A First Step Towards an Algorithm for Breast Cancer Reoperation Prediction Using Machine Learning and Mammographic Images

Emma Jönsson

LUND UNIVERSITY

Department of Mathematics

Abstract

Cancer is the second leading cause of death worldwide and 30% of all cancer cases among women are breast cancer. A popular treatment is breast-conserving surgery, where only a part of the breast is surgically removed. Surgery is expensive and has a significant impact on the body, and on some women, a reoperation is needed. The aim of this thesis was to see if there is a possibility to predict whether a person will be in need of reoperation with the help of whole mammographic images and deep learning.

The data used in this thesis were collected from two different open sources: (1) The Chinese Mammography Database (CMMD) where 1052 benign images and 1090 malignant images were used. (2) The Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM) where 182 benign images and 145 malignant images were used. With those images, both a simple convolutional neural network (CNN) and a transfer learning network using the pre-trained model MobileNet were trained to classify the images as benign or malignant. All the networks were evaluated using learning curves, confusion matrix, accuracy, sensitivity, specificity, AUC and a ROC-curve.

The highest results obtained belonged to a transfer learning network that used the pre-trained model MobileNet and trained on the CMMD data set. It got an AUC value of 0.599.

Sammanfattning

Cancer är idag det näst vanligaste dödsorsaken i världen, där 30% av alla cancerfall bland kvinnor är bröstcancer. En vanlig behandling är bröstbevarande operation, där en bit av bröstet kirurgiskt tas bort. Operationer är både dyrt och har en betydande inverkan på kroppen och för vissa kvinnor krävs en omoperation efter den första operationen. Syftet med detta arbete har varit att undersöka möjligheten att förutsäga om en person kommer att vara i behov av en omoperation med hjälp av hela mammografibilder och maskininlärning.

Datan som användes i arbetet hämtades från två olika öppna källor: (1) The Chinese Mammography Database (CMMD) där 1052 benigna bilder och 1090 maligna bilder användes. (2) The Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM) får 182 benigna bilder och 145 maligna bilder användes. Med dessa bilder tränades både ett enkelt konvoluionellt nätverk och ett överförningsinlärningsnätverk med den för-tränade modellen MobileNet för att klassificera bilderna som benigna eller maligna. Alla nätverken utvärderades med inlärningskurvor, confusion matrix, noggrannhet, känslighet, specificitet och en ROC-kurva.

De högsta resultaten som erhölls var ett AUC-värde på 0.599 och tillhörde ett överföringsinlärning nätverk som använt den för-tränade modellen MobileNet och tränat på CMMD-datauppsättningen.

Acknowledgements

I cannot begin to express my thanks to my supervisor Ida Arvidsson, who has been guiding me throughout this thesis. Her inputs have been highly valuable. I would also like to express my gratitude to my assistant supervisor Jennie Karlsson for the advice she has been giving me and that she advised me to do this project. Finally, I would like to thank Kristina Lång for the effort she put into trying to obtain the data needed for this degree project.

Contents

1

Introduction

Today, cancer is the second most common cause of death both in Sweden and worldwide. The most common cancer diagnosis for women is breast cancer, where 30% of all cancer cases for women are breast cancer [1]. The most common method of treatment for breast cancer is to surgically remove parts of the breast or the entire breast, but undergoing such surgery is both expensive and has a major impact on the body [2]. Surgery is something you only want to do once, but of those who were diagnosed with breast cancer in Sweden between 2008 and 2020, 9% have had a reoperation [3].

Artificial intelligence (AI) is a computer system that tries to imitate the human brain, more specifically tries to mimic the ability of the brain to obtain information, see connections, conclude, solve problems and learn from experience. The areas of application are many, for example, can an AI drive a car, play chess, and translate languages. AI is growing in popularity and is showing improved results for different medical image analysis tasks. One way to learn an AI to classify images is by showing images and telling what is being shown to the AI. From this, the AI can learn and draw conclusions about what a non before seen image shows. [4]

The aim of this thesis is to see if there is a possibility to predict, with the help of deep learning and classification tasks, whether there is a high risk that reoperation will be needed or not. The hope is that this will make it easier to make decisions and be able to reduce the number of reoperations that take place. While waiting on the necessary data to be collected, two open mammography datasets have been trained to classify whether a tumor is benign or malignant. By doing so it was imagined that the same network could be used to classify reoperations instead by reusing most of the learnt features but train for this task instead. However, the data was unfortunately not collected in time to try out if it worked to change labels for the reoperation task.

This master's thesis begins with this introduction and is followed by the aim of the thesis and some medical background. After that, theory regarding AI is provided and then the data that was used is presented. This is followed by a short presentation of related work, the method, the result, and finally a discussion regarding the results, conclusion and further aspects.

2

Aim of Thesis

The aim of this thesis is to make the first step towards an algorithm for predictions of breast cancer reoperation. This is done with the following steps:

- Collecting data.
- Using benign and malignant images, since it is easier to distinguish than reoperated and non-reoperated, to train the networks of the following type:
  - A Simple CNN.
  - Transfer learning with a pre-trained model.
- Adding augmentation
- Evaluating the best models
- Comparing the results

3

Medical background

Cancer is a group of diseases and is characterized by an abnormal growth of cells caused by damage to the cell's genome. The damage leads to an atypical cell division and cell differentiation which results in an imbalance of replication and death. A tumor can be benign or malignant, but a cancerous tumor is always malignant. With this said, breast cancer is a malignant tumor in the mammary gland. [2, 5]

In most developed countries, breast cancer is the most common type of cancer among women. Among men, breast cancer is uncommon but there are cases where men got breast cancer as well. The tumor often starts as a slowly growing, non-painful lump in one of the breasts. The tumor can spread by creating metastases often in the lymph nodes in the armpit . To detect cancer early is important since it provides more treatment options and a higher chance of survival. [5]

3.1 Anatomy of the breast

The female breast consists of fat cells called adipose tissue, with age the breast often becomes fattier, hence having more adipose tissue. A healthy breast has about 12-20 lobes. Those lobes are built up of smaller lobules which is the gland that produces milk. Both the lobes and the lobules are connected by milk ducts which can be seen as small tubes that carry the milk to the nipple. An illustration of the breast anatomy can be seen in Figure 3.1. [6]

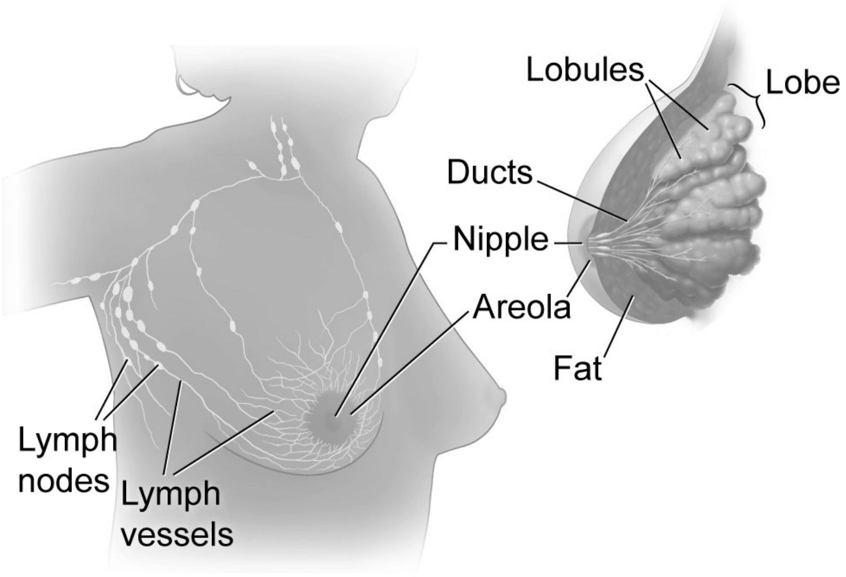


Figure 3.1 An illustration of the breast anatomy. Source: National cancer institute, Don Bliss (Illustrator)

3.2 Tumors

A tumor is a mass of abnormal tissue that is growing, it can be both benign and malignant, i.e. non-cancerous and cancerous. To determine if a tumor is benign or malignant a biopsy is performed. A benign tumor is often left alone if it is not causing any discomfort whereas on a malignant tumor the biopsy result can be used to determine the severity and aggressiveness of the tumor. A malignant tumor can also be metastatic cancer, which is when cancer migrates beyond its original placement to other parts of the body often using the lymph system or the bloodstream. [7]

3.3 Mammographic images

Mammographic imaging uses a low dosage of x-rays. The image has a black background and the breast tissue is in white and shades of gray. A denser breast appears whiter on the mammogram and a fattier, less dense breast appears more gray. [8]

Figure 3.2 shows a mammographic image of normal fatty breast tissue. As already mentioned, the gray areas in the image are fatty tissue, while the white areas consist of ducts and lobes that are denser. Figure 3.2 does not have any masses, but if it would have, the masses would also appear white on the mammogram, but the masses are generally more concentrated white due to being denser than the rest of the breast.[9]

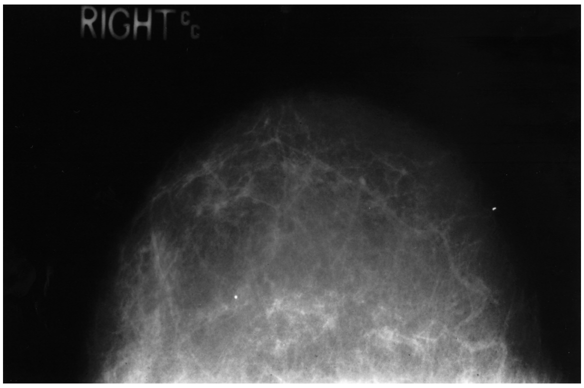


Figure 3.2 Mammographic image of normal fatty breast tissue. Source: Dr. Dwight Kaufman. National Cancer Institute.

Figure 3.3 shows two different mammograms of dense breasts without masses. Dense breasts often belong to young women who have not had children. A mammogram of a dense breast is typically harder to read due to less difference in density between normal and abnormal tissue. Worth mentioning is also that it has been shown that women with dense breasts tend to have a slightly increased risk of developing breast cancer.[9]

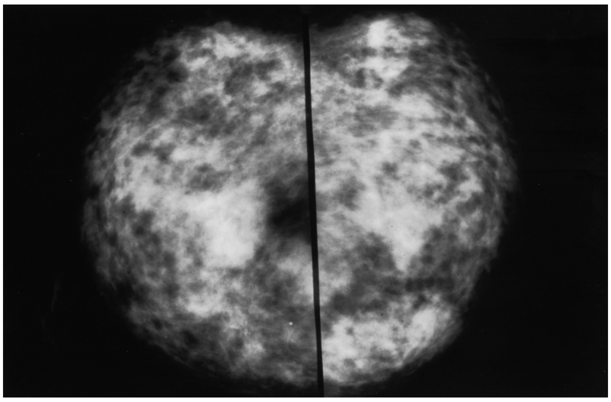


Figure 3.3 Mammographic image of normal dense breast tissue. Source: Dr. Dwight Kaufman. Division Of Cancer Treatment.

Figure 3.4 shows a mammogram with both fatty and dense tissue. The bottom left corner shows a concentrated white mass which is a malignant tumor. [9]

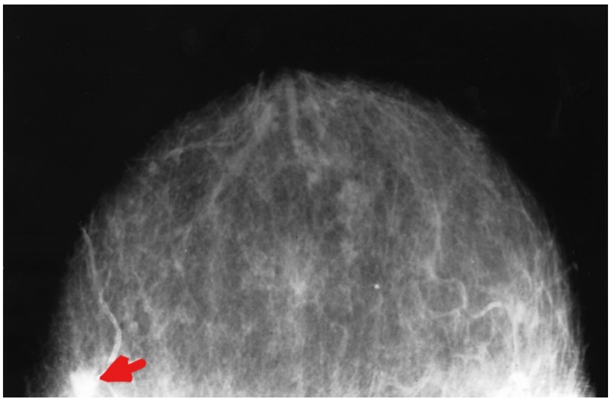


Figure 3.4 Mammographic image with cancerous mass in bottom left corner. Source: Dr. Dwight Kaufman. National Cancer Institute.

4

Theory - Artificial Neural Network

Artificial neural networks, also called ANNs, are inspired by the brain's structure and connections. The most basic network is a single layer network called the perceptron, but it is limited for a classification task. When using more than one layer it is called a multi-layer perceptron. When training on images, the most common ANN is a convolutional neural network. [10]

4.1 Convolutional Neural Network

The input of the convolutional neural network (CNN) is in vector form and compared to a regular ANN, the CNN uses weight-sharing to reduce the number of parameters. The CNN also focuses on feature detection in multiple layers. As the name indicates, CNN uses the mathematical operation convolution. When doing convolution on a 2-dimensional image, the input I (on vector form) with pixel coordinates (x,y) and the kernel K with dimensions (m,n) create a new vector H as the result of the convolution. Discrete convolution in two dimensions is calculated as follows,

$$H(y) = (I * K)(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=\infty}^{\infty} I(x-m, y-n)K(m,n). \tag{4.1}$$

The CNN is built up of three types of layers, which are the convolution layer, the pooling layer, and the fully connected layer. Stacking these layers results in the architecture of the CNN. An example of what this architecture can look like is shown in Figure 4.1. The different layers will be explained down below. [11, 10]

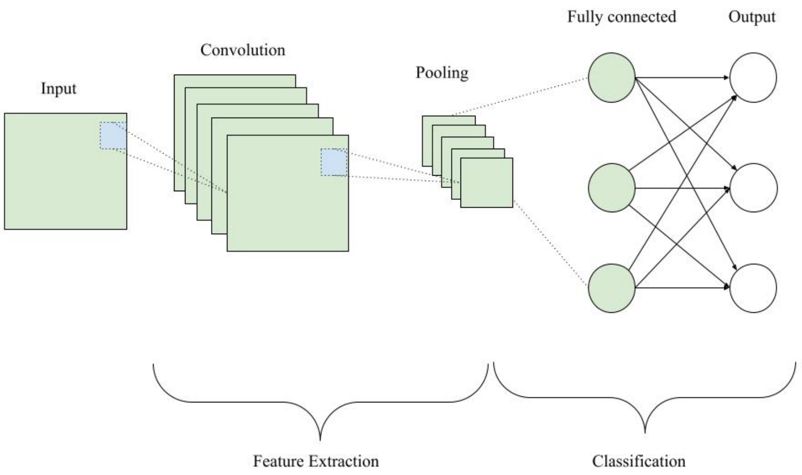


Figure 4.1 Illustration of the architecture of an CNN.

4.1.1 Convolution Layer

It is in the convolutional layer the trainable weights are. When the convolutional layer is added to the network three things need to be specified, kernel size, stride, and layer depth. The kernel can be seen as a filter or a small matrix and detects different features in the input, as defined in Equation 4.1. The size of the kernel is the height and the width of this matrix. The kernel is convolving across the input creating a feature map by calculating the dot product of the flipped filter and the input. The stride size decides how many steps the kernel will take and the layer depth sets the number of filters used in the convolution layer. [11, 10, 12]

Sometimes it is preferable to also compute non-linear features in the image, which is not possible when using only convolutions since it is linear. To solve it, a non-linearity layer is often placed directly after the convolutional layer to introduce the non-linearity to the activation map. Different operations can be used to make it non-linear such as sigmoid, tanh, and Rectified Linear Unit (ReLU), whereof ReLU often is used. The ReLU function [13] is defined as

$$\phi(x) = \max(0, x). \tag{4.2}$$

4.1.2 Pooling layer

The pooling layer performs downsampling of the feature map to reduce the dimension of the feature map and the number of parameters in the network. This is done by having the pooling layer operate over the output from the convolution layer and deriving a summary of the nearby outputs. There are several types of pooling functions such as weighted average, L2 norm, rectangular, and more. The most popular is max-pooling, which is represented in Figure 4.2. In max-pooling, the output represents the maximum output within the kernel size. The kernel size is usually of size 2x2 and has a stride size of 2. If this is the case it scales down the original map by 25% while keeping the depth. [10, 11]

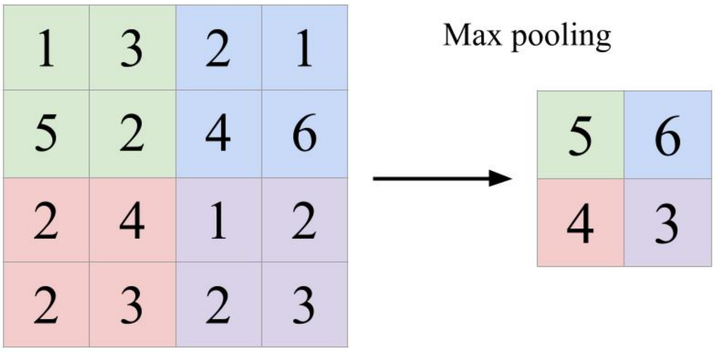


Figure 4.2 Illustration of max-pooling.

4.1.3 Fully connected layer

The fully connected layer contains neurons that are all directly connected to the two adjacent layers. In a categorical model, the last fully connected layer has the softmax activation function. [11] The definition of softmax is:

$$F(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}, \text{ for } i = 0, 1, 2, ..., k, \tag{4.3}$$

where x is the input vector of k real numbers i. An advantage of using softmax is that the sum of the output equals 1, which gives each class a value that can be interpreted as the probability for that class.

4.2 Important parameters for training

Before a CNN can be used it has to be trained. During training, some parameters are extra important to find the best weights for the classification task. The following subsections will mention some of those parameters.

4.2.1 Training, validation and test data

The training data is the data that is used to train the model. The model learns from this set of data. The validation data on the other hand is used for frequent evaluation of how the model fits the training data. The model is regularly evaluated on this data but does not learn from it. Lastly, the test data is used to provide an unbiased evaluation of how well the model fits the training data. This data is only used once when the model is finalized. [14]

4.2.2 Batch size, iteration, epoch and learning rate

The batch size refers to the number of training samples used in one iteration. The higher the batch size the more memory is needed. Iteration is the number of batches needed to complete one epoch. One epoch is defined as "when the entire dataset is passed forward and backward through the neural network only once". [15] Learning rate is a hyper-parameter, often between 0 and 1, that controls how much the weights of the network are adjusting with respect of the loss. [16]

4.2.3 Loss function

The loss function calculates the distance between the current output and the expected output. There are different kinds of loss functions, one common is cross-entropy. The cross-entropy loss function computes the difference between two probability distribution functions. There are different kinds of cross-entropy; binary, categorical, and sparse. They all give a value between 0 and 1 where 0 is perfect and 1 is bad. [17]

4.2.4 Optimization function

The optimization function is used to change parameters such as weights and learning rate to reduce the loss. The most common one is the Stochastic Gradient Descent (SGD) which is a variant of Gradient Descent. Another popular is Adaptive Moment Estimation (Adam) which is fast and converges rapidly, it also rectifies vanishing learning rates and high variance. A variant of Adam has been used in this thesis and is called Nadam which is similar to Adam but uses Nesterov's momentum. [18, 19]

4.3 Transfer learning

Since models can be hard to train from scratch to achieve good results, transfer learning can be used. Transfer learning is often used when the dataset is too small to train a model from start to finish, which often is the case with medical applications. A typical workflow can look like the following [20]:

- Use layers from the previously trained model.
- Freeze the layers to avoid destroying information for the future.
- Add new trainable layers on top of the pre-trained frozen layers.
- Train the new layers.
- Optional step is to fine-tune, which means unfreezing parts of or the whole model and retraining it with a low learning rate.

In this thesis, three different pre-trained networks will be used in the transfer learning network: MobileNet, VGG16, and Resnet50. All of the pre-trained networks are trained on the ImageNet data set, which is a data set that is commonly used in machine learning and consists of 14 million images of more than 21'000 classes [21].

4.3.1 MobileNet

MobileNet is based on depthwise separable convolution, except for the first layer which is a normal convolution. That a layer is depthwise separable means that it performs a single convolution on each channel instead of flattening it. Except for this, the size of the network is smaller than many other networks, which makes it good for mobile devices. Table 4.1 shows the architecture of the MobileNet. [22]

Table 4.1 The architecture of MobileNet. s stands for stride and and dw stands for depthwise convolution

MobileNet architecture		
Type	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
5x Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

4.3.2 VGG16

The VGG network got its name from the department of Visual Geometry Group at the University of Oxford. The VGG16 consists of 16 layers whereas the first 13 layers are convolution layers with the number of filters going from 64 to 512 and max pooling, this is followed by 3 fully connected layers whereof the last one is the output layer. [23, 24]

4.3.3 ResNet50

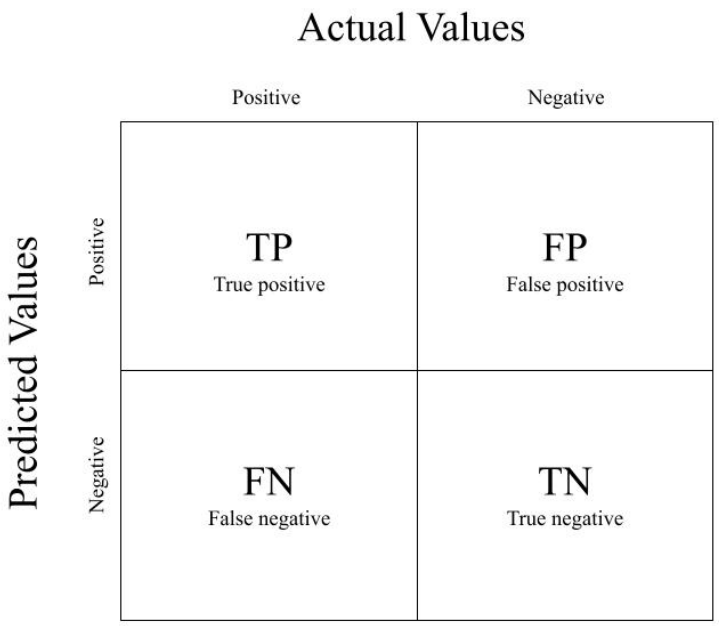
ResNet stands for Residual Network and the number 50 means that the network is 50 layers deep. The architecture of ResNet50 consists of a convolution layer followed by a max-pooling layer. This is followed by 4 bottleneck layers where each bottleneck layer contains 3 convolutional layers. One of the most important things with the ResNet is the skip connections which reduces the risk of vanishing gradients, which can be a problem for deep networks. [25, 26]

4.4 Evaluation

To evaluate the performance of a model different evaluation methods and metrics can be used. The following subsections will briefly explain a few of them.

4.4.1 Confusion matrix

The confusion matrix is a summary of the predicted results on a classification problem. It is useful for measuring recall, precision, specificity, sensitivity, accuracy, and ROC-curves. Figure 4.3 shows an illustration of a confusion matrix. TP stands for true positive, and is when the predicted value is positive and the actual value is positive. TN stands for true negative and is when the predicted value is negative and the true value is negative. FP stands for false positive and is when the predicted value is postitive but the actual value is negative, and lastly FN stands for false negative and is when the predicted value is negative and the actual value is positive. The values of TP, NP, TN, and FN can be used to calculate precision as seen in Equation 4.4, accuracy as seen in Equation 4.5, sensitivity as seen in Equation 4.6, and specificity as seen in Equation 4.7.



		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP True positive	FP False positive
	Negative	FN False negative	TN True negative

Figure 4.3 Illustration of confusion matrix

$$Precision = \frac{TP}{TP+FP} \tag{4.4}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.5}$$

“

$$Sensitivity = \frac{TP}{TP+FN} \tag{4.6}$$

$$Specificity = \frac{TN}{TN+FP} \tag{4.7}$$

4.4.2 Accuracy and Loss

Accuracy is the total number of correct predictions divided by the total number of predictions. Equation 4.5 shows how to calculate the accuracy of a binary classification. The result is between 0 and 1 where none or all were correctly predicted. Accuracy may be deceptive in the case of imbalanced data, therefore it is important to have a balanced dataset to begin with, or use a weighted accuracy instead. Loss is a value between 0 and 1 that calculates how close the predictions are to the actual output. The higher the loss the worse the model is. [11]

4.4.3 Learning Curves

Learning curves are widely used in machine learning. It is common to use both the training learning curve that is calculated from the training data and the validation learning curve that is calculated from the validation data and shows how the model is generalizing. It is also common to look at both an optimization curve i.e. loss and a performance curve i.e. accuracy. [27]

One of the most important parts of learning curves is to observe different dynamics such as under-fit, over-fit, and optimal fit. Figure 4.4 displays examples of this. The top part illustrates the classification problem with data points and the bottom part illustrates it in a deep learning view, which is the way seen when training.

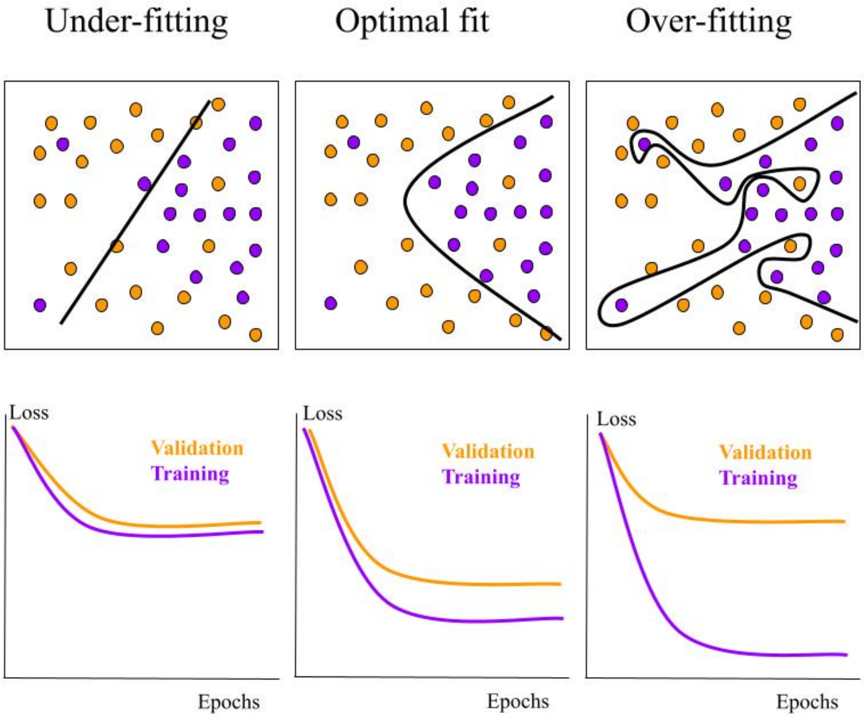


Figure 4.4 Illustration of a under-fitted, optimal and a over-fitted model. The top row shows data points illustrated as circles and the bottom row shows how it looks like when training a network.

4.4.3.1 Under-fitted model

A model that is under-fitting refers to a model that is unable to model either training data nor new data. Studying the lower plots in Figure 4.4, this can be identified as the training loss either is a horizontal line with high loss or that is decreasing and continues to do so at the end of the training, i.e. it could have been trained further. [27]

4.4.3.2 Over-fitted model

An over-fitted model refers to a model that knows its training data too well. This causes problems since the more specialized the model becomes, the less well it will generalize to new data and therefore the error will increase. This often happens when a model has more capacity than needed or high flexibility. It may also occur if the model is trained for too long. Studying the lower plots in Figure 4.4, this can be identified as the training loss continues to decrease or the validation loss decreases to a certain point and then starts to increase again. It can also be seen that there is a large difference between the training and validation loss. [27]

4.4.3.3 Optimal fit model

A good fit model is the goal and is achieved in the middle of over-fitting and under-fitting. When studying the lower plots in Figure 4.4, a well-fitted model can be seen as the training loss decreases to a point of stability and the validation loss does the same and with only a small gap to the training loss. [27]

4.4.4 ROC and AUC

The Receiver Operating Characteristic (ROC) curve is a probability curve and Area Under the Curve (AUC) represents the degree of separability. They are both important evaluation metrics for checking the model's performance. An illustration of the ROC-curve can be seen in Figure 4.5. The higher the AUC the better the model is on predictions. The curve is plotted with True Positive Rate (TPR) on the y-axis and the False Positive Rate (FPR) on the x-axis. TPR is defined as the sensitivity, which is defined in equation 4.6, and the FPR is defined as 1 - Specificity. [28]

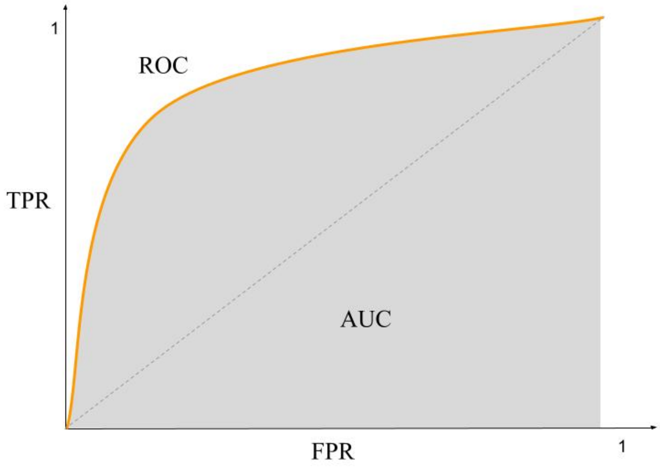


Figure 4.5 Illustration of the ROC-curve.

5

Datasets

The data used in this thesis was conducted from two different open sources. A third data set containing data from either Malmö Hospital or Optimam was supposed to be used, but due to a problem with receiving this data, the third data set was omitted. Optimam [29] is a large resource with mammography from the United Kingdom breast screening centers with clinical details. The images were collected over a 10-year period and has data from over 17'000 women. In Table 5.1 a summary of the data sets used is shown.

5.1 The Chinese Mammography Database

The Chinese Mammography Database [30] (CMMD) was downloaded from the cancer imaging archive and contains two sets of data, where the first set contains information on left and right breast, age, type of abnormality, and classification (begin vs malignant). The second set contained a subtype in addition to the above-mentioned. Some patients had both benign and malignant tumors, those were omitted from the data. Only the first set was used and it ended up containing 1052 images of benign tumors and 1090 of malignant tumors. All the images have the size 1914x2294.

5.2 Curated Breast Imaging Subset of Digital Database for Screening Mammography

The Curated Breast Imaging Subset of Digital Database for Screening Mammography [31] (CBIS-DDSM) was downloaded from the cancer imaging archive and is an updated version of the Digital Database for Screening Mammography (DDSM). The database contains 1566 participants and images of both normal, malignant, and benign images, as well as zoomed-in and whole breast images. Only the whole breast images were used. In the end, 182 benign images and 145 malignant images were used. The images vary in size.

Table 5.1 A summary of the images distribution for each data set

Data set	Benign	Malignant	Total
CMMD	1052	1090	2142
CBIS-DDSM	182	145	327

6

Related Works

No previous work has been found regarding neural networks and predictions of reoperations. On the other hand, a lot has been done regarding the classification of breast cancer as benign and malignant.

With DDSM and CBIS-DDSM as datasets and by using a 5-fold cross-validation and patches, John D. Miller, Vignesh A. Arasu, Albert X. Pu, Laurie R. Margolies, Weiva Sieh, and Li Shen manage to get a test set AUC of 0.687 with a patch size of 96x96 and a test set accuracy of 0.763 when using a model called BYOL [32],which is a self-supervised Learning method based on a strong augmentation. They used the including ROIs of the CBIS-DDSM and pre-trained the networks on tiled patches drawn from the whole mammograms and evaluated them on the whole image. More specifics regarding their work can be found in their report "Self-Supervised Deep Learning to Enhance Breast Cancer Detection on Screening Mammography" [33].

Kevin Wu, Eric Wu, Yaping Wu, Hongna Tan, Greg Sorensen, Meiyun Wang, and Bill Lotter competed in the DREAM challenge which is a competition where training nor testing data are publicly available. Wu et al. manage to get an AUC result of 0.93 +- 0.01 on their model by first training on patches from Optimam and then using end-to-end training on a full-scale image from the DDSM data set. The backbone of the network was MobileNet and the training consisted of (1) patch-level training on DDSM Optimam data set. (2) image-level training on DDSM Optimam and (3) image-level training on the DREAM data set. If interested in their work more can be found in their paper "Validation of a deep learning mammography model in a population with low screening rates" [34]

Benjamin Stadnick, Jan Witowski, Vishwaesh Rajiv, Jakub Chłędowski, Farah E. Shamout, Kyunghyun Cho, and Krzysztof J. Geras compared five different models on seven different international data sets, where CMMD was one of the data sets, in their paper "Meta-repository of screening mammography classifiers"[35]. The models used differ in architecture, training procedures, and preforms differently. For the CMMD data they manage to get an AUC of 0.534 with the End2end (with DDSM as training data) model, 0.449 with the End2end (with Inbreast as training data) model, 0.806 with the Faster R-CNN, 0.740 with the DMV-CNN model, 0.825 with the GMIC (single) model, 0.831 with the GMIC (top-5 ensemble) model and 0.785 with the GLAM model.

7

Method

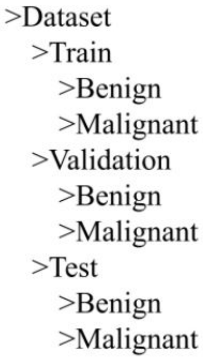
The method used in this thesis can be split up into 2 different parts regarding the type of network. In the first part, a Simple CNN was trained with the CMMD data set and in the second part, transfer learning was used. The second part can be split into three different sub parts regarding the different datasets used and their combination. In the first subpart, only the CMMD data was used, in the second subpart, the CMMD data was combined with the data from CBIS-DDSM so that the CBIS-DDSM ended up as the test data. In the third subpart, the CMMD and CBIS-DDSM data were combined and evenly distributed over both training, validation, and test. The method was suppose to have an additional subpart containing data collected from either Malmö hospital or Optimam, but as stated in the introduction, this data was unfortunately not received in time which led to the this step being omitted. In addition to training the network for the best performance only, an overtrained network was also trained, with the purpose to check that the network actually could learn from the information. The integrated development environment (IDE) used to train the networks was Google Colab.

7.1 Classes

Both the data from CMMD and CBIS-DDSM were separated into two classes; benign and malignant, where class 0 was benign and class 1 was malignant. CBIS-DDSM had a third class called benign with callback, which was simply excluded.

7.2 Data split

The data were in general split into 70% training, 15% validation, and 15% test according to Figure 7.1. The only time it was not split according to this was when CBIS-DDSM was used as test data, then all the data from CBIS-DDSM was used as test data, 80% of the CMMD was used as training data and the rest 20% was used as validation data. When combining CMMD and CBIS-DDSM, the datasets were first split according to 70%, 15,% 15% separately, and then the different parts were combined to evenly distribute them. This was important due to CBIS-DDSM having fewer images.



```
>Dataset
  >Train
    >Benign
    >Malignant
  >Validation
    >Benign
    >Malignant
  >Test
    >Benign
    >Malignant
```

Figure 7.1 A sketch of the folder structure that was used.

7.3 Image resizing

The smallest original image found were 1914x2294 pixels. Using this size made the network crash due to lack of RAM-memory. Resizing it to 20% of its original size (383x586) was still too large. Due to this and that MobileNet needs an input of size 224x244 the images were resized to 224x224.

7.4 Augmentation

Image data augmentation was used to create more training data from existing data. This was done by using Keras ImageDataGenerator [36]. Since some images had the breast on the left side and some on the right side, a horizontal flip was used. In addition to this, zoom, rotation, and shear were used. A summary of the augmentation can be seen in Table 7.1

Table 7.1 A summary of the augmentation used.

Zoom_range	0.2
Rotation_range	90
Shear_range	0.2
Horizontal_flip	True

7.5 Parameters

Table 7.2 shows the different parameters used when training the different networks.

Table 7.2 Training parameters for the different networks.

Network	Simple CNN	Transfer learning, only CMMD	Transfer Learning, CMMD and CBIS-DDSM mixed	Transfer Learning, CBIS-DDSM as test data
Optimizer	Adam	Nadam	Nadam	Nadam
Epochs	25	50	50	50
Learning rate	0.001	0.00001	0.00001	0.00001
Batch size	16	16	16	16

7.6 Simple CNN

Figure 7.2 shows the architecture of the Simple CNN used when training only the CMMD data. It was constructed of four convolution layers with the number of channels being 16, 32, 64, and 128 with an kernel size of 3x3. Between the layers drop out of 20% and batch normalization were used as well as an max pooling layer with a kernel size of 2x2. This was followed by a flattened layer and two dense layers, where the first one had size 8 with ReLU as activation function and the last one had size 8 with softmax as activation function.

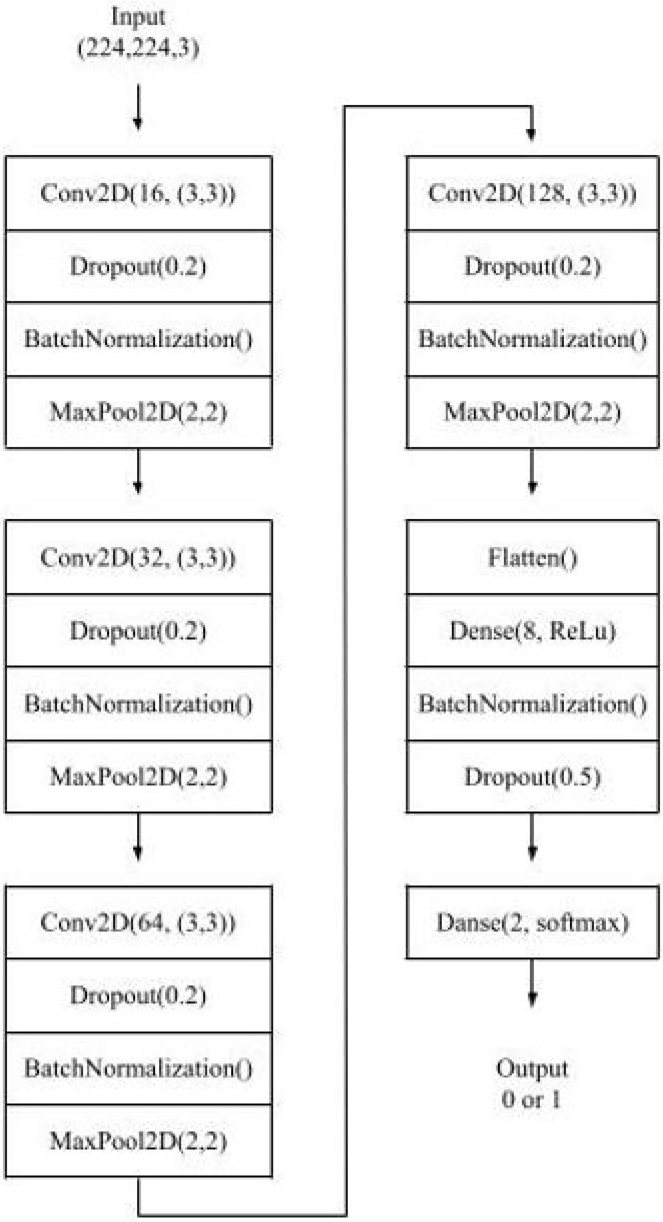


Figure 7.2 The architecture of the Simple CNN.

7.7 Transfer Learning

The transfer learning network was trained by using the three pre-trained models, MobileNet, VGG16, and ResNet50, one at a time. After training the three networks they were compared to each other and the best one was used for evaluation and further training.

A base model was created using MobileNet, which was the pre-trained model that gave the best result, without including the top. The model is illustrated in Figure 7.3, and was created with a sequential model followed by the base model. An average pooling with a size of 7 was used after the base model to get a smaller network. After that, flatten the model and add a dense layer with size 1024 and ReLU as activation function and then three more dense layers with size 500 and ReLU as activation function. In between all dense layers, a dropout of 50% was added. Lastly, a dense layer of size 2 with softmax as an activation function was added.

7.8 Overtraining

When doing the overtrained network, the same schematics as Figure 7.3 was used, but all augmentation was removed and no dropout was added.

7.9 Evaluation

When each network was trained, an evaluation was performed. Firstly the learning curves were plotted to get an idea of how the network performed. This was done by generating a plot with the training and validation loss and accuracy against the number of epochs. The next step was letting the model predict the data sets. To do this, the true and predicted labels were obtained from the test data and generated a confusion matrix with the help of the scikit-learn library [37]. A confusion matrix from the validation data were also generated for the transfer learning networks. The results from the confusion matrix were used to calculate the accuracy with Equation 4.5, as well as the sensitivity with Equation 4.6 and specificity with Equation 4.7. Finally, the ROC curve were plotted, also with the help of the scikit-learn library. In addition to this Keras classification report was used to get an overview of the performance.

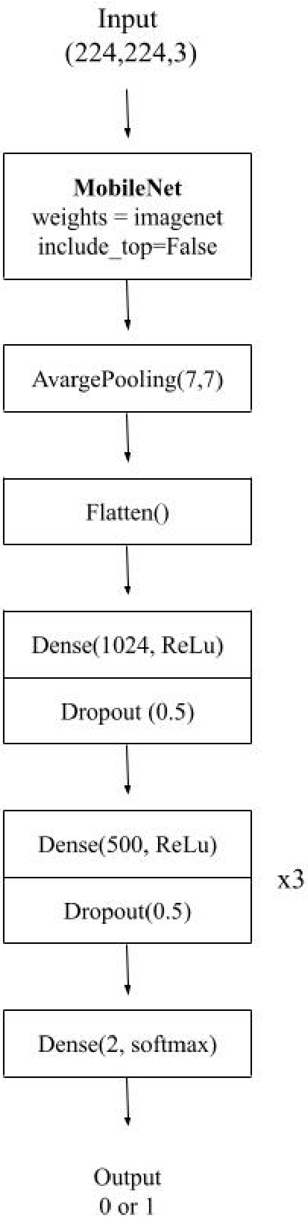


Figure 7.3 Schematic of the layers in the transfer model with MobileNet.

8

Result

8.1 Simple CNN

Figure 8.1 shows the accuracy and loss plotted against the number of epochs for the CMMD data trained on the Simple CNN. The classification report can be seen in Table 8.2 and Figure 8.2 shows the ROC curve. Table 8.1 shows the performance of the network, calculated from the confusion matrix of the test data shown in Figure 8.3.

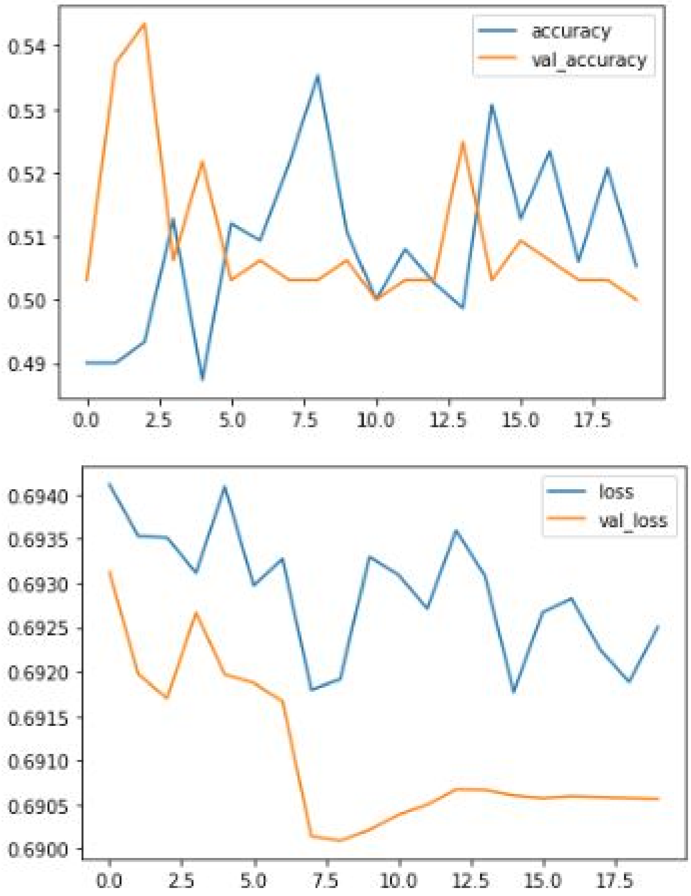


Figure 8.1 Training and validation accuracy and loss for the Simple CNN trained with CMMD data.

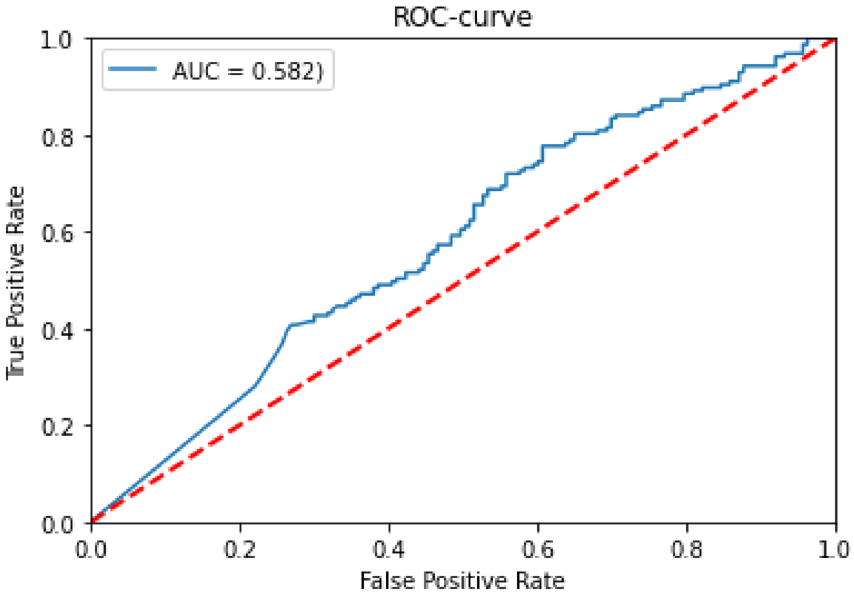


Figure 8.2 ROC curve for the Simple CNN trained with CMMD data.

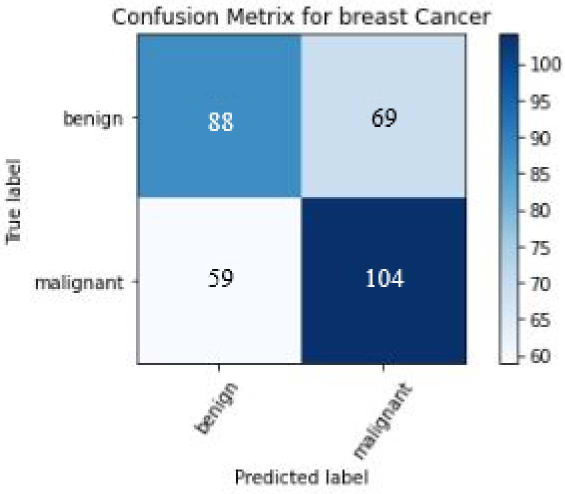


Figure 8.3 Confusion matrix made with test data from the Simple CNN trained with CMMD data.

Table 8.1 Performance of the the Simple CNN trained with CMMD data.

Accuracy	Sensitivity	Specificity
0.60	0.599	0.601

Table 8.2 Classification report for the Simple CNN trained with CMMD data. Label 0 refers to the benign class and label 2 the malignant class. Support is the number of images used for the calculations.

label	Precision	Sensitivity	f1-score	support
0	0.60	0.56	0.58	157
1	0.60	0.64	0.62	163

accuracy				0.60	320
macro avg		0.60	0.60	0.60	320
weighted avg		0.60	0.60	0.60	320

8.2 Transfer Learning with MobileNet

8.2.1 CMMD as dataset

Figure 8.4 shows the accuracy and loss plotted against the number of epochs for the transfer learning network, using MobileNet, trained on the CMMD data. Table 8.4 shows the classification report from the network and Table 8.3 states the performance of the model calculated from the confusion matrices shown in Figure 8.6a and 8.6b.

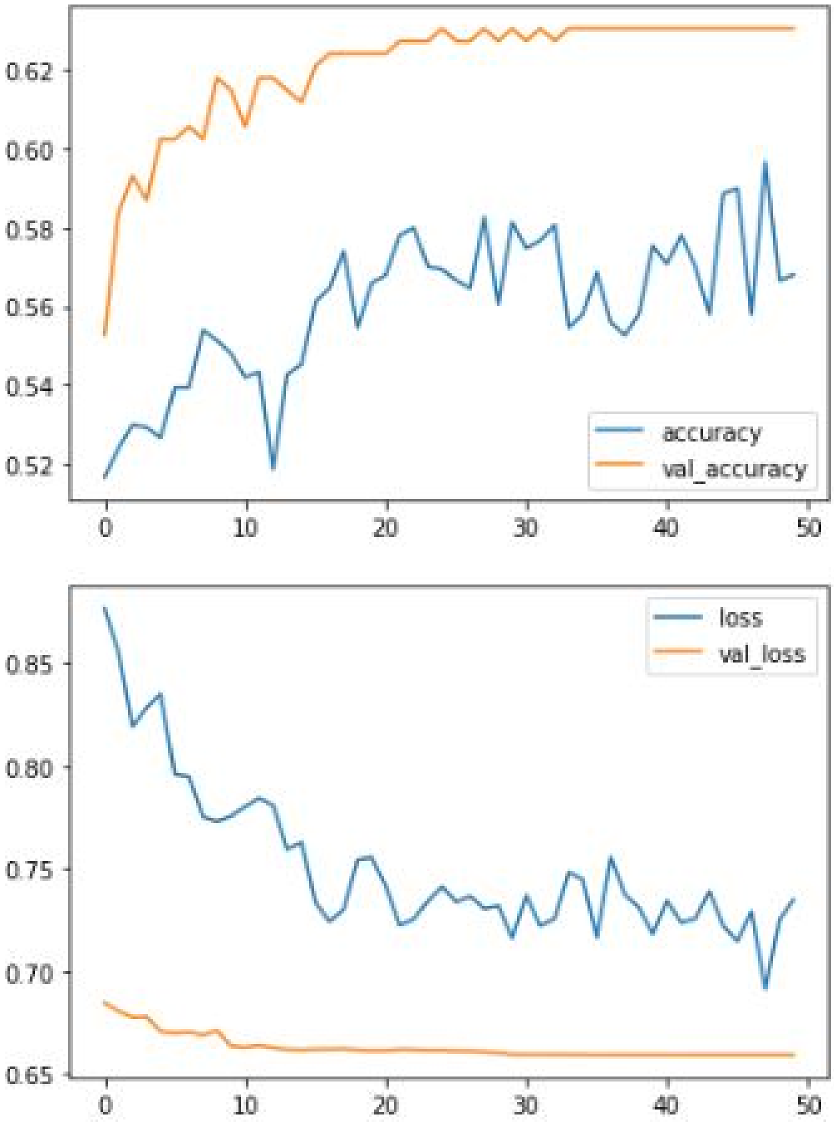


Figure 8.4 Training and validation accuracy and loss for the transfer learning network, using MobileNet, trained on the CMMD data

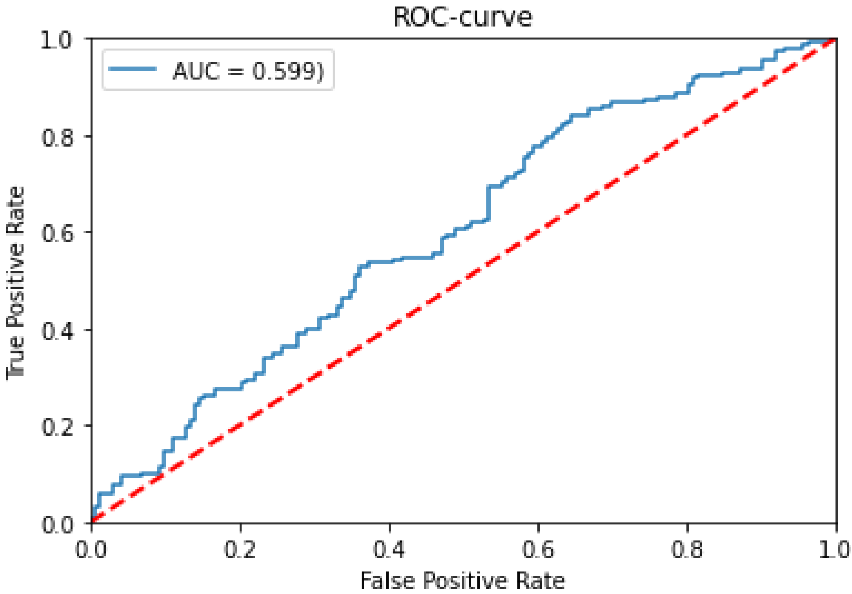
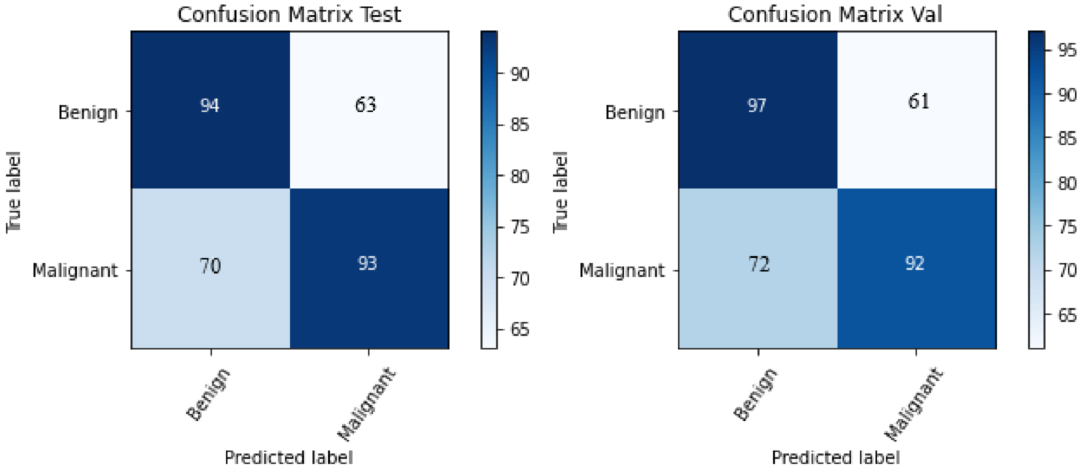


Figure 8.5 ROC curve for the transfer learning network, using MobileNet, trained on the CMMD data



(a) Confusion matrix made from test data

(b) Confusion matrix made from validation data

Figure 8.6 Confusion matrix from the transfer learning network, using MobileNet, trained on the CMMD data. Shows both a confusion matrix made with validation data and test data.

Table 8.3 Performance of the transfer learning network, using MobileNet, trained on the CMMD data, calculated from the confusion matrix made from test data as well as validation data.

Measurement	Accuracy	Sensitivity	Specificity
Test data	0.584	0.573	0.596
Validation data	0.587	0.574	0.601

Table 8.4 Classification report for the transfer learning network, using MobileNet, trained on the CMMD data. Label 0 refers to the benign class and label 2 the malignant class. Support is the number of images used for the calculations.

	label	Precision	Sensitivity	f1-score	support
	0	0.63	0.46	0.53	157
	1	0.59	0.74	0.66	163
accuracy				0.60	320
macro avg		0.61	0.60	0.59	320
weighted avg		0.61	0.60	0.59	320

8.2.2 CMMD with CBIS-DDSM as test data

Figure 8.7 shows the accuracy ans loss plotted against the number of epochs for the transfer learning network, using MobileNet, trained with CMMD and containing test data from CBIS-DDSM. The classification report can be read in Table 8.6. Two confusion matrices were made, one from test data and one from validation data, those can be seen in Figure 8.9a and 8.9b. From the confusion matrices performance has been calculated, the results can be seen in Table 8.5. The ROC curve can be seen in Figure 8.8.

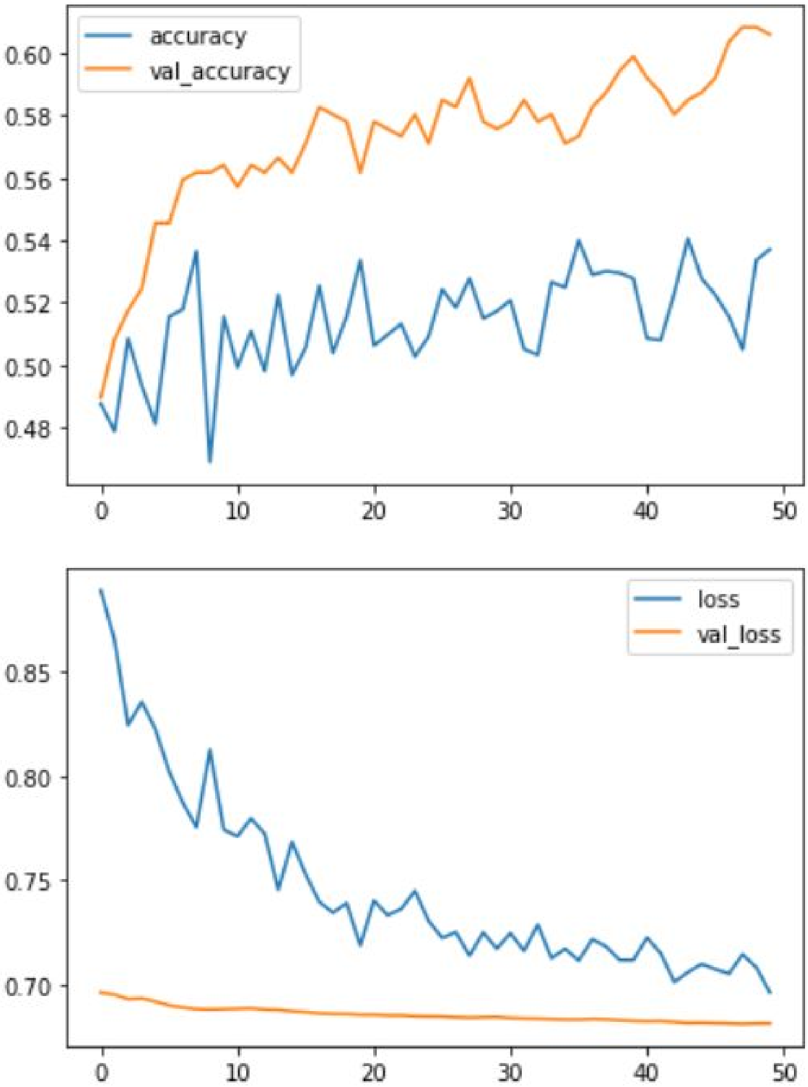


Figure 8.7 Training and validation accuracy and loss for the transfer learning network, using MobileNet, trained on the CMMD data and using CBIS-DDSM as test data.

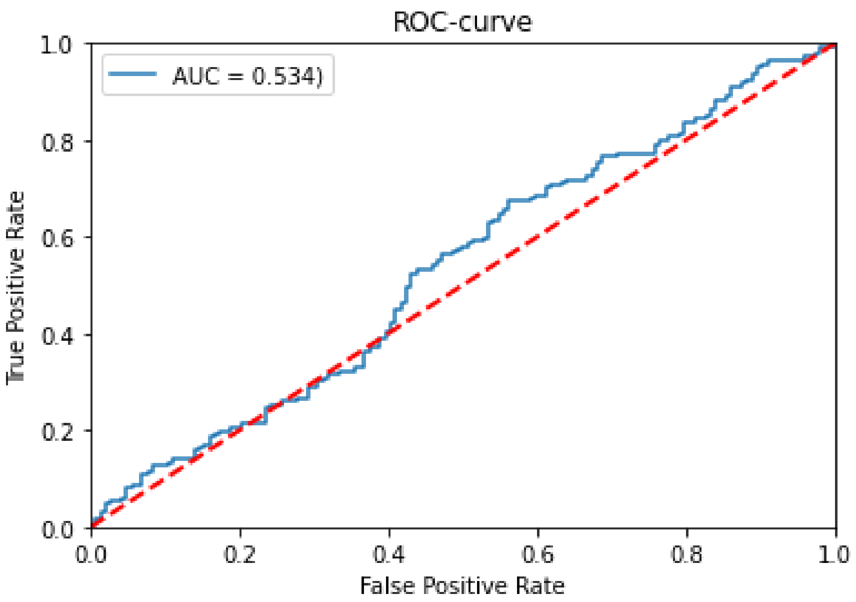
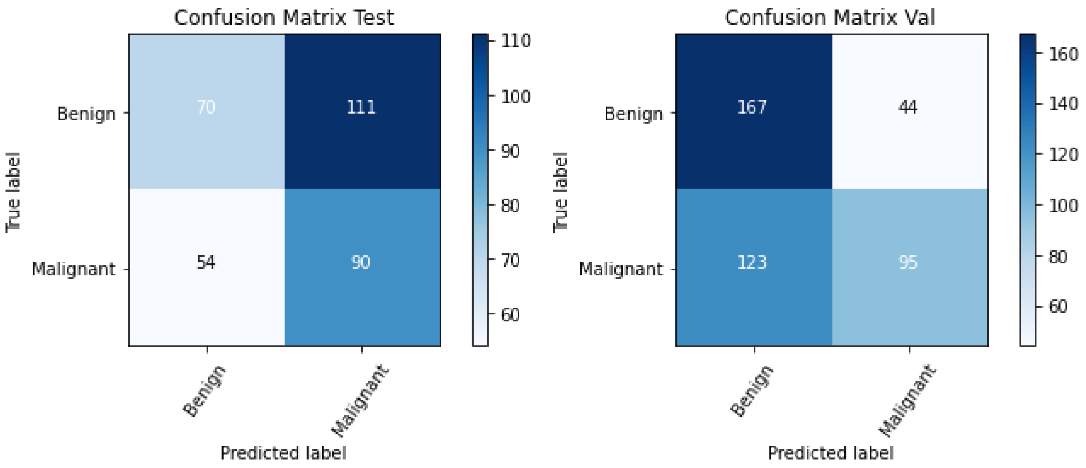


Figure 8.8 ROC curve for the transfer learning network, using MobileNet, trained on the CMMD data and using CBIS-DDSM as test data.



(a) Confusion matrix made from test data

(b) Confusion matrix made from validation data

Figure 8.9 Confusion matrix from the transfer learning network, using MobileNet, trained on the CMMD data and using CBIS-DDSM as test data. Shows both a confusion matrix made with validation data and test data.

Table 8.5 Performance of the transfer learning network, using MobileNet, trained on the CMMD data and using CBIS-DDSM as test data. Calculated from the confusion matrix made from test data as well as validation data.

Measurement	Accuracy	Sensitivity	Specificity
Test data	0.492	0.565	0.552
Validation data	0.611	0.675	0.683

Table 8.6 Classification report for the transfer learning network, using MobileNet, trained on the CMMD data and using CBIS-DDSM as test data. Label 0 refers to the benign class and label 2 the malignant class. Support is the number of images used for the calculations.

	label	Precision	Sensitivity	f1-score	support
	0	0.64	0.38	0.48	181
	1	0.49	0.74	0.59	144
accuracy				0.57	325
macro avg		0.57	0.56	0.53	325
weighted avg		0.57	0.54	0.53	325

8.2.3 CMMD mixed with CBIS-DDSM

Figure 8.10 displays the accuracy and loss plotted against the number of epochs for the network that was trained with a mix of CMMD and CBIS-DDSM. The network used transfer learning with the pre-trained model MobileNet shown in Figure 7.3. The ROC-curve can be seen in Figure 8.11 and the confusion matrices used to calculate the performance shown in Table 8.7 can be seen in Figure 8.12a and 8.12b. The classification report can be seen in Table 8.8.

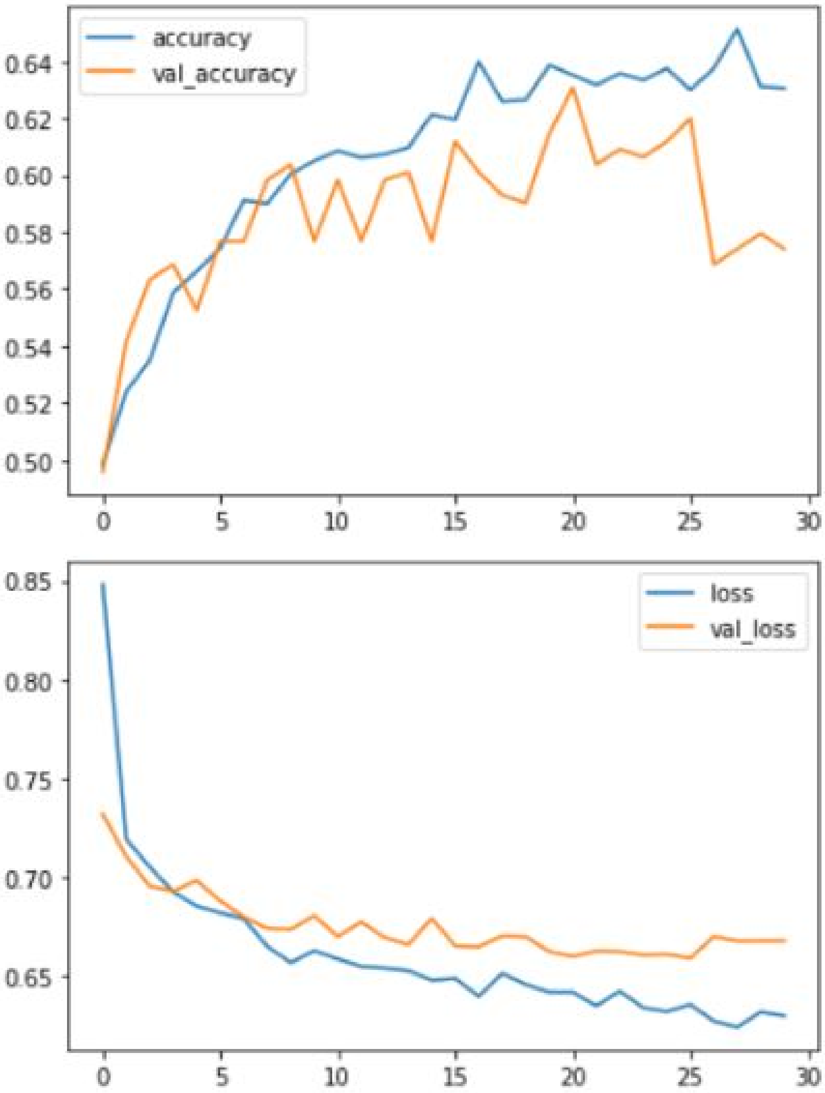


Figure 8.10 Training and validation accuracy and loss for the transfer learning network, using MobileNet, trained on the CMMD and CBIS-DDSM mixed.

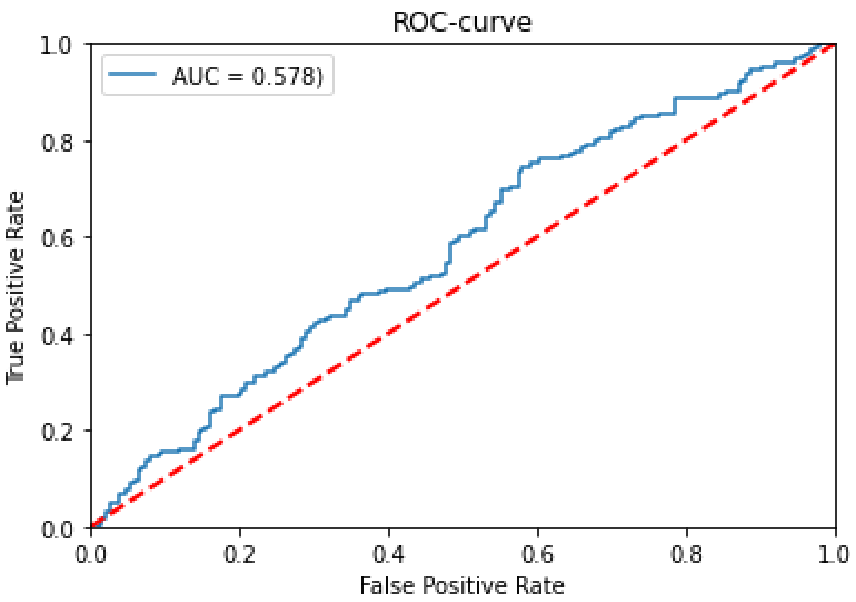
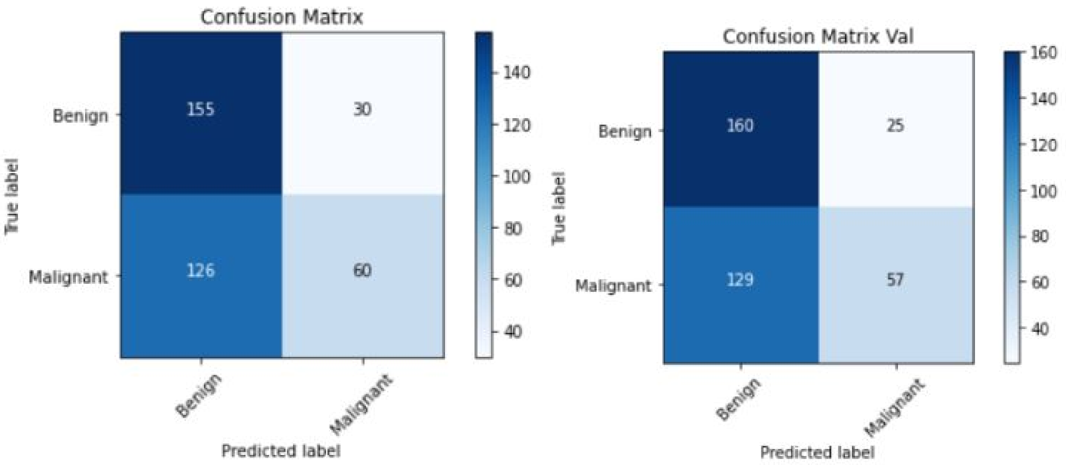


Figure 8.11 ROC curve for the transfer learning network, using MobileNet, trained on the CMMD and CBIS-DDSM mixed.



(a) Confusion matrix made from test data

(b) Confusion matrix made from validation data

Figure 8.12 Confusion matrix from the transfer learning network, using MobileNet, trained on the CMMD and CBIS-DDSM mixed. Shows both a confusion matrix made with validation data and test data.

Table 8.7 Performance of the transfer learning network, using MobileNet, trained on the CMMD and CBIS-DDSM mixed. Calculated from the confusion matrix made from test data as well as validation data.

Measurement	accuracy	sensitivity	specificity
Test data	0.580	0.552	0.667
Validation data	0.584	0.553	0.695

Table 8.8 Classification report for the transfer learning network, using MobileNet, trained on the CMMD and CBIS-DDSM mixed. Label 0 refers to the benign class and label 2 the malignant class. Support is the number of images used for the calculations.

	label	Precision	Sensitivity	f1-score	support
	0	0.59	0.75	0.66	185
	1	0.66	0.49	0.56	186
accuracy				0.62	371
macro avg		0.63	0.62	0.61	371
weighted avg		0.63	0.62	0.61	371

8.2.4 Overtraining

The results for the network that was forced to overtrain are shown in Figure 8.13, where the top plot shows the accuracy and the bottom part shows the loss. As mentioned in section 4.4.3 the network is overtraining which can be seen since the gap between the training and validation loss and accuracy is large. It can also be seen since the performance on the validation data isn't better than random while the training data received almost 100% accuracy.

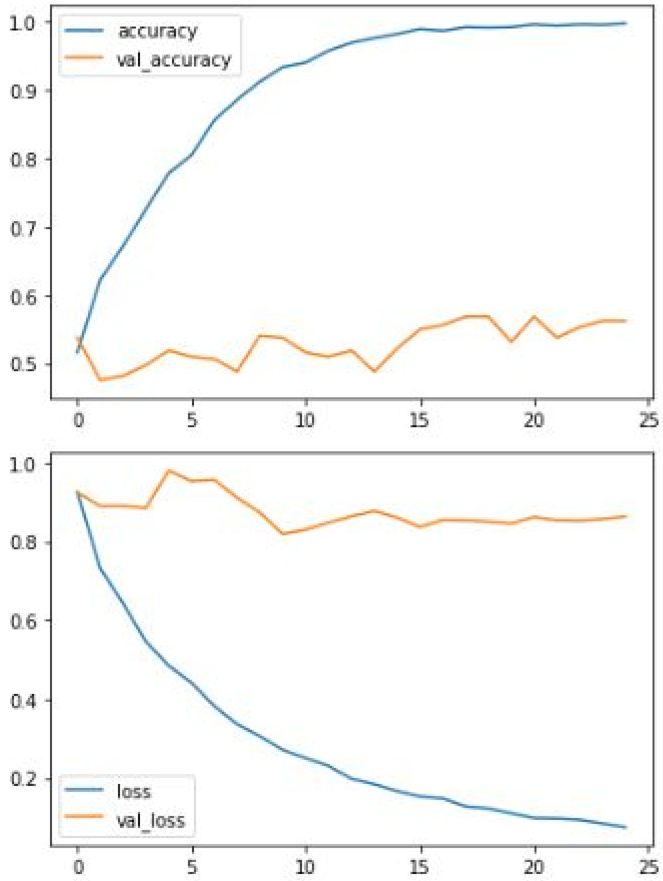


Figure 8.13 The top plot shows the training and validation accuracy and the bottom plot shows the training loss and validation loss when the network was overtrained.

9

Discussion

9.1 Evaluation of the data sets

9.1.1 The Chinese Mammography Database

The images in the CMMD were all similar in size which was an advantage since when resizing the images they all get compressed similarly. The classes were also quite balanced with only 3% more malignant images than benign. However, the data contained both mammograms with calcifications, masses, and both of them. In the folder they are split up, first comes all images with calcifications, then all that has both, and lastly all masses. If the images are not shuffled enough the training becomes hard due to training on calcifications and testing with masses and the network will not recognize the masses. But even when shuffled it makes training harder since the network has to recognize two abnormalities that look different from each other.

9.1.2 Curated Breast Imaging Subset of Digital Database for Screening Mammography

The images in CBIS-DDSM varied in pixel size, this probably makes it harder to classify after the size was decreased since all images will compress differently due to different aspect ratios. The data sets classes were not balanced either with 20% more benign images than malignant. It was also noticed that the images have some artifacts and noise that can make the images harder to train on. This database also contains different types of abnormalities, both masses, and calcifications where there are more calcifications in the right breasts than in the left ones. This data set also contains some text information on the images which can be problematic when training because the network can recognize the text instead of the abnormalities.

9.1.3 Combining data sets

Together, the data sets are balanced with only one image difference between the two classes. Due to it being text in the images from CBIS-DDSM it is crucial to make sure that all of those images do not end up in training since it probably would affect how the network recognizes the different classes.

9.2 Performance

9.2.1 Simple CNN

9.2.1.1 CMMD

The result from the Simple CNN trained with CMMD got the second-best result when looking at the AUC value (58.2%) and the confusion matrix 8.3 looks good in comparison to the other networks. It tends to classify more images as malignant than as benign but the majority of the images from each class got classified correctly. Looking at the learning curves, Figure 8.1, it looks like the loss goes down and up again which is a sign of overtraining. But in reality, this line is pretty much flat even if it does not look like it, due to the y axis being zoomed in.

9.2.2 Transfer learning

When using transfer learning, there are different models to choose from. In this thesis, both VGG16 and Resnet50 were tried out in addition to MobileNet. The reason why MobileNet was used in the end, was that it was the fastest and gave slightly better results. For example, Resnet50 took more than 8 hours to finish training. During these 8 hours, the IDE often crashed or stopped for various reasons. This made it hard to collect a result, and therefore there is no result from another transfer learning model than MobileNet.

9.2.2.1 Transfer learning with CMMD as data

The results from the transfer learning network trained with only the CMMD showed the best results with an AUC of 59.9%. The reason for this might be that all the data was from the same data set and the images were captured in the same way with the same size of all images, and therefore the images are getting compressed in the same way. The validation loss curve is rather flat, and looking at the confusion matrix, Figure 8.6a and 8.6b, it can be seen that most of the images are predicted as benign.

9.2.2.2 Transfer learning with CMMD as train data and CBIS-DDSM as test data

Looking at the confusion matrices, Figure 8.9a and 8.9b, it can be seen that the validation data gets predicted as benign but the test data gets predicted as malignant. This must be due to the test data and validation data belonging to different data sets. This also means that the network does not generalize well. A reasonable reason for this could be that the data from the two different datasets are too different from each other, for example, does the CBIS-DDSM data contain text that the data from CMMD does not. The network got an AUC of 53.4% and an accuracy of 57% which is over the rate of just guessing but not far over.

9.2.2.3 CMMD and CBIS-DDSM combined

The combined data has the best-looking learning curves but might be on the line to under-fit according to Figure 4.4. The AUC value is higher when combining the data sets instead of using the CBIS-DDSM as test data, which makes sense because the network has seen both data sets before. Looking at the confusing matrix, it can be seen that the network classifies more images as benign than malignant.

9.3 Overtraining

In the beginning, it was a struggle to get an accuracy higher than 50%. Due to this, a network was made to overtrain to see if the network could learn more or if about 50% was the maximum. As seen in the result, Figure 8.13 shows that the training accuracy goes up to almost 100%. The validation accuracy stays between 50% and 60%, just like the other networks did at the beginning and not far from what the results showed.

9.4 Simple CNN vs transfer learning for CMMD

The CMMD data set was used alone when training both the Simple CNN and the transfer learning network had similar results. The AUC value of the two networks was pretty close to each other, 58.2% and 59.9%, but the transfer learning network performed slightly better. While looking at the confusion matrices for the test data 8.3 and 8.6a the two networks performed equally well with classification.

10

Conclusion and Further Work

10.1 Conclusion

The result obtained in this thesis is not as good as the related work, which has better performance overall. This might be due to using whole mammography images to train the network, which contains a lot of information instead of segmenting out the important or the interesting parts as some of the related work did. In addition to this, the images used had to be resized to a smaller size, both due to the RAM-memory size and that the MobileNet needed an input of 224x244. Some important aspects of the image might have disappeared when resizing an image that much and changing the shape from a rectangle to a square since the image gets compressed.

The highest received value of AUC was from the transfer learning network with only the CMMD data set. The second best was the Simple CNN with the CMMD data set. It can also be seen that the confusion matrices for the networks trained with only the CMMD data set were more accurately predicted than the networks trained with both data sets that tend to classify more images as benign or malignant. This indicates that the images are pretty different and that the network does not generalize well. It can also depend on that the CMMD data set had all the same original size, while the CBIS-DDSM data set had different original sizes, and therefore when the size of the images has decreased the images and therefore the tumors were compressed differently and made harder to classify.

10.2 Further work

It is believed that by further optimizing the network and using data with the correct labels, it would be possible to predict reoperations in the future. A first step could be to change the data used for training, validation, and testing while keeping the same network to see how and if it can learn from the new data. Furthermore, it could be tested to use transfer learning with the networks already created.

In the future, if continue to work with this thesis, the most important thing is to incorporate data with information regarding reoperations of adequate size, such as the Optimam data set. It should also be considered to develop or incorporate segmentation in an attempt to increase the performance of the network. The CBIS-DDSM data set has a region of interest (ROI) mask that could be used, and since this data set has few images, cross-validation could be a way to increase the data. Another suggested thing is to use an alternative integrated development environment (IDE) than Google Colab, due to it crashing and losing connection regularly. It also only offers a RAM size of 12.68 GB which was not enough but still more than the computer that was used had.

Figure 3.3 shows a mammogram of a dense breast, if compared to Figure 3.4 with a tumor, it can be seen that it is hard to separate what is a tumor and what is dense tissue and therefore making it harder to find the tumor and the outer regions of the tumor. This might make further work with the classification of reoperation harder than just changing the labels. However, the results of this thesis can be seen as a pre-training for a network that can be reused for this new task by using i.e. transfer learning.

Bibliography

[1] Folkhälsomyndigheten. Dödlighet i bröstcancer, 2021. URL `https://www.folkhalsomyndigheten.se/folkhalsorapportering-statistik/tolkad-rapportering/folkhalsans-utveckling/resultat/halsa/brostcancer-dodlighet/`. (accessed: 22-01-17).

[2] Cancerfonden. Bröstcancer, 2021. URL `https://www.cancerfonden.se/om-cancer/cancersjukdomar/brostcancer`. (accessed: 22-02-07).

[3] Nationellt kvalitetsregister för bröstcancer. Enbart en operation (ingen omoperation p.g.a. tumördata) i bröst. URL `https://statistik.incanet.se/brostcancer/`. (accessed: 22-01-17).

[4] Peter Gärdenfors Gärdenfors, Jonas Skeppstedt, and Christian Balkenius. artificiell intelligens. URL `https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/artificiell-intelligens`. (accessed: 2022-05-27).

[5] Jörgen Malmquist Karin Söderlund Leifler Torsten Landberg. Bröstcancer. URL `https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/br%C3%B6stcancer`. (accessed: 22-02-07).

[6] Breast anatomy, Feb 2022. URL `https://www.nationalbreastcancer.org/breast-anatomy`. (accessed: 2022-03-16).

[7] Breast tumors, Aug 2021. URL `https://www.nationalbreastcancer.org/breast-tumors/`. (accessed: 2022-03-16).

[8] Breast cancer mammogram: How does a mammogram work? URL `https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/mammograms/mammogram-basics.html`. (accessed: 2022-03-16).

[9] MS Jaime R. Herndon. What do breasts look like on a mammogram? URL `https://www.verywellhealth.com/mammogram-images-descriptions-and-details-4020351`. (accessed: 2022-03-22).

[10] S. Kevin Zhou, Hayit Greenspan, and Dinggang Shen. *Deep Learning for Medical Image Analysis*. Elsevier Inc., 2017.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`, (accessed: 22-02-29).

[12] Michelle Torres and Francisco Cantú. Learning to see: Convolutional neural networks for the analysis of social science data. *Political Analysis*, 30(1):113–131, 2022. doi:10.1017/pan.2021.9.

[13] Mattias Ohlsson and Patrik Edén. Introduction to artificial neural networks and deep learning, 2021. (accessed: 22-02-21).

[14] Tarang Shah. About train, validation and test sets in machine learning, 12 2017. URL `https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7`. (accessed: 2022-04-20).

[15] Sagar Sharma. Epoch vs batch size vs iterations, 9 2017. URL `https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9`. (accessed: 2022-04-20).

[16] Hafidz Zulkifli. Understanding learning rates and how it improves performance in deep learning, Jan 2018. URL `https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10`. (accessed: 2022-05-27).

[17] Christophe Pere. What are loss functions?, 6 2020. URL `https://towardsdatascience.com/what-is-loss-function-1e2605aeb904`. (accessed: 2022-04-04).

[18] Gaurav Singh. From sgd to adam, 5 2020. URL `https://medium.com/mdr-inc/from-sgd-to-adam-c9fce513c4bb`. (accessed: 2022-04-03).

[19] Timothy Dozat. Incorporating nesterov momentum into adam. 2016.

[20] François Chollet. Transfer learning fine-tuning, 04 2020. URL `https://keras.io/guides/transfer_learning/`. (accessed: 22-02-21).

[21] *ImageNet*. URL `https://image-net.org/index`. (accessed: 2022-05-25).

[22] Mobilenet and mobilenetv2. URL `https://keras.io/api/applications/mobilenet/`. (accessed: 22-03-08).

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv. doi:10.48550/ARXIV.1409.1556. URL `https://arxiv.org/abs/1409.1556`.

[24] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. volume 1, pages 541–551, 1989. doi:10.1162/neco.1989.1.4.541.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi:10.1109/CVPR.2016.90.

[26] Panagiotis Tzirakis, Stefanos Zafeiriou, and Björn Schuller. Chapter 18 - real-world automatic continuous affect recognition from audiovisual signals. In Xavier Alameda-Pineda, Elisa Ricci, and Nicu Sebe, editors, *Multimodal Behavior Analysis in the Wild*, Computer Vision and Pattern Recognition, pages 387–406. Academic Press, 2019. ISBN 978-0-12-814601-9. doi:https://doi.org/10.1016/B978-0-12-814601-9.00028-6. URL `https://www.sciencedirect.com/science/article/pii/B9780128146019000286`.

[27] Jason Brownlee. How to use learning curves to diagnose machine learning model performance, Aug 2019. URL `https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/`. (accessed: 2022-04-22).

[28] Sarang Narkhede. Understanding auc - roc curve, Jun 2021. URL `https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5`. (accessed: 2022-04-22).

[29] Mark D Halling-Brown, Lucy M Warren, Dominic Ward, Emma Lewis, Alistair Mackenzie, Matthew G Wallis, Louise S Wilkinson, Rosalind M Given-Wilson, Rita McAvinchey, Kenneth C Young, and et al. Optimam mammography image database: A large-scale resource of mammography images and clinical data. *Radiology. Artificial intelligence*, Nov 2020. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8082293/`.

[30] Hongmin; Fan Zhihao; Zhang Ling; Dan Tingting; Li Jiao; Wang Jinghua. Cui, Chunyan; Li Li; Cai. The chinese mammography database (cmmd): An online mammography database with biopsy confirmed types for machine diagnosis of breast. The Cancer Imaging Archive., 2021. doi:https://doi.org/10.7937/tcia.eqde-4b16.

[31] Assaf Hoogi Daniel Rubin Rebecca Sawyer Lee, Francisco Gimenez. Curated breast imaging subset of ddsm [dataset]. The Cancer Imaging Archive, 2016. doi:https://doi.org/10.7937/K9/TCIA.2016.7O02S9CY.

[32] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL `https://arxiv.org/abs/2006.07733`.

[33] John D. Miller, Vignesh A. Arasu, Albert X. Pu, Laurie R. Margolies, Weiva Sieh, and Li Shen. Self-supervised deep learning to enhance breast cancer detection on screening mammography. arXiv, 2022. doi:10.48550/ARXIV.2203.08812. URL `https://arxiv.org/abs/2203.08812`.

[34] Kevin Wu, Eric Wu, Yaping Wu, Hongna Tan, Greg Sorensen, Meiyun Wang, and Bill Lotter. Validation of a deep learning mammography model in a population with low screening rates. arXiv, 2019. doi:10.48550/ARXIV.1911.00364. URL `https://arxiv.org/abs/1911.00364`.

[35] Benjamin Stadnick, Jan Witowski, Vishwaesh Rajiv, Jakub Chłędowski, Farah E. Shamout, Kyunghyun Cho, and Krzysztof J. Geras. Meta-repository of screening mammography classifiers. arXiv, 2021. doi:10.48550/ARXIV.2108.04800. URL `https://arxiv.org/abs/2108.04800`.

[36] Tf.keras.preprocessing.image.imagedatagenerator nbsp;: nbsp; tensorflow core v2.9.1. URL `https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator`. (accessed: 2022-05-27).

[37] scikit-learn. URL `https://scikit-learn.org/stable/`. (accessed: 2022-05-27).