example of a multiple layered perceptron with one input layer, three hidden layers and one output layer.

A network can consist of a single layer or multiple layers of perceptrons, where the latter is called Multiple Layer Perceptron (MLP). MLPs usually have one input layer, a number of hidden layers and an output layer. An example of this kind of network can be seen in Figure 2.3, where all the neurons are connected with the neurons in the previous and in the next layer. This type of hidden layers are described as fully connected layers and they build up a fully connected network [17].

Networks with two or more hidden layers are usually referred to as deep networks and can be trained to solve complex classification tasks with high accuracy. One type of a deep network that is commonly used in image analysis is the Convolutional Neural Network (CNN). In these networks the different layers consist of feature maps that recognize different geometrical features in the images [17].

2.2.1 Convolutional neural networks

CNNs are designed for object recognition and pattern detection as they take an input image and then the layers of neurons, the filters, detect local features, like edges or corners, and create feature maps. Since similar features occur in different parts of the image, one type of filter can be used in multiple areas. This means that a certain filter detects only one certain feature, for example one filter might detect edges while another one detects corners [17]. In this way the filters are translation invariant, implying that the network can learn the object features irrespective of where they are [18]. With this knowledge the number of weights can be greatly reduced compared to regular fully connected networks since the weights are shared between the neurons in the filter [19]. This is the most important characteristic of CNNs since it makes them cheaper to train regarding computational power and it also reduces the risk of overfitting [17].

The feature map created from one filter is fed to the next where the mathematical operation convolution is performed, creating a new feature map. The filter performing convolution is referred to as convolutional layers, hence the name Convolutional Neural Network. These networks also consist of other types of layers, for example

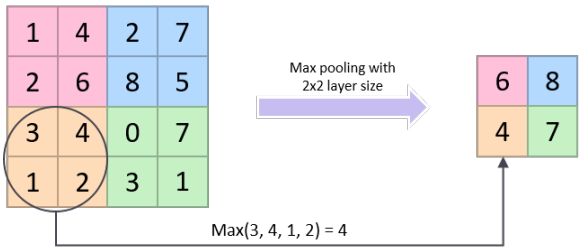


Figure 2.4: A demonstration of how max pooling works. It can be seen in the figure how the max pooling layer reduces the size of the feature map and creates a smaller, translation invariant representation of the map.

pooling layers that are connected directly to a convolutional layer. Pooling layers simplify the output from the convolutional layer which can be done in different ways, for example by the use of max pooling [17]. This method gives the maximum output within a neighborhood making the representation of the feature map invariant to small translations of the input. In Figure 2.4 an illustration of how a max pooling layer works is shown [18].

Most images share the same type of low-level features like edges, geometric shapes, changes in lighting, etc. This can be utilized for the purpose of transfer learning where a pre-trained network can be retrained to perform a different task than previously trained to do. It is often applied in cases where the data set associated with the new task is small. Therefore it could be an advantage if the network has already learned the basic features and is only fine-tuned on the data set corresponding to the new task [18].

In CNNs most of the basic features are learned by the lower layers which means that these layers will be shared for most image dependent tasks and the upper will be task dependent. This is why it is, in most cases, sufficient to only retrain the upper layers when doing transfer learning [18]. There are several different open source pre-trained networks available that can be used for transfer learning, for example the Google Inception, Microsoft ResNet and Oxford VGG-networks [20, 21, 22].

VGG19 is a well-known and well-established CNN architecture for image classification tasks, where the number 19 stands for the number of weight layers in the network. The network was developed by Simonyan and Zisserman and their team VGG for the competition ImageNet Challenge 2014 [22].

For the competition it was trained on the data set from the 2012 edition of the ImageNet Challenge. This set is a subset from the large ImageNet dataset and contains 1,000 classes where the training set consists of 1.3 million images, the validation set of 50,000 images and the test set of 100,000 images. Since the network performs a multiple classification task it has a softmax function as output layer that classifies the input image into one of the 1,000 classes from the dataset. The architecture of the VGG19 network is presented in Figure 2.5 together with the size of each feature map and the input image [22].

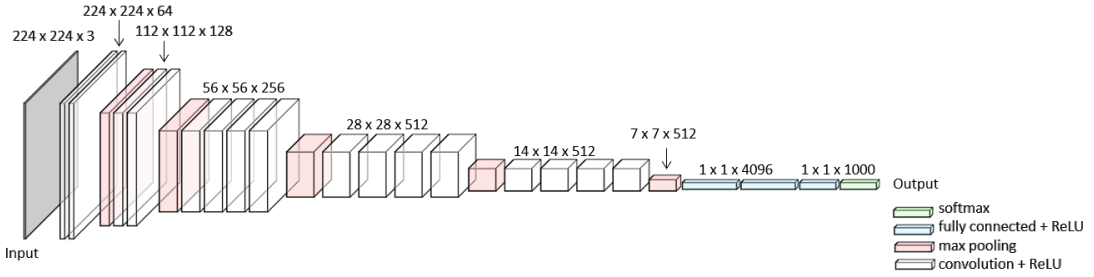


Figure 2.5: The VGG19 network architecture with its corresponding filter sizes.

There are networks that are a lot deeper than VGG19, like GoogLeNet and InceptionV3. These networks consist of 22 and 48 weight layers respectively and have also been tested on the 2012 ImageNet challenge with acceptable results [23, 24]. Recently, large networks have been increasingly doubted and it has been investigated if there are smaller networks that can perform at the same level. Usually when larger networks are trained a technique called pruning is employed where neurons with small contributions are removed and the network is thereby decreased in size.

Frankle and Carbin [25] have presented a technique for retrieving a smaller sub-network contained within a larger network that yields equal or higher performance while the size is reduced by 10–20%. The sub-network is formed by the remaining neurons from pruning a trained network. By then resetting the weights of the sub-network with the same weights as when initializing the original larger network, Frankle and Carbin succeeded to either meet or exceed the original network's performance. Though when randomly initializing the weights all over again the sub-network performed far worse, implying that structure alone cannot explain the success [25].

2.2.2 Multimodal deep learning

Usually when making more complex decisions or classification tasks more than one type of information, modality, is used. For example when diagnosing diseases, information from X-ray or MRI scans is weighted together with blood samples, examination results and other types of interventions before taking a final decision. In machine learning this is called multimodal deep learning and has been used in several different applications for example when diagnosing cervical dysplasia, as mentioned in Section 1.5.1 [9].

There are different ways of performing multimodal deep learning. Specifically there are several techniques for when and how to merge the different kinds of modalities, for example late or early fusion, hybrid fusion or model ensemble. These methods have in common that the features, either final or intermediate, are joint together to make a final decision [26].

The two most common methods are late and early fusion. Late fusion implies that

the two separate networks give separate decisions that are merged together to give a final decision. When doing early fusion, features are taken out from the two separate networks and merged together into a new vector. This vector is used as input into another network with more hidden layers and an output layer that gives a final decision [9].

2.2.3 Training

Neural networks require training before they are able to perform certain tasks such as classification or segmentation. The training is an iterative process and can be divided into two subgroups: supervised and unsupervised.

Supervised learning means that there exists defined outputs. The network takes the inputs and desired outputs and updates its internal state in order for the predicted output to be as close as possible to the true value [18]. Contrarily, unsupervised learning does not require any knowledge about the outputs. In this case, the goal is instead for the network to learn the underlying patterns of the input data to identify subsets that contain similarities [15].

Before the training phase can start the hyperparameters have to be set. These parameters refer to the settings of the network and determine the structure such as number of layers and hidden units but they also determine the learning process [18]. The latter include among others the learning rate, optimization method, cost function and activation function which all will be explained in following sections. Another set of hyperparameters are the weight and bias initialization. Since the optimal values are unknown at this point, prior to training, the weights are usually chosen randomly [17] and the biases may be set to zero [18]. When applying the technique of transfer learning, described in Section 2.2.1, the weights and biases from a pre-trained network are used for the initialization instead. This implies that the network has already learned certain image features and will not have to start from zero.

The choice of hyperparameters is essential for the performance of the model but finding the best values is challenging. One approach is to split the original data set into three subsets: training, validation, test and then manually tune the parameters based on the validation set performance. The validation set will initially provide an unbiased evaluation of the model since it is not used during training. However, as the tuning proceeds the network will become progressively more biased as information from the validation set will be incorporated into the model settings. The purpose of the test set is therefore to work as a final and completely unbiased evaluation [27].

Optimizer

The training process can be seen as an optimization problem where the optimal values of the weights and biases minimizes the cost function, C. In many cases it

is sufficient to stop training when the cost is small enough instead of when it has reached global minimum, implying that the obtained solution does not have to be unique [17]. A typical approach for finding a solution is to apply gradient descent, meaning that in each iteration small increments $\delta\mathbf{w}^l$ is added to the weight vector $\mathbf{w}^l$ in layer l as seen in equation (2.2)

$$\mathbf{w}^l \leftarrow \mathbf{w}^l + \delta\mathbf{w}^l, \quad \delta\mathbf{w}^l = -\eta\frac{\partial C}{\partial \mathbf{w}^l}, \tag{2.2}$$

where the increment equals the partial derivative of the cost function with respect to the weight of interest multiplied by a learning rate $\eta > 0$ [19]. The idea is to take steps in the steepest descent direction, hence the minus sign in front of the learning rate, until hopefully the global minimum of the cost function is reached [17]. The learning rate translates to the step size and has to be chosen wisely since it is a compromise between the speed of convergence and the accuracy of the learning. A common practice is therefore to employ a decaying learning rate which tends to optimize this trade-off. The biases are updated in a similar fashion, also adding small increments at each iteration.

A more efficient way of optimizing the parameters is to use Stochastic Gradient Descent (SGD). SGD allows for faster convergence since only a single input is used for updating the parameters instead of the entire training set. The step direction will not necessarily point downhill but fluctuate instead, yielding a stochastic path through the weight and bias space. This makes the algorithm less likely to get stuck in local minima [17]. However, due to the stochastic behavior the final performance may be worse than gradient descent. A common practice is therefore to use a small number of randomly chosen inputs, a mini-batch. This approach works as a compromise between gradient descent and SGD since it still ensures fast convergence but also improves the accuracy. Even though SGD only uses one input or a smaller sub-set called mini-batch for updating the parameters, new inputs are chosen at each iteration so eventually the whole training set will be passed to the network. A complete pass of the training data is called an epoch [15].

Momentum is a popular technique to use together with SGD which speeds up the convergence further without sacrificing the accuracy. Instead of updating the parameters based only on the current gradient, it also utilizes past gradients. The step size will increase towards the global minimum if the gradients point in the same direction [19]. However, applying momentum results in yet another hyperparameter to set in terms of the percentage of gradients retained in every iteration.

Adam optimizer is another gradient-based optimization algorithm and utilizes an adaptive learning rate. Learning rate is believed to be the most important hyperparameter [18], but tuning it can be a tedious process. Since Adam computes individual learning rates for each parameter, it will not only entail faster convergence but also make it more robust to the initialization of the hyperparameters. This is an advantage compared to SGD which requires careful tuning [28]. Nonetheless, all methods have their flaws. In the paper by Wilson et al. [29], experiments show that Adam generalize worse to test data than SGD despite its accurate training

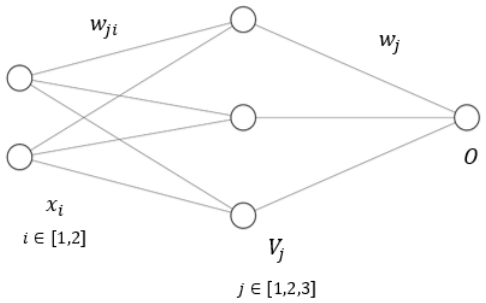


Figure 2.6: Illustration of a simple artificial neural network with inputs x_i, hidden neurons V_j and output O. The weights w_{ji} and w_j are also displayed corresponding to the different layers.

performance.

Backpropagation

In order to efficiently compute the gradients from Equation (2.2) an algorithm known as backpropagation is employed. This technique has been proven very successful for many different methods, including deep neural networks [17]. Backpropagation is about understanding how changes of the weights and biases affect the performance of the network. It propagates errors from the output layer back to previous layers and, while going backwards, the contribution to the overall cost of each weight and bias is calculated. These parameters are then consequently adjusted in such a way that the cost is minimized [19].

The mathematics behind backpropagation is simply the chain rule. For the network in Figure 2.6, the gradient for updating weight w_j leading from neuron V_j in the hidden layer to the output $O = g(B) = g(\sum_{j=1}^{3} w_j V_j + \Theta)$ is computed as

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial O}\frac{\partial g(B)}{\partial B}\frac{\partial B}{\partial w_j} = \frac{\partial C}{\partial O} g'(B) V_j. \tag{2.3}$$

Accordingly, the choice of the cost function, C, and activation function, g, will hence affect the outcome.

Moving backwards to weight w_{ji} leading from the input x_i to the hidden neuron $V_j = g(b_j) = g(\sum_{i=1}^{2} w_{ji} x_i + \theta_j)$, the gradient is now calculated as

$$\frac{\partial C}{\partial w_{ji}} = \frac{\partial C}{\partial O}\frac{\partial g(B)}{\partial B}\frac{\partial B}{\partial V_j}\frac{\partial V_j}{\partial w_{ji}} = \frac{\partial C}{\partial O} g'(B) w_j g'(b_j) x_i. \tag{2.4}$$

The term $\frac{\partial C}{\partial O} g'(B)$ in Equation (2.3) is referred to as the error related to the output layer. As described above it is propagated backwards to the previous layer which Equation (2.4) confirms by showing the presence of that term. Similarly is

$\frac{\partial C}{\partial O} g'(B) w_j g'(b_j)$ the error associated with the hidden layer. The errors are thereby determined recursively in terms of the error from the layer to the right [17]. If the model would have been deeper, containing additional layers, the errors from each layer would have kept propagated backwards all the way back to calculating the error related to the first hidden layer.

When all parameters in all layers have been updated, a new iteration begins. The input data is now fed forward through the network and the cost is calculated at the output using the updated weights and biases. Then again the errors will propagate backwards, updating the parameters once more. These cycles will continue until the training process terminates after a user defined number of epochs.

Cost function

The purpose of the cost function, also referred to as loss function, is to measure the performance of the network. As mentioned earlier, the goal is to minimize the cost and it is therefore crucial that the chosen function is suitable for the upcoming task [18].

Cross-entropy functions measure dissimilarities between the predicted label and the true label distribution and works well for classification tasks. The binary cross-entropy is to prefer for binary classification problems [19] and is defined below:

$$C = -\sum_i y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i). \tag{2.5}$$

Here, y is the true label assigned either 0 or 1 whereas $\hat{y}$ and $(1-\hat{y})$ is the probability of the output belonging to class 0 or 1 respectively. The output of the equation will be close to zero if the prediction is close to the true label for all inputs i [19].

Activation function

Activation functions can be both linear and non-linear but the ones included in the latter group are essential for the learning of complex data such as images. The non-linearity allows for creating complex mappings between the inputs and the outputs. Without them only linear transformations can be applied, reducing solvability of complicated tasks [18]. Nevertheless, adding complexity to the model will increase its ability to overfit on random effects that might only be present in the training data and hence fail to generalize to unseen data.

A common activation function for the output layer is the sigmoid function defined in equation (2.6) [17]

$$\sigma(b) = \frac{1}{1 + e^{-b}}. \tag{2.6}$$

It outputs a single value between 0 and 1 making it suitable for binary classification problems. If more than two classes are included in the classification task, another activation function called softmax is preferable since it outputs probabilities for each class [17].

Rectified Linear Units, ReLU(b) = $max\{0, b\}$, has lately been preferred as activation function in the hidden layers since it is computationally efficient [19]. The ReLU function has piece-wise constant derivatives which makes it faster to evaluate compared to, for example, the sigmoid function [17].

Overfitting

Overfitting is a common consequence when training neural networks. What happens is that the model starts to learn more specific features of the training data instead of just general characteristics. The model will thus become too closely fitted to the current data but will fail spectacularly on previously unseen data [19]. An example of this is the quote by John von Neumann [30]:

> With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

This means that there are four relevant parameters that give all the information needed to capture the shape of an elephant. But the fifth and last parameter is less important since it only wiggles the elephant's trunk.

Deeper networks have a higher tendency to overfit since they contain more neurons which imply a more thorough representation of the fine details of the training set [17]. However, reducing the size of the network is not a preferable approach for avoiding this problem since shallow networks are not as powerful and the efficiency will decrease [19].

To reduce overfitting regularization methods are applied. These can also be combined in order to further improve the performance. L2-regularization or weight decay encourages the network to choose smaller weights by adding an extra term in the cost function that will work as a constraint in the optimization problem. The idea is that smaller weights will reduce the importance of less important features and hence suppress overfitting [17].

Dropout is another popular regularization method that is surprisingly effective despite its simplicity. At each iteration in the training process the network randomly selects a subset of hidden neurons that will be temporarily ignored, preventing both the incoming and outgoing weights of these neurons to be updated [19]. The dropout process corresponds to training different networks in each iteration. These networks will overfit differently but by averaging over them the net effect will hopefully reduce overfitting [19].

Overfitting can also be reduced by simply training on a larger data set since this

will help the network to generalize. However, available data is often limited but an idea is to generate data artificially which is referred to as augmentation. Common image-based augmentation techniques include rotating, flipping, scaling and blurring etc. [17].

2.2.4 Evaluation

Evaluation of neural networks and machine learning algorithms can be performed in several different ways. In this section two main techniques will be explained, performance metrics and visualization. In the concept of performance, different kinds of metrics are included for example accuracy and recall. Visualization comprises saliency maps and heatmaps.

Performance metrics

The most intuitive way to quantify the performance of an ANN is by calculating its accuracy. This measures the total number of correct classifications with respect to the total number of classifications made. This can however be misleading if there is an imbalance between the classes in the data set. Additional evaluation metrics are therefore often essential and the most commonly used are presented in Equations (2.7)–(2.9). The acronyms TP, TN, FP, FN stand for True Positive, True Negative, False Positive, and False Negative respectively.

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2.7}$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN+FP}} \tag{2.8}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{2.9}$$

Recall, also called sensitivity or True Positive Rate (TPR), measures a model's ability to correctly classify the positive samples whereas specificity, the True Negative Rate (TNR), is the ability to correctly classify the negatives. Precision is instead considered as the exactness of the model, it tells how many of the predicted positive samples were actually classified correctly [31].

Other metrics used for evaluating a binary classification model are the Receiver Operating Characteristic (ROC) curve and the precision-recall curve [32]. The ROC is a curve of probabilities where the TPR is plotted against the False Positive Rate (FPR), i.e., one minus specificity, at all classification thresholds. Figure 2.7 illustrates an example of two distributions together with this threshold. When moving the vertical line, the classification ratios change yielding different values for TPR

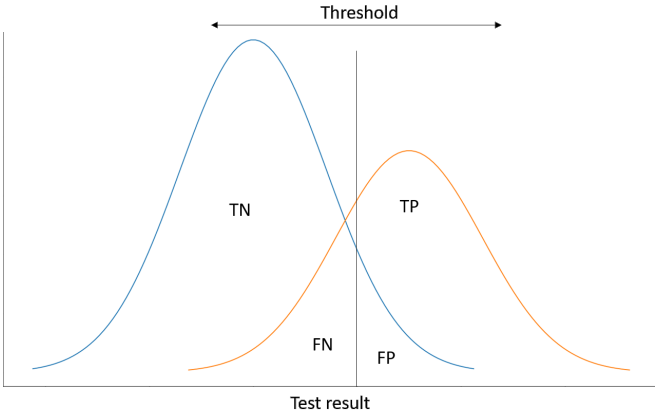


Figure 2.7: An example of distribution curves of a positive and negative class. Moving the threshold will yield different True Positive Rate (TPR) and False Positive Rate (FPR) from which a Receiver Operating Characteristic (ROC) curve can be calculated. The acronyms TP, TN, FP, FN stand for True Positive, True Negative, False Positive and False Negative respectively.

and FPR from which the ROC curve can be made. More separated distributions implies a better classifier [33]. The Area Under the ROC Curve (AUC) value indicates how good the classifier is at distinguishing between the classes and ranges from 0 to 1. Hence, a larger value equals a larger area which corresponds to a better model. Similarly, the precision-recall curve is constructed by plotting the precision against the recall at all possible thresholds. This curve outputs an Average Precision (AP) value between 0 and 1 which, as the name suggests, is the average precision for the different thresholds [34].

Visualization

One of the disadvantages with deep learning models is that they are usually considered to be 'black boxes'. This means that it is, in general, difficult to distinguish what the model actually learns and what types of features it interprets as important. Though for CNNs there are methods to understand what the network finds interesting, two of these are saliency maps and heatmaps.

Saliency maps highlights areas of an image with respect to a given class. This is done by calculating a class score function $S_c(I)$ based on an image I_0 and a class c and then ranking the pixels of the image I_0 based on their influence on the class score. Class score can be calculated in different ways depending on what is suitable for the model [35]. An example of a class score is the linear score model which can be calculated for a class c as in Equation (2.10):

$$S_c(I) = w_c^T I + b_c. \tag{2.10}$$

where w_c is the weight vector and b_c is the bias for the model [35].

Heatmaps, on the other hand, are created from a technique that is called Class Activation Map (CAM) visualization. This technique introduces CAM in the image by creating a 2D grid belonging to a certain class where the grid visualizes the gradient of the class score [27]. The class score is obtained from the last convolutional layer and the gradient is calculated with respect to the feature map from this layer. These gradients are then averaged over the whole image to obtain the importance of certain areas corresponding to the class. This creates a coarse heatmap with the same size as the last feature map which can be enlarged and applied to the original image to visualize important areas for the specific class in the image [36].

2.2.5 Keras

`Keras` is a deep learning framework written in Python that provides an easy and user-friendly way to build and train deep-learning models. It supports CNNs, recurrent neural networks and other deep learning models with both single and multiple input and outputs. It is a model-level library which means that it only creates the building blocks of the model and needs to be supplemented with a low-level library to handle operations like tensor manipulation and differentiation. A low-level library is called *backend engine* and has a well-optimized tensor library to handle the previously mentioned operations. Keras is currently compatible with three different backends; `Tensorflow`, `Theano` and `CNTK` where `Tensorflow` is developed by Google, `Theano` by the MILA lab at Université de Montréal and `CNTK` is developed by Microsoft [27]. `Tensorflow` has `Keras` as its official frontend which means that it works seamlessly with its workflow [37].

The `Keras` framework is distributed under open MIT license which means that it can be and has been used in several commercial projects by all ranges of users, from students to larger companies [27].

3

Method

This chapter contains four main sections. First, the study is placed in a context concerning the employed research methodology. Then sections on Data, Implementation and Evaluation are presented. In the Data section the provided data sets will be described together with the preparation of them. The section on Implementation covers the chosen deep learning framework, network structure and training details. Lastly, the different methods used for measuring the performance of the network are presented in Section 3.4 (Evaluation).

3.1 Research methodology

In the ABC framework for Software Engineering Research by Stol and Fitzgerald [38], research is classified by the level of *Obtrusiveness* and *Generalizability* within the categories "Actors" (A), "Behaviour" (B) and "Context" (C). Obtrusiveness refers to how much the researcher intrudes on the research setting, the data collection, and how much the data is manipulated. Generalizability concerns the result of the outcome and how it can be applied in other settings and if any statistically generalizing conclusion can be drawn from the research [38].

Stol and Fitzgerald [38] grades eight different research methodologies after how obtrusive and generalizable they are for each ABC. For this project the methodology is applied to less obtrusive research since the data is provided by VGR without any intrusion in the data collection. These characteristics apply to both "Field studies" and "Sample studies". Field studies involve researching in a real-world setting for studying a specific phenomenon which means that it will capture realistic context, C. The research will be very specific for this setting and thereby the generalizability over the actors, A, is low. In the concept of Sample studies the generalizability over A is high while C is low since the research setting is neutral and same for all samples. Therefore, the most suitable research methodology for this project is Sample studies since it gives a high generalizability for the whole population and the research setting; the ultrasound examination, is the same for all the patients in the screening program [38].

3.2 Data

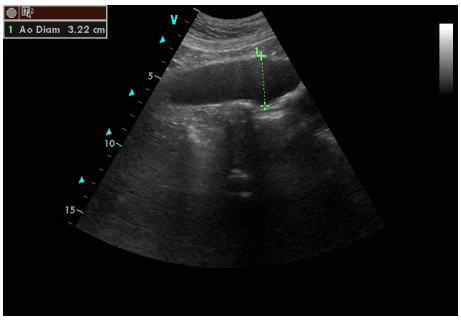
The data used in this project was acquired from a prospective cohort study called Gothia 3A made by VGR and is also a part of the screening program mentioned in Section 1.1. This database has ethical approval and the participants have given their consent to share their data for research purposes. The data is completely anonymized and can not be tracked back to a specific person.

The data consists of ultrasound images of the abdominal aorta generated at the screening visit together with complementary information about the patients such as smoking habits, medications and blood pressure. Table 3.1 presents the full list of parameters from the database that were used as inputs. Note that the explanations in Table 3.1 correspond to the input parameters after pre-processing. Each patient is related to one ultrasound image that is taken in either the sagittal plane, from the side, or the axial plane, from above. Figure 3.1a and Figure 3.1b illustrates examples of these different angles.

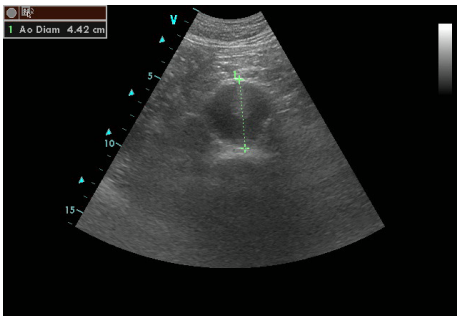
Table 3.1: The table shows the patient data used as inputs to the non-image branch of the network together with an explanation.

Patient data	**Description**
Aortic diameter	Largest diameter of the aorta in mm.
Smoking	Years of smoking, regardless of the patient being an active smoker or not.
Snus	Years of using snus, regardless of the patient is active or not.
Trombyl	Anticoagulant medication. Yes or no.
Statin	Cholesterol lowering medication. Yes or no.
Blood pressure	Patient is classified as having either high or normal blood pressure.

Of a total of 204 participants, 142 were diagnosed as sick. The rest, 63 men, were considered healthy based on their aortic diameter, however 11 of these have a sub-aneurysmal dilatation in between 25–29 mm as explained in Section 1.1. This sub-group was removed from the data and set aside until the final classification. The rest of the patients were split into training, validation and test sets as seen in Table 3.2. These splits were based on finding a good ratio between positive and negative images to help the network learn features from both classes and classify them equally likely. It was also considered important to reach a final split after augmentation of 80/20 between the training and validation set.



(a) An example of a sagittal ultrasound image.



(b) An example of an axial ultrasound image.

Figure 3.1: Examples of the provided ultrasound images.

Table 3.2: This table presents how many samples that were used for training (before any augmentation techniques have been applied), validating and testing the network. The ratio between the positive and negative classes (P/N) are displayed within the parentheses.

	Pre-training set (P/N)	**Final set (P/N)**
Training	232 (18/214)	76 (56/20)
Validation	143 (18/125)	96 (70/26)
Test	28 (9/19)	21 (15/6)

Additional ultrasound images of the abdominal aorta were also received, originating solely from the screening program and were not part of the prospective study. A total of 403 images were used of which 45 were classified as sick and 358 as healthy. These were used for pre-training of the CNN described in Section 3.3.1 and were also split into the three subsets according to Table 3.2. The division was justified similarly as above.

3.2.1 Preparatory tasks

The ultrasound images contained the largest measured aortic diameter identified by the technician during the screening visit. This can be seen in the top left corner of the images in Figure 3.1. Since they were used as input parameters to the neural network, they needed to be collected. The text extraction was carried out using the existing Optical Character Recognition (OCR) tool for Python, that is `Pytesseract` which runs on Google's Tesseract-OCR engine.

In this project, only supervised learning was employed meaning that the data sets had to be annotated. The pre-training data was therefore labeled as one of the two classes, either 0 or 1 depending on if the patient was declared healthy or diagnosed as sick. These classes will also be referred to as the negative and positive class further on in this report. This decision was solely based on the aortic diameter with 30 mm as limit. A slightly modified annotation was employed for the data set used

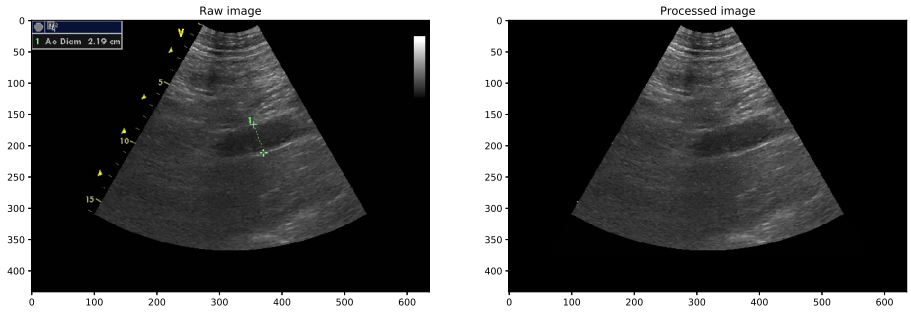


Figure 3.2: The left image shows the raw ultrasound image while the right shows the same image after clean-up.

for the final model. The limit for a healthy classification was now lowered to 25 mm instead of 30 mm. The reason for doing this was that the patients within the interval of 25–29 mm, have a sub-aneurysmal aortic dilatation. These cases were not labeled in order to see whether the network would classify the sub-aneurysms as sick or healthy and since this sub-group did not take part in the training phase, labels were not required.

3.2.2 Pre-processing of data

Before feeding images to a CNN it is preferable to select and enhance the region of interest and eliminate any other possible distractions.

In addition to the imprinted value, there was a colored dashed line located on the aorta representing that measurement and a scale along the side of the ultrasound image that also needs to be removed. Similar to the text extraction another Python tool was utilized to reduce the appearance of the dashed line, namely `OpenCV`. To further help erase it, the final images were converted to gray scale. By cropping the images in a trapezoidal shape the remaining imprinted measurement and scale were removed. The result of these steps are demonstrated in Figure 3.2. Lastly, the images were re-sized to the shape (224, 224, 3) since a pre-trained network was used, which required specific size and dimension.

Pre-processing the non-image data included simplification of the smoking habits and the blood-pressure data. The smoking habits were converted by combining two categories answering if the patient smokes today and if so for how many years as

well as if the patient has been smoking and if so for how many years. The final category will now instead only answer for how many years the patient has been smoking independent of his current usage.

The blood pressure data was translated into a binary category where 1 represents high blood pressure and 0 normal or low. The choice of combining normal and low into one category was due to the fact that only high blood pressure will affect the growth of an aneurysm. Finally, the non-image data was also normalized and scaled to unit variance, N(0,1), by removing the mean due to its varying scale across the six inputs.

Augmentation

Partly because the training data was imbalanced between the classes and partly because it was relatively small, various image augmentation techniques were applied to even out the difference and to expand the data set. This will also help to prevent overfitting as discussed in Section 2.2.3. Horizontal flipping, histogram equalization and adjustment of the brightness were chosen as methods and applied to the whole negative set. To prevent the data set being biased and to expand it further, a smaller amount of the positive images were also augmented. There were now a total of 776 images in the training set with a positive/negative-ratio of 456/320. Figure 3.3 demonstrates the results of the different techniques that were employed.

Similarly, the same augmentation techniques were applied to the training data used for pre-training. However, in that case the negative class included a lot more samples than the positive, thus only a small part of the negative set while the whole positive set were augmented. This resulted in a total of 700 images whereof 288 were positive and 412 negative.

For Figures (B) and (C) in Figure 3.3 `OpenCV's` modules `flip` and `createCLAHE` were used and for Figure (D)–(F) the package and module `skimage.exposure.adjust_gamma` were used. The `createCLAHE` module applies something that is called Contrast Limited Adaptive Histogram Equalization (CLAHE) meaning that it adjusts the contrast of certain parts in the image depending on the surrounding area.

There are several other types of augmentation techniques that could be employed, like vertical flipping, rotation, zooming, sharpening, or adding noise. But since ultrasound images have the characteristic trapezoidal shape and are quite noisy from the beginning, those methods were not deemed to be suitable in this case.

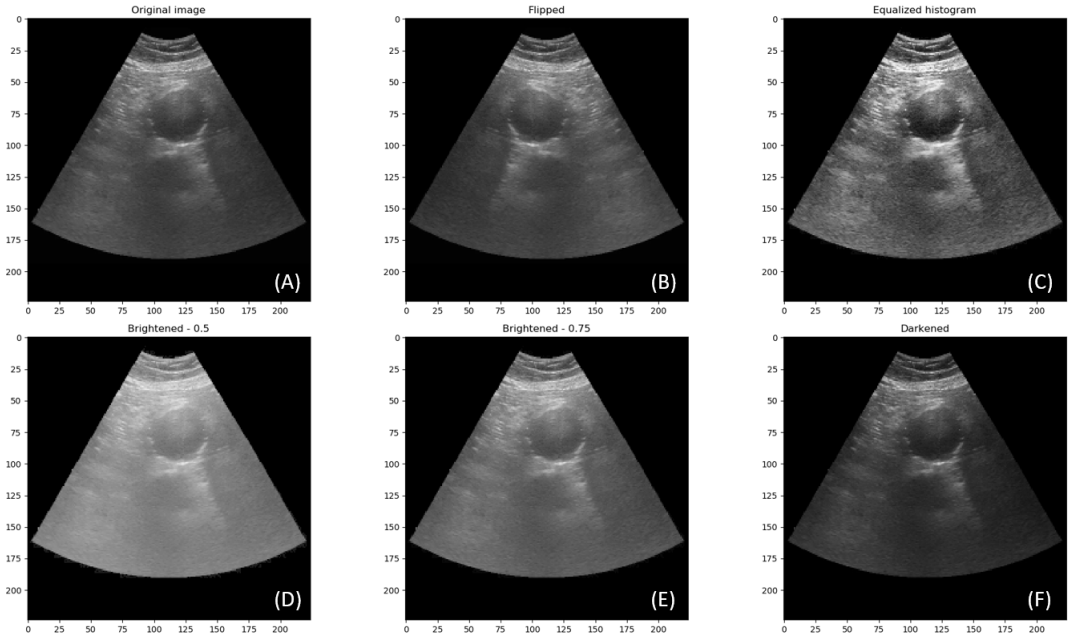


Figure 3.3: The three different augmentation techniques used. Figure (A) shows the original image, Figure (B) the flipped one and Figure (C) the histogram equalized image. Figure (D), (E) and (F) shows different levels of brightness adjustment.

3.3 Implementation

This section describes the implementation of the network used in this project and the training procedure. Since two modalities, ultrasound images and patient data were used as inputs, the neural network consisted of two branches before being combined to a single output. A CNN was implemented for the image branch while a network including only one fully connected layer was used for analyzing the non-image data. The structure of the full network was inspired by Xu et al. [7] algorithm for diagnosing Cervical dysplasia and Bonnett [39] for classifying e-commerce products.

All implementations were done using Python's deep learning framework `Keras` version 2.2.4 with `Tensorflow` version 1.10.0 as backend. The training was then carried out on a computer with a NVIDIA Titan X graphics card with a memory of 12 GB.

3.3.1 Pre-training

Since the provided data set was relatively small even after augmentation, it was decided to use a pre-trained network as basis for the image branch and then fine-tune it to fit the classification task in this project. From `Keras Applications` module, three different network structures with pre-trained weights and biases were downloaded: InceptionV3, VGG16 and VGG19. These were all initially tested to

get an overview of their performance and a final network was selected for further tuning.

VGG19 qualified to the more comprehensive tuning and was initialized with the weights and biases obtained from training on the ImageNet database. Since the network should perform a binary classification, the activation function in the output layer was changed from softmax to sigmoid in order to only generate one output. The cost function was then set to binary cross-entropy.

Besides being pre-trained on ImageNet, the network was trained on the additional ultrasound images obtained from the screening program. This helped the network learn more image features related to the final classification task. Different choices of hyperparameters were employed to find which setup yielded the most satisfactory results and the corresponding weights and biases were saved. As optimizer both mini-batch SGD and Adam were tested while altering the values of the remaining hyperparameters. Dropout and L2-regularization were both applied to the second to last fully connected layer of the structure. The complete list of adjusted hyperparameters is:

- Learning rate
- Learning rate decay
- Momentum
- Mini-batch size
- Epochs
- Trainable layers
- Dropout rate
- L2-regularization

3.3.2 Combining and training the final network

As described earlier the final network consisted of two branches, one fully-connected neural network and one CNN with a structure as presented in the previous section. This structure was inspired by the networks used for Bonnett's e-commerce application [39] and Xu et al.'s algorithm for diagnosing cervical dysplasia [7]. The complete architecture of the final network is shown in Figure 3.4.

The merging of the two branches is done using the `Keras` layer `Concatenation` and by applying the technique of early fusion. This means that the branches' output features are merged together without performing any classification. The classification will instead take place after the features have passed through an additional fully-connected layer. Both of the branches were designed to output the same number of features, six, which corresponds to the number of different input parameters of the patient data. As above, different settings of the hyperparameters were tested, but now the image branch was initialized with the saved pre-trained weights and biases. L2-regularization was applied to the last three fully connected layers of the image branch while dropout only was applied to the last two of them to replicate

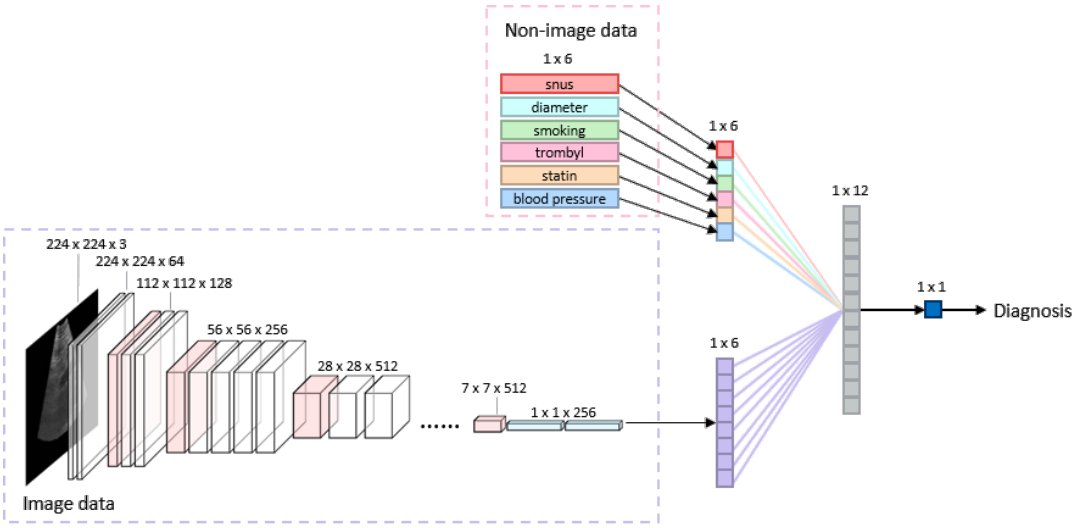


Figure 3.4: The final structure of the combined network. The image branch is a Convolutional Neural Network based on the VGG19 structure presented in Figure 2.5 and the non-image branch consists of one fully-connected layer. These are both combined into an output layer performing binary classification.

the implementation in Bonnett [39].

3.4 Evaluation

During the training phases the network's performance was monitored every epoch by four different metrics: loss, accuracy, precision, and recall, in order to observe how they changed over time. The weights and biases related to the best metrics on the validation set were saved for further evaluation. For this part a ROC curve with a corresponding AUC-value was made using the `scikit-learn` package. An optimal threshold was determined that would maximize the TPR and minimize the FPR of the validation set. A precision-recall curve was also built of the validation set and provided an AP-value using the same `scikit-learn` package. Another optimal threshold was obtained from this curve that maximizes both the precision and the TPR.

Heatmaps and saliency maps were used to see what the network found interesting in the images. These were calculated by CAM and from the class score respectively. A technique called permutation importance was employed to identify which of the input parameters that had the biggest impact on the output. Each column of the input data was randomly shuffled while the rest were left untouched. The model then classified this modified data and the new accuracy was compared to the original value. Ten shuffles was carried out per column allowing a mean value of how much the metrics decreased to be calculated. This function was built from scratch but influenced by the pre-defined function found within the `eli5` package. There, the number of shuffles was set to five by default but in this case it was slightly increased

Table 3.3: This table shows statistics from the meta-studies in Section 1.5.2 that will be used as comparison to the results obtained from classifying the sub-group. The percentages correspond to how many of the men's sub-aneurysm that progressed into a full-size aneurysm with a diameter ≥ 30 mm within 5 and 10 years, respectively.

Number of years	**Full-size aneurysm**
5	55–88 %
10	96 %

to obtain more precise estimates.

After evaluating the final network's performance for different settings, the best set of hyperparameters was used for the classification of the test set. Another classification using the thresholds obtained from the ROC-curve and the precision-recall curve was also made to see if the results improved. The network was also applied on the group with a sub-aneurysmal aortic dilatation to try to predict which men will most likely develop a full-size aneurysm. This prediction was then compared to the results from the meta-studies mentioned in Section 1.5.2 and a summary of the most relevant statistics is presented in Table 3.3. Marcus Langenskiöld, vascular surgeon at Sahlgenska University Hospital, was consulted for verification of the results.

4

Results

This chapter presents the results obtained from training and evaluating the network during both pre-training and the final training. Manual tuning of the hyperparameters was applied in both cases to achieve a satisfactory performance based on the different evaluation metrics. By a satisfactory performance a high accuracy, precision and recall paired with a low loss is intended.

4.1 Pre-training

The values of the hyperparameters that yielded the most satisfactory performance during pre-training are displayed in Table 4.1.

The set of hyperparameters was chosen as most optimal based on the precision and recall for the validation set since these are important when working with medical diagnoses. This decision was confirmed by Marcus Langenskiöld, mentioned in Section 3.4. A high value of these metrics imply that a few number of both sick and healthy people are misclassified. Also, the accuracy and the loss tended to stay around the same levels in most settings while precision and recall varied within a larger interval.

Table 4.2 presents the corresponding results of the evaluation metrics for the train-

Table 4.1: Hyperparameters used for both pre-training and for the final model.

Hyperparameters	**Pre-training**
Learning rate	1×10^{-4}
Decay	1×10^{-5}
Optimizer	SGD
Momentum	0.5
Mini-batch size	32
Total number of epochs	50
Trainable layers	Upper 6
Dropout	0.2
L2-regularization	0

Table 4.2: Results for training, validation and test from pre-training of the Convolutional Neural Network obtained from the last epoch.

Metric	Training	Validation	Test
Accuracy	100%	95.8%	93.0%
Loss	0.013	0.168	0.290
Precision	100%	85.8%	100%
Recall	100%	79.9%	78.0%

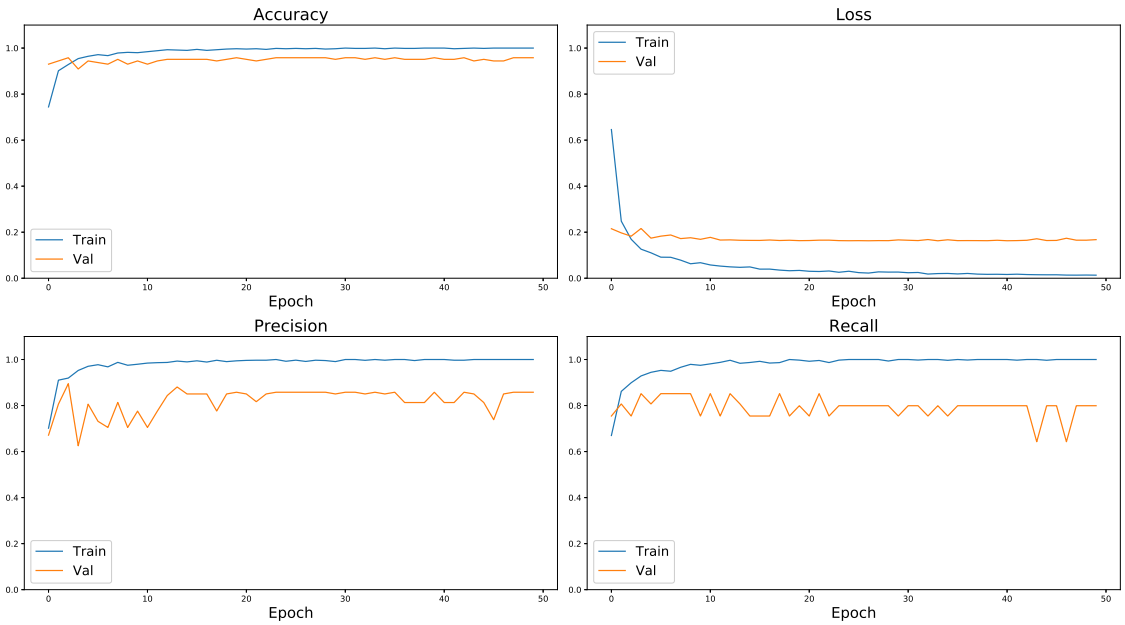


Figure 4.1: The plots show the performance metrics during the pre-training of the convolutional neural network when using the best set up of hyperparameters.

ing, validation, and test sets.

Figure 4.1 shows the trends of the different performance metrics during training using the setup in Table 4.1. As seen, both the accuracy and the loss are relatively stable while minor spikes are present for precision and recall. In Appendix A.1 (Figure A.1), the performance metrics monitored over time for pre-training with another setup of the hyperparameters can be found.

4.2 Final model

As with the pre-training, the final model was tested with different setups of the hyperparameters and evaluated on the validation set. The values of the hyperparameters yielding the most optimal performance are summarized in Table 4.3, and Figure 4.2 illustrates the trends from using this setup during training. Compared to the results from pre-training there are stronger fluctuations present for all metrics

Table 4.3: Hyperparameters used for the final model.

Hyperparameters	**Final training**
Learning rate	1×10^{-5}
Decay	1×10^{-6}
Optimizer	Adam
Mini-batch size	32
Total number of epochs	100
Best epoch	65
Trainable layers	All
Dropout	0.5
L2-regularization	$1 * 10^{-4}$

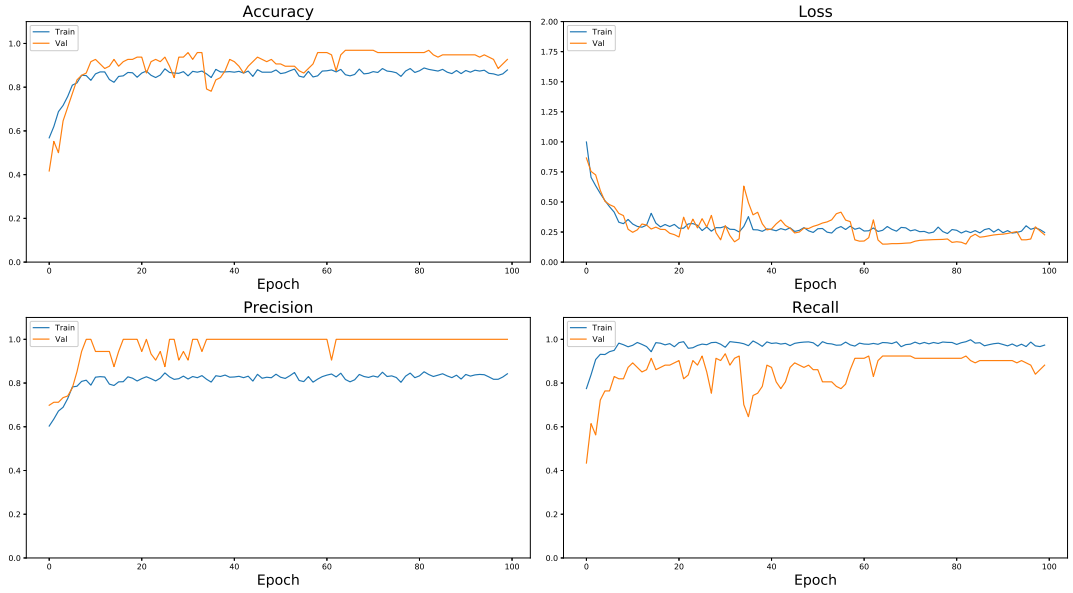


Figure 4.2: Visualization of the training performed using the hyperparameters listed in Table 4.3. The evaluation metrics are plotted for every epoch for both the training and validation set.

now. Additional trends from training with other setups of hyperparameters can be found in Appendix A.2, (Figures A.2–Figure A.4).

Epoch 65/100 performed best during training with the aforementioned hyperparameters and the corresponding weights and biases were saved and used for evaluating the test set. The resulting performance metrics for each sub-set are shown in Table 4.4. Even though some epochs from the training presented in Appendix A.2 generated better values of the performance metrics than the ones presented in Table 4.4, the overall trends were more satisfactory for the chosen setup. This was especially true for the validation loss.

Below follows images that were misclassified in the test and validation set. Figure 4.3 shows a sick patient that has been misclassified as healthy whereas Figure 4.4 demonstrates the healthy images misclassified as sick. It seems that the network tends to

Table 4.4: Resulting evaluation metrics for the training, validation and test set from evaluating the final network obtained from epoch 65/100.

Metric	Training	Validation	Test
Accuracy	85.2%	96.90%	95.2%
Loss	0.296	0.150	0.849
Precision	80.6%	100%	93.8%
Recall	98.4%	92.4%	100%

Figure 4.3: Image from the test set. A healthy patient misclassified as sick.

misclassify patients whose images were taken in the sagittal plane.

Turning now to the evaluation using the ROC-curve and precision-recall curve. Figure 4.5a demonstrates the obtained ROC-curve together with the AUC-value of 0.99. The most optimal threshold was calculated to 0.38 which coincides with the distribution curves of the validation set in Figure 4.5b. It is clearly seen that the number of misclassifications are minimized using this threshold, no FNs and only two FPs occurred. However, when this threshold was applied for reclassifying the test set, the performance metrics did not change. The precision-recall curve is illustrated in Figure 4.5c with its average precision-value of 1.00. Calculating the most optimal threshold from this curve resulted again in 0.38.

The results from visualization using heatmaps and saliency maps are demonstrated in Figure 4.6. Figure 4.6a shows an example of a correctly classified healthy patient with the heatmap rendered from the network and the image's corresponding saliency map. An example of a correctly classified sick patient is displayed in Figure 4.6b.

When the patients were misclassified the heatmaps could instead look like in Figure 4.7. Here Figure 4.7a shows the heatmap for a healthy patient classified as sick while the heatmap for a sick patient misclassified as healthy is shown in Figure 4.7b.

Figure 4.8 shows the images' saliency maps from the misclassified patients mentioned in the previous paragraph. The misclassified healthy patient is shown in Figure 4.8a,

Figure 4.4: Images from the validation set that were misclassified as healthy.

Table 4.5: The values represent how much the model's performance decreased in average in terms of accuracy with a random shuffling.

Feature	**Train**	**Validation**	**Test**
Images	0.470 ± 0.022	0.240 ± 0	0.238 ± 0
Aortic diameter	0	0	0
Smoking	0	0	0
Snus	0	0	0
Trombyl	0	0	0
Statin	0	0 ± 0.004	0.010 ± 0.019
Blood pressure	0	0	0

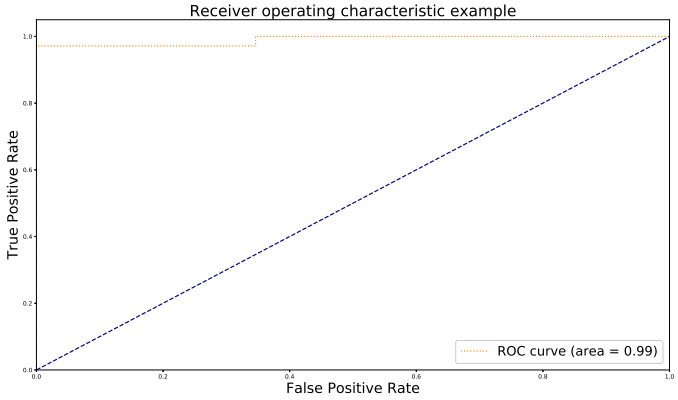
and Figure 4.8b demonstrate the saliency map of the misclassified sick patient.

Based on the heatmaps in Figure 4.6 and Figure 4.7, for a healthy classification the network tends to miss the aorta completely while it registers at least parts of it for a sick classification.[1] The saliency maps are slightly highlighted in the area of the aorta regardless of classification as seen in Figure 4.8. Though the result is varying and some of the saliency maps' highlighting is less distinguishable, examples of these are shown in Appendix A.4.

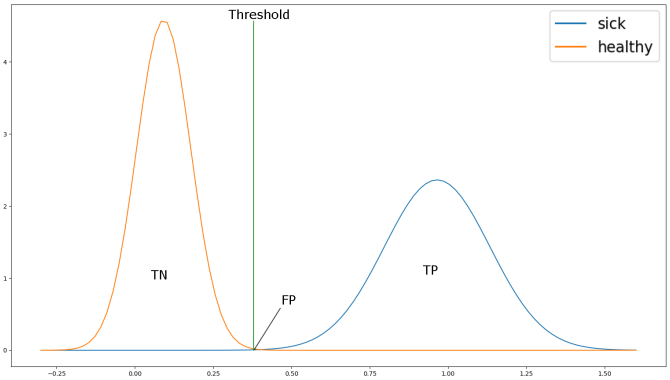
The results from the permutation importance are summarized in Table 4.5. The ultrasound images appeared to have the highest impact out of all input parameters since they decreased the accuracy the most when shuffled. Remaining inputs did not seem to affect the results at all.

Finally, the network was used to predict if the patients with sub-aneurysms would develop a full-size aneurysm. Out of these 11 patients all of them were predicted to do so, i.e. 100%. This percentage does not agree entirely with any of the statistics in Table 3.3 obtained from the meta studies.

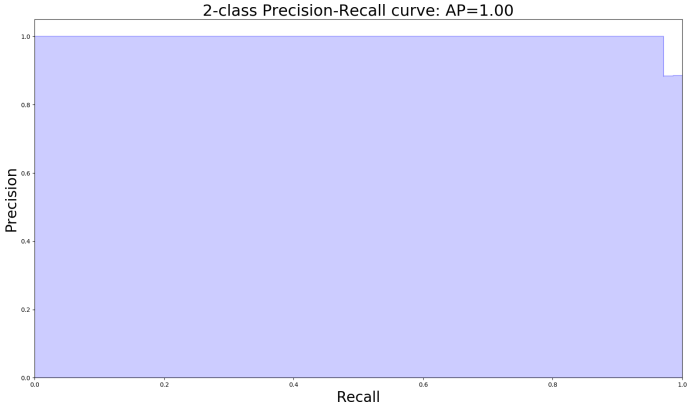
[1] Additional heatmaps can be found in Appendix A.3.



(a) ROC-curve and corresponding AUC value generated from the validation set.

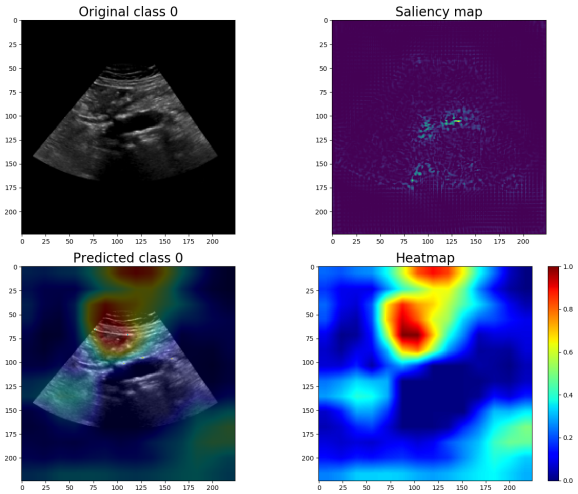


(b) Distribution curves of sick and healthy classes of the validation set with the most optimal threshold located at 0.38.

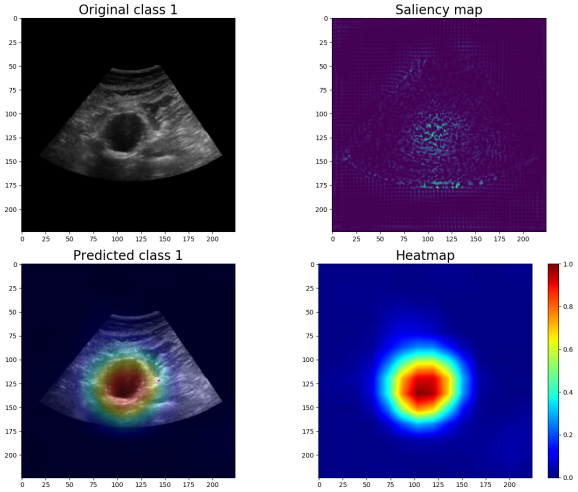


(c) Precision-recall curve of the validation set with its Average Precision (AP) value of 1.00.

Figure 4.5: Graphical representations of the evaluation for the final model. Figure 4.5a shows the ROC-curve together with its corresponding AUC-value while Figure 4.5b demonstrates the distribution curves of the sick and healthy classes. Finally, Figure 4.5c illustrates the precision-recall curve.

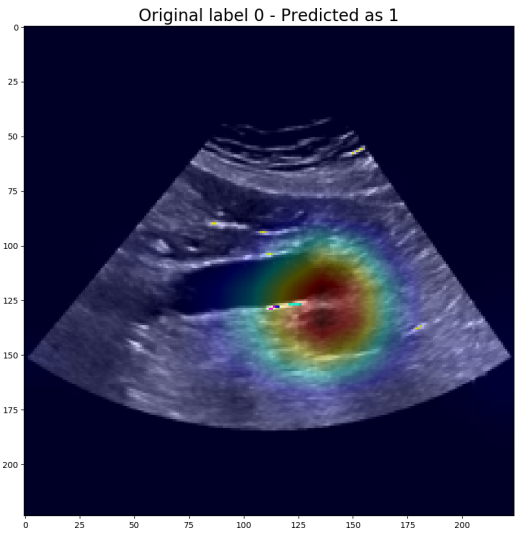


(a) Heatmap and saliency map from a correctly classified healthy patient.

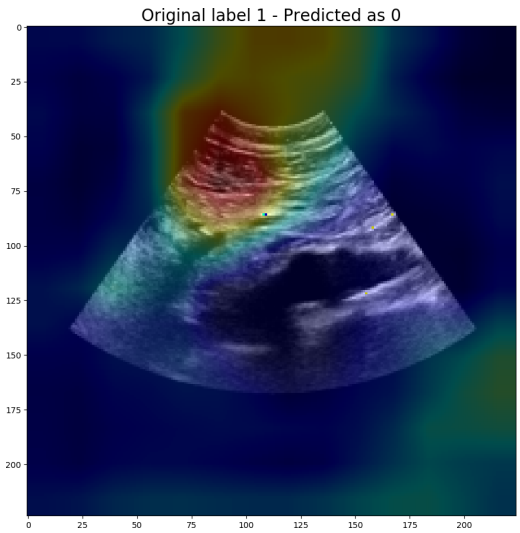


(b) Heatmap and saliency map from a correctly classified sick patient.

Figure 4.6: Heatmaps and saliency maps for correctly classified patients.

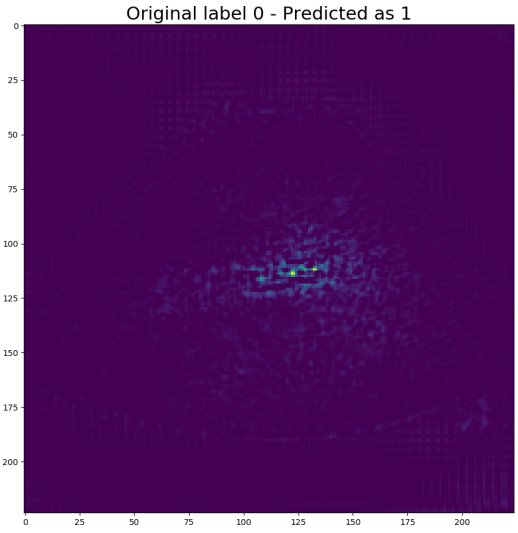


(a) Healthy patient misclassified as sick.

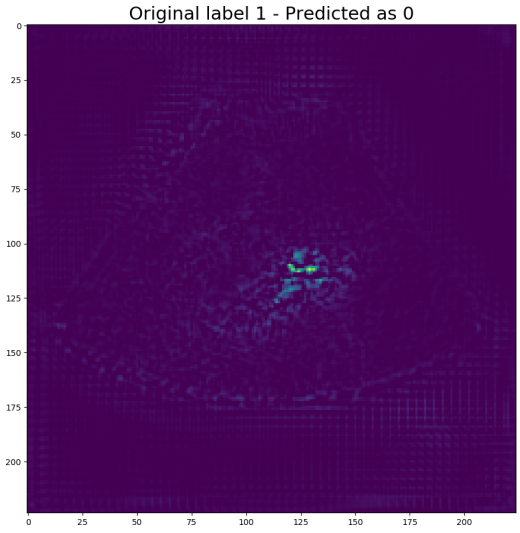


(b) Sick patient misclassified as healthy.

Figure 4.7: Examples of heatmaps from misclassified patients.



(a) Healthy patient misclassified as sick.



(b) Sick patient misclassified as healthy.

Figure 4.8: Examples of saliency maps from misclassified patients.

5

Analysis

This chapter will provide an analysis of the results from the previous chapter. The pre-training step and the final model will be discussed here.

5.1 Pre-training

During pre-training, the validation set contained spikes in the trends for both precision and recall as seen in Figure 4.1. The classification of this set is also performed using mini-batches and since they are chosen randomly, several images that will be misclassified might end up in the same batch. This could have a degrading impact of these metrics resulting in the visible spikes. However, the overall trends and the performance metrics from the pre-training were promising and provided a solid foundation for constructing and training the final network.

5.2 Final model

As seen from the general trends in Figure 4.2, and from the performance metrics acquired from the best epoch in Table 4.4, the network succeeded to classify the data from both the validation and the test set more accurately and with higher precision than for the training set. At a first glance this might seem strange since the network has been optimized according to the training data. But one possible explanation is the dropout regularization. Dropout basically forces the network to become a collection of weaker classifiers. One individual weak classifier will have a poor predictive performance compared to when they are used altogether as an ensemble model. This technique is only applied during training making the corresponding accuracy suffer. When classifying the validation set or the test set, the dropout is turned off allowing weak classifiers to be combined. Hence the accuracy will improve. Another explanation is that the training set is a lot larger than the validation and the test set. It is therefore more likely to contain patients that are harder to classify and those possible misclassifications will decrease the accuracy.

Regarding the fluctuations of the validation data seen in Figure 4.2, it is most likely

due to the small number of samples included in the set compared to the training set. But an increase of the data in the validation set would have led to a decrease in training data which was undesirable. Even though additional augmentation techniques could have been applied in order to increase the training set, it is preferable to train on as many unique images as possible. A larger variety of the images implies a better generalization to new data. Despite the oscillations, the overall trends were satisfactory. Since the validation set is relatively small, a local minimum is most likely just noise due to the randomness when choosing the mini-batches. If the model would have been modified based on that, it might overfit and, hence, fail to generalize to the test set.

The misclassifications seen in Figures 4.3–4.4, that seemed to be based on the elongated shape of the aorta, might be due to unevenly distributed training, validation and test sets. When the data set was divided into the sub-sets, the ratio of sagittal/axial images was never controlled for. The division was thereby unbiased but if a predominant amount of axial images ended up in the training set, the network might run a higher risk of failing to generalize to the sagittal images in the validation and test set and, thus, classify them incorrectly. The misclassifications could also be due to the fact that ultrasound images are very subjective; what is captured in the images depends on the technician that controls the probe. How close the aorta is to the probe or if the technician chooses to zoom will affect the appearance of the aorta and might even mislead the network to perceive it as larger or smaller than it actually is.

Evaluation using the ROC-curve and precision-recall curve resulted in almost a perfect classification algorithm due to the high values of the AUC and AP. However, these results might be a bit deceptive due to the relatively small validation set. The algorithm does not necessary have to perform as well if more data was available. When reclassifying the test set using the optimal threshold obtained from the validation set, nothing changed. Dropping the threshold from the default level at 0.5 to 0.38 made no improvements since the model turned out to be very confident in its decisions, i.e. all outputs are either very close to 0 or almost equal to 1.

From the heatmaps, a classification pattern can be seen. A sick classification implies that the network found the aorta or parts of it, and a healthy did not find it at all. This can be due to the darkness of the aorta and its rounded shape, which would give a sharp gradient at the border of it. The network will then connect this shape to the disease and classify the patient as sick. So in other words, the network is quite good at finding aortas with a round shape as the one in Figure 4.6[1]. The results from the saliency maps were harder to interpret. Sometimes the maps were good at highlighting which areas the aorta was located in, as seen in Figure 4.8, but in other cases it showed nothing in particular as in Appendix A.4 (Figures A.6b–A.6c). The explanation for this was not fully understood and therefore more attention was directed towards the results from the heatmaps.

The results from the permutation importance demonstrated that the ultrasound

[1]This is also valid for the images in Appendix A.3, Figure A.5a, A.5b and A.5d

images have the largest impact on the network's decision while the rest of the input parameters are negligible. This however, indicates that the two modalities being fed to the network are not equally important even though that was intended.

When classifying the sub-group using the trained network the entire set was diagnosed sick, i.e., 100%. This does not entirely coincide with the statistics from the meta-studies presented in Table 3.3, though the percentage of progressed subaneurysm after ten years (96%) is close to the obtained results. Since the positive samples in the training data were predominant, the network might be more inclined to vote for a sick classification rather than a healthy and especially since the majority of the diameters were closer to 30 than 25 mm. However, the result would most likely have been different if the sub-group contained a lot more patients. Nevertheless, since no knowledge about these patients actually developing an aneurysm is given, it is impossible to state whether or not the obtained result is inaccurate.

6

Discussion

This thesis investigated three main questions regarding the prediction of further development of sub-aneurysmal aortic dilatation, most important image features and the significance of the different input parameters. The first question covers the comparison between the obtained result from classifying the sub-group in this project and the ones summarized from the meta-studies. As presented, all men in the sub-group were classified as sick, which does not entirely agree with the percentages provided by the studies. However, as discussed in Section 2.1 an aneurysm grows faster the larger it is. This implies that the network classifying all sub-aneurysm as sick is not completely unfeasible as the sub-aneurysm would behave and grow the same way as a small full-size aneurysm, like an aneurysm of 30–35 mm.

The second question regarded the investigation of important image features and provided a lot of valuable information. The heatmaps used for this purpose showed that the network tends to find sick aortas but miss the healthy ones. Furthermore, the network also seems to base its decision on the shape of the aorta rather than the actual diameter. It can be discussed whether this gives an accurate prediction since the shape mostly depends on from what plane the image was captured in and not the actual anatomical shape of the aorta. This brings up the topic of subjectiveness of ultrasound images once again since the shape of the aorta also depends on the angle from which the ultrasound operator finds the largest diameter.

Permutation importance was employed to answer the last question about which of the input parameters that turned out to be most significant. The results stated that the image was the most important, which was interesting since ultrasound images are very subjective as mentioned above. Contradictory to how the real diagnosis is made, the patient data, and especially the aortic diameter, is not affecting the outcome. Smoking is considered to be the most important risk factor according to [2], thus it was also surprising that it did not receive any attention from the network. Once again, this outcome was not completely unexpected as the ultrasound images are what change the most between the patients due to them being subjective.

To summarize the answers and connect to the aim of this thesis; as stated above the result of all sub-aneurysms being classified as sick is not unreasonable. Therefore, the approach of using a neural network that combines information from two modalities seems promising even though the lack of follow-up data makes it hard to confirm the classifications.

The investigation of important image features and input parameters was an essential part of this thesis. In a future algorithm implemented in the health care it would be necessary to understand and explain to the patients why they were diagnosed as sick or declared healthy. This is very important across the whole medical domain and not just in this project. The software should provide explanations of why it predicted as it did.

For example, if two patients with similar symptoms are classified differently, the doctors can check that it makes sense and thereby be convinced. If an algorithm makes a prediction or suggestion, the doctors need to be able to motivate why in order for it to be reliably. The implemented algorithm in this project is not quite there yet, it needs to be further developed to be more reliable in its decisions and be better at explaining the outcomes. It is not enough by just showing that it found the aorta and therefore the patient is sick since, hopefully, all examined patients have an aorta.

6.1 Justification of methods

One of the most important factors to have in mind when working with medical data is patient privacy, implying that there are a lot of legal processes that need to be handled. These processes are time consuming and in our case they resulted in receiving the data late in the project and, thus, limiting the time for fine-tuning of the hyperparameters. In addition, this meant that the testing of different backends and frontends was excluded from the project.

Also due to the lack of time, most of the inspiration for the network's structure was retrieved from Bonnett's blog post about merged networks for classification in the e-commerce business from [39]. The only thing changed from this structure is the number of output features from each network. This part was instead taken from the network structure by Xu et al. [7], where the number of features from each branch were the same. The aim of this was to make the two modalities equally important.

VGG19 was implemented as an image branch in this project, with the weights corresponding to the training on ImageNet. This meant that the network already had learned general image features. It was then trained on additional ultrasound images to teach the network more ultrasound specific image features.

The splitting of the data set was performed to yield a good trade-off between sick and healthy patients in the three different sub-sets. Even though this would create biased data sets, it was carried out with the ambition to help the network learn features from both classes.

7

Threats to validity

In the upcoming sections, threats against the validity of the project are presented. They are divided into: Threats to internal validity, Threats to external validity, Threats to construct validity, and Threats to conclusion validity. Threats to internal validity corresponds to causes that have effected the outcome [40]. Threats to external validity explains different factors that can influence the generalizability of the project. Threats to construct validity presents changes made to the project with an intention to influence the results of it, for example the pre-processing of the image data. Threats to conclusion validity introduces factors that had a statistically significant effect on the outcome of the project.

7.1 Threats to internal validity

The biggest threat to the validity of this project's result is the lack of data. As mentioned, the data set was collected from VGR's research database Gothia 3A. It originally contained 824 patients, but due to some hospitals missing from VGR's image register and some images being of too poor quality, only 205 of the patients were available for the project. Although it is possible to augment the training data, the validation and test set will still be very small and the results from these sets might be biased.

Pre-processing of the non-image data can also be considered a threat since it involved modifying some of the input parameters. For instance, the smoking input was simplified into number of years of smoking. This meant that it will probably not reflect the complete picture and the result can be misleading. For example if a patient has been smoking two or three cigarettes every day for 10 years and another patient has been smoking 10 or more cigarettes per day during the same time period, these patients will obtain the same input value even though the latter patient has potentially caused more physical harm to himself.

The VGG19 network was used in this project but as mentioned in Section 2.2.1, large networks like this has lately been considered unnecessarily computational expensive. Instead, shallow networks are more and more investigated as the paper by Frankle and Carbin [25] discusses. This is something that could probably have been em-

ployed in this project and might yield a better performing network. It could also have been further developed by additional pre-training on other ultrasound images, not only showing abdominal aortas. This would imply that the network could learn even more ultrasound specific features before specializing on abdominal aortas.

The structure was reproduced from Bonnett [39] and Xu et al. [7] with minimal amount of changes. Though, larger adjustments, which perhaps might have improved the network's performance further, could have been applied if there would have been more time.

Another threat originating due to the lack of time was the tuning of the hyperparameters. There can be hyperparameters that were not tested but which would have yielded better results. Similarly, only one type of backend and deep learning framework was used. There are other types that could have been tested but instead the focus was directed towards data processing and finding hyperparameters that were good enough.

7.2 Threats to external validity

Since the data used in this project originates from Sweden, which is not included in the meta-studies referred to in Section 1.5.2, the generalizability of the results might not be valid for the comparison. In those countries the lifestyle might differ from Sweden, which could explain deviating results [11, 10]. The data was also only acquired from a small region and the patients were unevenly distributed across it as compliance for the screening program is better in higher socioeconomic areas [2]. This meant that there was a geographical limitation to the project and the results might not reflect the entire population of Sweden.

7.3 Threats to construct validity

The division of the training, validation and test set can be considered as a threat since it was not randomly generated. Instead it was performed to yield biased data sets and will not reflect the natural occurrence of the disease in Sweden.

The pre-processing of the images can also be a threat against the validity in many ways. Firstly, when shrinking the images to (224,224,3) it can give rise to artifacts. These are partly avoided when using `OpenCV`'s re-sizing algorithm with the method `INTER_AREA`. This method re-samples the image using pixel area relations which are supposed to minimize artifacts, especially line patterns that can emerge due to shrinking.

Secondly, artifacts can also be created when the dashed line in the images is removed. Attempts to avoid these have been made by trying to make the lines into

uniform color, in this case white, and then use the `OpenCV` in-painting technique `INPAINT_TELEA`. This technique replaces the dashed line's pixels by a normalized weighted sum of all known pixels in its neighbourhood. It also ensures that the pixels near the boundary of the lines are replaced first and receive the most contribution from the known pixels. This was supposed to prevent artifacts from forming, though some traces of the dashed line were still visible.

7.4 Threats to conclusion validity

Another factor that could affect the validity of the results is the lack of follow-up data. This meant that a comparison to other studies had to be made, in this case two meta-studies. Since the sub-group was very small (11 patients) the network's prediction might contradict the result from those studies as they contained a more extensive amount of patients.

8

Conclusion

This thesis investigated if a multimodal deep learning algorithm could assist in the diagnosis of sub-aneurysmal aortic dilatation. The results from the network did not agree entirely with the results obtained from meta-studies, though the algorithm still has potential to evolve and become a useful tool for diagnosing AAA.

Despite the conflicting results of the sub-aneurysmal group, the performance metrics from evaluating the network on the test set were promising. Also, the heatmaps rendered in the project clearly showed which image features the network found most important. They unwrapped the black box aspects of the image branch of the network and contributed to an easy explanation of its decisions. To analyze both the branches and find which input parameter that yielded the highest significance, permutation importance was employed. It showed that the image was the most important which strengthened the results from the heatmaps.

The main weaknesses with this study was the restricted access to data, especially the image data, and the lack of follow-up data for the patients with sub-aneurysms. These challenges limited the extent of the project and made it more difficult to carry out. It also made it harder to interpret the results as both uncertainty and bias were introduced. Two valuable lessons can be learned for the future when working with deep learning and health care. Firstly, be sure to have access to the data from the beginning and secondly, legal processes take time.

The prevalence of AAA is currently decreasing which means that the cost per quality-adjusted life year will increase and eventually it will not be profitable to keep screening to the same extent we do today. A decision support system could help decrease this cost by planning the visits of the follow-up examinations. The system presented in this work is a first step, which can be used for this purpose by predicting and diagnosing AAA.

8.1 Future work

Suggestions for future work involve everything from improving the existing algorithm to investigate other areas of use. For improving the existing algorithm further, more data is required. A larger data set introduces a higher variability, which will help

the network to generalize better.

Another improvement would be to access other types of patient data such as BMI, diabetes and heredity. This would yield a deeper knowledge of the disease profile and the training might be improved, resulting in more reliable predictions. A more certain evaluation of the results could also be achieved if follow-up data of the sub-aneurysms was available.

An interesting area to investigate is if algorithms like this can be used to create a more personalized follow-up plan for monitoring AAA. One solution could be to train the network to perform a multi-class classification instead, dividing the patients into healthy, sub-aneurysm, sick without a growing aneurysm, and sick with a growing aneurysm.

Another suggestion would be to predict the growth of the aneurysms. Since the growth is nonlinear and unpredictable, an accurate prediction may contribute to a more personalized follow-up plan for the patients. Both these solutions could help exclude patients whose aneurysm will never rupture and thereby prevent overdiagnosis. It might also allocate resources to more needed areas.

Bibliography

[1] Socialstyrelsen. *Screening för bukaortaaneurysm*. Socialstyrelsen, 2016. ISBN 978-91-7555-388-7.

[2] Socialstyrelsen and SBU. *Vetenskapligt underlag, Bilaga 1*. Socialstyrelsen, 2016. ISBN 978-91-7555-388-7.

[3] A. Wanhainen *et al.* Outcome of the swedish nationwide abdominal aortic aneurysm screening program. *Circulation*, 134(16):1141–1148, 2016. doi: 10.1161/CIRCULATIONAHA.116.022305. URL `https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.116.022305`.

[4] L. Lundkvist. Statistiskt meddelande: Sveriges framtida befolkning 2019–2070, 2019. URL `https://www.scb.se/contentassets/24496c5905454373b2910229c29001ec/be0401_2019i70_sm_be18sm1901.pdf`. visited on 2019-06-06.

[5] Nvidia. Deep learning Advances in Medicine, 2017. URL `https://www.nvidia.com/object/deep-learning-in-medicine.html`. visited on 2019-06-06.

[6] Q. K. Al-Shayea. Artificial neural networks in medical diagnosis. *IJCSI International Journal of Computer Science Issues*, 8, Mar 2011. ISSN 1694-0814. URL `https://www.researchgate.net/publication/285912467_Artificial_Neural_Networks_in_Medical_Diagnosis`. visited on 2019-06-06.

[7] T. Xu, H. Zhang, X. Huang, S. Zhang, and N.D. Metaxas. Multimodal deep learning for cervical dysplasia diagnosis. In S. Ourselin, L. Joskowicz, M.R. Sabuncu, G. Unal, and W. Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 115–123, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46723-8.

[8] D. Song, E. Kim, X. Huang, J. Patruno, H. Muñoz-Avila, J. Heflin, L. R. Long, and S. Antani. Multimodal entity coreference for cervical dysplasia diagnosis. *IEEE Transactions on Medical Imaging*, 34(1):229–245, Jan 2015. ISSN 0278-0062. doi: 10.1109/TMI.2014.2352311.

[9] T. Xu, X. Huang, E. Kim, L. R. Long, and S. Antani. Multi-test cervical cancer diagnosis with missing data estimation. *Proc.SPIE*, 9414, 2015. doi:

10.1117/12.2080871. URL https://doi.org/10.1117/12.2080871. visited on 2019-06-06.

[10] J.B. Wild *et al.* A Multicentre Observational Study of the Outcomes of Screening Detected Sub-aneurysmal Aortic Dilatation. *European Journal of Vascular and Endovascular Surgery*, 45(2):128–134, Feb 2013. ISSN 1078-5884. doi: 10.1016/j.ejvs.2012.11.024. URL https://doi.org/10.1016/j.ejvs.2012.11.024. visited on 2019-06-06.

[11] C. Hamel *et al.* Potential benefits and harms of offering ultrasound surveillance to men aged 65 years and older with a subaneurysmal (2.5-2.9 cm) infrarenal aorta. *Journal of Vascular Surgery*, 67(4):1298–1307, 2018. ISSN 0741-5214. doi: https://doi.org/10.1016/j.jvs.2017.11.074. URL http://www.sciencedirect.com/science/article/pii/S0741521418300193. visited on 2019-06-06.

[12] L. Wang *et al.* Associations of Diabetes and Obesity with Risk of Abdominal Aortic Aneurysm in Men. *Journal of obesity*, 2017:3521649, 2017. ISSN 2090-0716. doi: 10.1155/2017/3521649. URL https://www.ncbi.nlm.nih.gov/pubmed/28326193. visited on 2019-06-06.

[13] J. Raffort, F. Lareyre, M. Clément, R. Hassen-Khodja, G. Chinetti, and Z. Mallat. Diabetes and aortic aneurysm: current state of the art. *Cardiovascular research*, 114(13):1702–1713, Nov 2018. ISSN 1755-3245. doi: 10.1093/cvr/cvy174. URL https://www.ncbi.nlm.nih.gov/pmc/PMC6198737/. visted on 2019-06-06.

[14] J. Golledge, P. E. Norman, M. P. Murphy, and R.L. Dalman. Challenges and opportunities in limiting abdominal aortic aneurysm growth. *Journal of vascular surgery*, 65(1):225–233, Jan 2017. ISSN 1097-6809. doi: 10.1016/j.jvs.2016.08.003. URL http://www.ncbi.nlm.nih.gov/pubmed/27641464. visited on 2019-06-06.

[15] I. V. da Silva, D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves. Introduction. In *Artificial Neural Networks*, pages 3–19. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-43162-8_1. URL https://link.springer.com/chapter/10.1007%2F978-3-319-43162-8_1. visited on 2019-06-06.

[16] N. Shahid, T. Rappon, and W. Berta. Applications of artificial neural networks in health care organizational decision-making: A scoping review. *PloS one*, 14(2):e0212356–e0212356, Feb 2019. ISSN 1932-6203. doi: 10.1371/journal.pone.0212356. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0212356. visited on 2019-06-06.

[17] B. Mehlig. Artificial Neural Networks, Jan 2019. URL http://arxiv.org/abs/1901.05639. visited on 2019-06-06.

[18] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org. visited on 2019-06-06.

[19] M. A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2018. URL http://neuralnetworksanddeeplearning.com/. visited on 2019-06-06.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. *arXiv e-prints*, art. arXiv:1512.00567, Dec 2015. visited on 2019-06-06.

[21] K. He, X.Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. visited on 2019-06-06.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. visited on 2019-06-06.

[23] *Going Deeper with Convolutions*, 2015. URL https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf. visited on 2019-06-06.

[24] *Rethinking the Inception Architecture for Computer Vision*, 2015. ISBN 1512.00567v3. URL https://arxiv.org/pdf/1512.00567.pdf. visited on 2019-06-06.

[25] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7. visited on 2019-06-06.

[26] K. Liu, Y. Li, N. Xu, and P. Natarajan. Learn to Combine Modalities in Multimodal Deep Learning. *arXiv e-prints*, art. arXiv:1805.11730, May 2018. visited on 2019-06-06.

[27] F. Chollet. *Deep Learning with Python*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2017. ISBN 1617294438, 9781617294433.

[28] *Adam: A Method for Stochastic Optimization*, Dec 2014. URL https://arxiv.org/abs/1412.6980. visited on 2019-06-06.

[29] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4148–4158. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7003-the-marginal-value-of-adaptive-gradient-methods-in-machine-learning.pdf.

[30] F. Dyson. A meeting with Enrico Fermi. *Nature*, 427(6972):297, 2004. ISSN 1476-4687. doi: 10.1038/427297a. URL https://doi.org/10.1038/427297a.

[31] A. Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 2018. ISSN 2210-8327. doi: https://doi.org/10.1016/j.aci.2018.08.003. URL http://www.sciencedirect.com/science/article/pii/S2210832718301546.

[32] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learn-*

ing, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143874. URL http://doi.acm.org/10.1145/1143844.1143874.

[33] *On ROC Curve Analysis of Artificial Neural Network Classifiers*, 2017. URL https://aaai.org/ocs/index.php/FLAIRS/FLAIRS17/paper/view/15512. visited on 2019-06-06.

[34] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):1–21, Mar 2015. doi: 10.1371/journal.pone.0118432. URL https://doi.org/10.1371/journal.pone.0118432.

[35] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv e-prints*, art. arXiv:1312.6034, Dec 2013. visited on 2019-06-06.

[36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *CoRR*, 2016. URL https://arxiv.org/abs/1610.02391. visited on 2019-06-10.

[37] F. Chollet *et al.* Keras, 2015. URL https://keras.io. visited on 2019-06-06.

[38] K. Stol and B. Fitzgerald. The ABC of Software Engineering Research. *ACM Transactions on Software Engineering and Methodology*, 27:1–51, 2018. doi: 10.1145/3241743.

[39] C. Bonnett. Classifying e-commerce products based on images and text — Adventures in Machine Learning, 2016. URL http://cbonnett.github.io/Insight.html. visited on 2019-06-06.

[40] R. Feldt and A. Magazinius. Validity threats in empirical software engineering research - an initial survey., 2010. visited on 2019-06-06.

A

Appendix

In the upcoming sections additional training results from both pre-training and the training of the final network are shown. The difference in the settings of the hyperparameters, compared to the values yielding the best performance, is described in the caption.

A.1 Results from pre-training

Additional trends from the pre-training can be seen in Figure A.1.

A.2 Results from training of the final network

Additional trends from the training of the final network can be seen in Figure A.2, A.3, A.4.

A.3 Additional heatmaps

In Figure A.5 additional heatmaps to the ones presented in Section 4.2 are shown. They are produced using the same CAM-algorithm as describe in Section 3.4.

A.4 Additional saliency maps

In Figure A.6 additional saliency maps to the ones presented in Section 4.2 are shown. They are produced using the same class score as describe in Section 3.4.

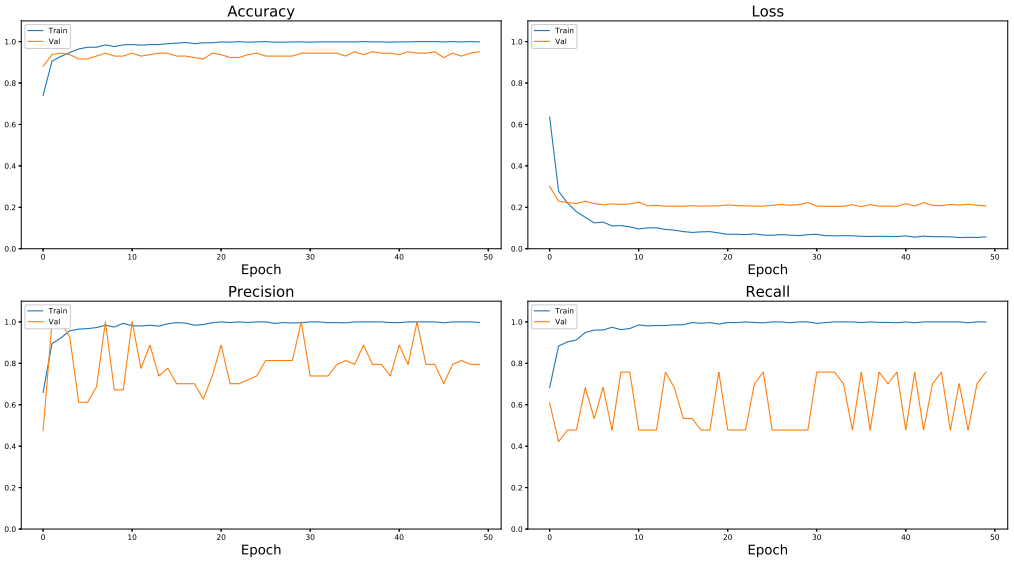


Figure A.1: The figure shows the performance metrics of the pre-trained network when using l2-regularization= 10^{-5}.

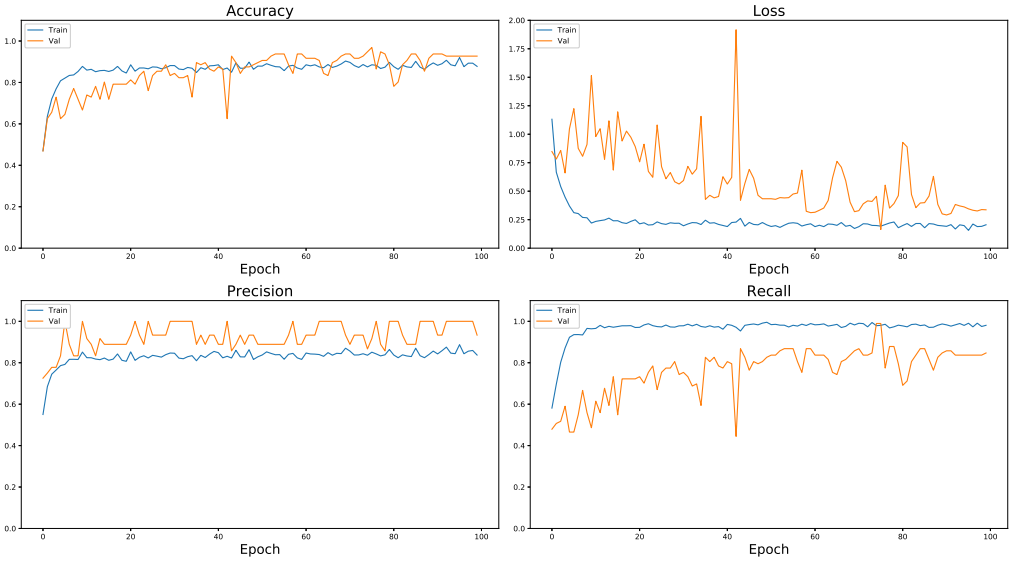


Figure A.2: The figure shows the performance metrics of the final network when using l2-regularization= 10^{-5}.

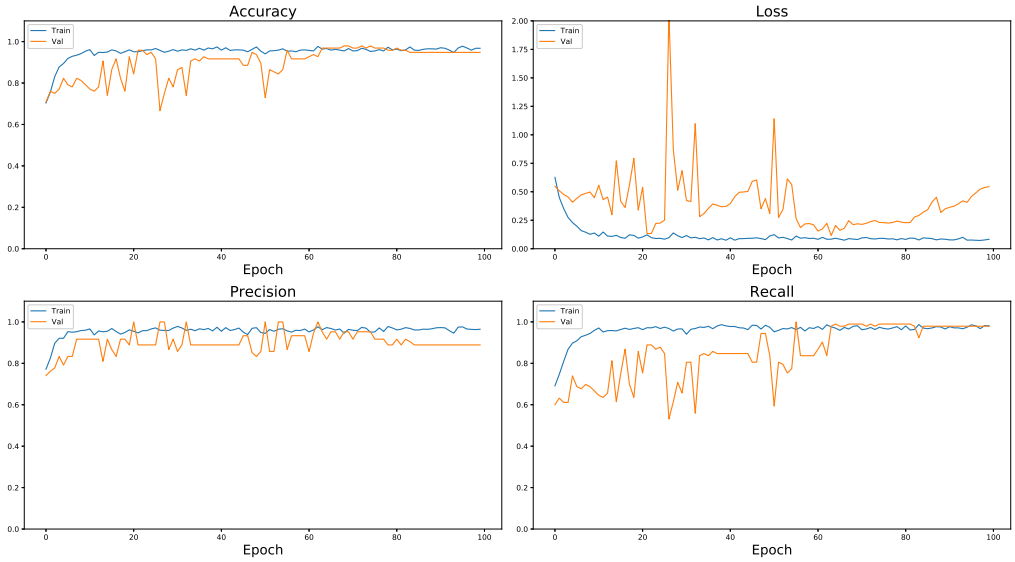


Figure A.3: The figure shows the performance metrics of the final network when using a decay rate= 10^{-5} and l2-regularization= 10^{-5}.

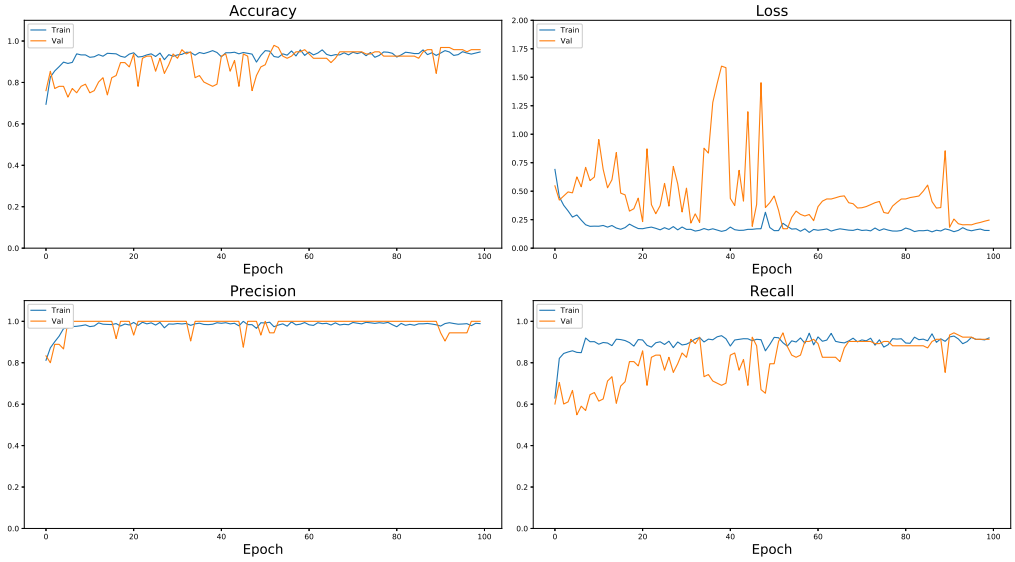


Figure A.4: The figure shows the performance metrics of the final network when using a decay rate= 10^{-5}.

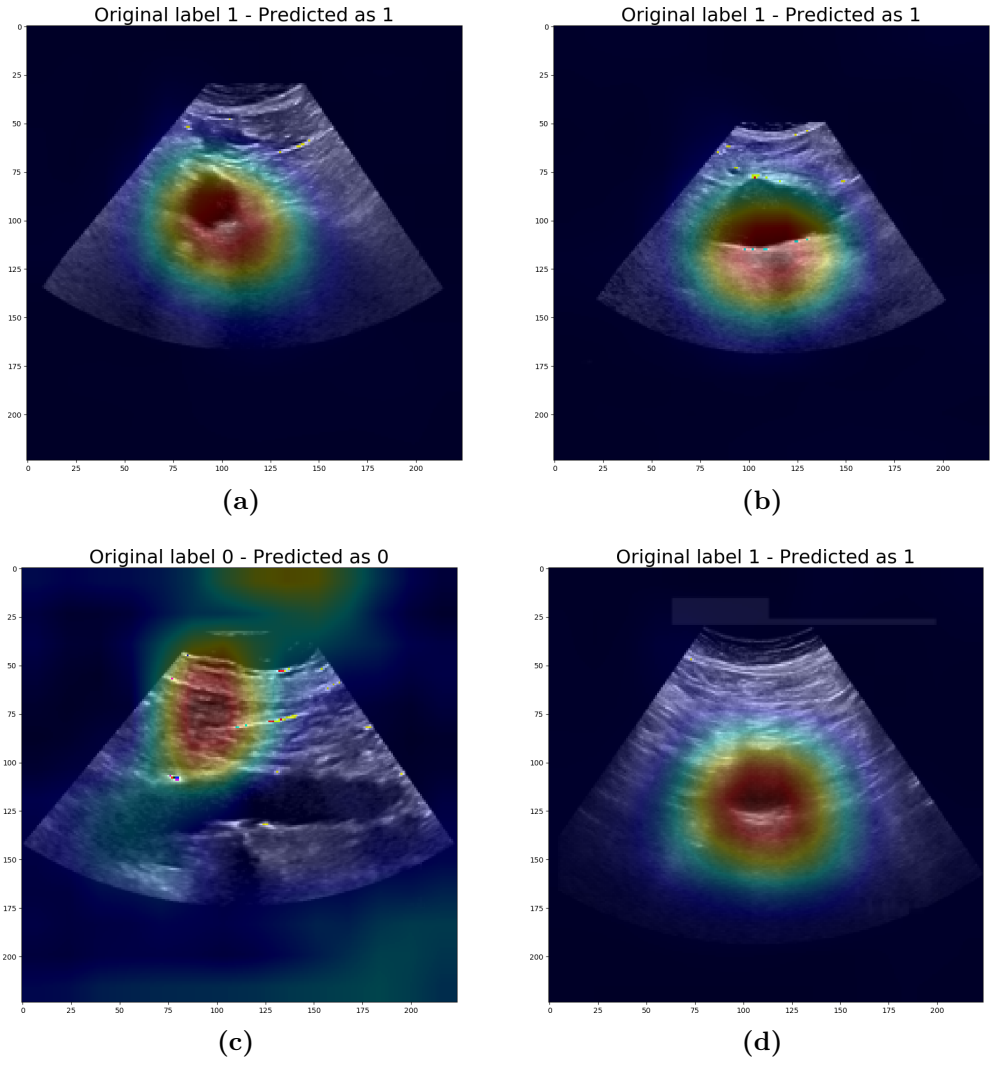
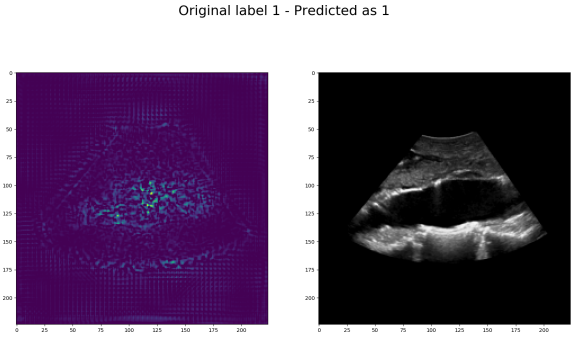
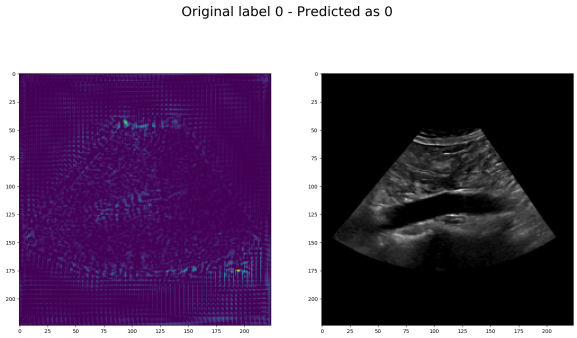


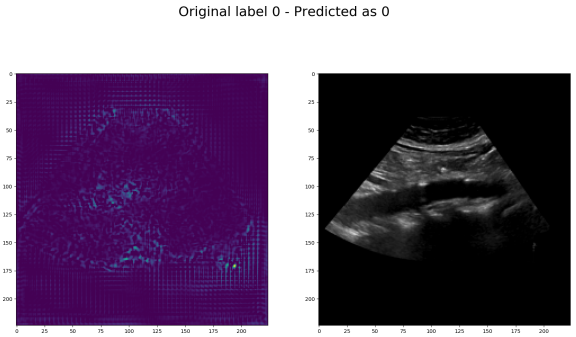
Figure A.5: Additional heatmaps to the ones presented in Section 4.2.



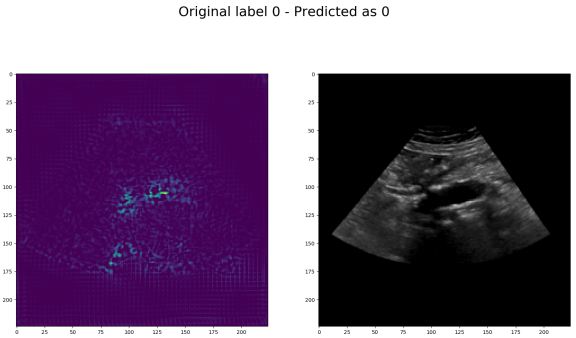
(a)



(b)



(c)



(d)

Figure A.6: Additional saliency maps to the ones presented in Section 4.2.