

# PREOPERATIVE PREDICTION OF SENTINEL NODAL STATUS USING MAMMOGRAPHY IMAGES

MALIN HJÄRTSTRÖM , MAREN HØIBØ

Master's thesis  
2021:E37



LUND INSTITUTE OF TECHNOLOGY  
Lund University

Faculty of Engineering  
Centre for Mathematical Sciences  
Mathematical Statistics

## Abstract

Breast cancer is the most common cancer among women in Sweden, accounting for approximately 30 % of the cancer cases. The overall prognosis is good, but worsens if the cancer metastasizes from the primary tumor. In order to exclude or confirm lymph node metastasis in clinically node negative breast cancer, axillary lymph nodes are examined by sentinel lymph node biopsy (SLNB). Up to 85 % of the patients have benign sentinel nodes, and do not benefit therapeutically from SLNB. This project was part of a goal to decrease unnecessary surgeries by preoperative prediction of sentinel nodal status.

A cohort of 800 patients, diagnosed with primary breast cancer in Scania, Sweden, between 2009 and 2012 was studied. The cohort was previously used by Dihge et al. to predict axillary lymph node metastasizing, using a multilayer perceptron (MLP) on clinicopathological data. The aim of this project was to determine whether including information from mammograms would improve the artificial neural network's prediction. A similar MLP to that of Dihge et al. was constructed, and convolutional neural networks were used to extract features from the mammography images ( $n = 705$ ). The features were used as additional input to the MLP. The results were evaluated with area under the ROC curve (AUC) score.

The addition of features from mammograms did not improve the predictions. The MLP's AUC-score without features from mammograms was 0.7190 (std 0.0465), it decreased to 0.6573 (std 0.0470) when features from mammography images were added. Nevertheless, the results have demonstrated behaviors of the models and may therefore be used to guide future attempts at using mammograms to improve sentinel lymph node prediction.

## **Acknowledgements**

We would like to thank our supervisors Mattias Ohlsson and Professor, Senior Consultant Lisa Rydén for guidance and support throughout the project. Thank you, Mattias, for valuable discussions regarding the machine learning challenges we encountered. Thank you, Lisa, for sharing your vast knowledge on breast cancer and breast cancer surgery. We would also like to thank M.D. PhD Locket Dihge for providing us with background information on the clinicopathological data used. Thank you, PhD Magnus Dustler, for discussions regarding the use of mammograms, for providing us with extra computer power, and for starting TeamViewer for us day after day.

The work has been divided equally between the authors.

## Nomenclature

The following definitions are used throughout the report:

**Iteration** Training over all epochs once using the same hyperparameter combinations and values.

**Architecture** The structure of the model (i.e. number, types and properties of layers).

**Model** An architecture trained on a specific dataset.

## List of acronyms

**Adam** Adaptive Moment Estimation

**ALND** Axillary Lymph Node Dissection

**ANN** Artificial Neural Networks

**AUC** Area Under the ROC Curve

**CC** Craniocaudal

**CNN** Convolutional Neural Networks

**ER** Estrogen Receptor

**FNR** False Negative Rate

**GAP** Global Average Pooling

**HER2** Human Epidermal Growth Factor 2

**LVI** Lymphovascular Invasion

**ML** Mediolateral

**MLO** Mediolateral Oblique

**MLP** Multilayer Perceptrone

**PR** Progesteron Receptor

**ROC** Receiver Operating Characteristic

# Neuronnätverk för prediktion av portvaktkörtelstatus

Vid operation av brösttumör utförs alltid ett ingrepp i armhålan för att undersöka om cancer har spridit sig till lymfkörtlarna. Detta ingrepp kan ge onödiga biverkningar, varför preoperativ diagnostisering av metastasering är eftersträfvansvärt. Vi har undersökt om prediktion av spridning med hjälp av neuronnätverk kan förbättras om både klinisk data och mammografibilder används, jämfört med om nätverket endast tränas på klinisk data.

Bröstcancer är den cancersjukdom som skördar flest kvinnoliv<sup>1</sup> i världen. I Sverige utgör sjukdomen ungefär 30 % av all cancer som drabbar kvinnor. Den generella prognosen är god, till stor del tack vare tidig diagnostisering. Sprider sig cancer till andra delar av kroppen försämrar prognosen, därför är portvaktkörtelbiopsi viktig för att bestämma terapeutiska insatser. Ingreppet innebär att 1-4 av de lymfnoder i armhålan som först nås av eventuell metastasering undersöks genom kirurgi. Biopsin i armhålan sker alltid i samband med bortopereringen av brösttumören, och kan medföra betydande men för patienten i form av svullnad och minskad rörlighet i arm- och axelparti.

Bröstcancer metastaserar inte för upp emot 85 % av de drabbade och för dessa patienter har inte portvaktkörtelbiopsi någon terapeutisk fördel. Därför pågår forskning för att avgöra om spridning av cancer skett, utan att göra ingrepp i kroppen. Dihge et al. (2019) [2] har undersökt möjligheten att prediktera denna metastasering med hjälp av artificiella neuronnätverk. De använde sig av data om patienten och tumören för att skapa en olinjär modell, med målet att kunna avgöra om spridning skett med samma säkerhet som med portvaktkörtelbiopsi. Forskarna lyckades prediktera spridning med relativt god säkerhet, och metoden visar potential för att kunna minska antalet onödiga portvaktkörtelbiopsier. Den data forskarna använde var inhämtad efter operation av brösttumören, men de flesta variabler går att mäta redan innan operation genom vävnadsbiopsi av tumören. Forskarna vill använda preoperativa variabler och därmed helt undvika ingreppet i armhålan.

Syftet med vårt projekt var att vidareutveckla Dihge et al.'s modell genom att tillföra data. En modell liknande Dihge et al.'s modell utvecklades för att kunna jämföra resultaten inom projektet. Utöver kliniskt data användes mammografibilder för att prediktera statusen hos lymfkörtlarna. Informationen från mammografibilderna erhöles genom att filtrera ut viktiga egenskaper hos bilderna med hjälp av convolutional neural networks. Vi utvecklade tre olika nätverk av denna typ för att undersöka deras förmåga att hitta generella egenskaper hos mammogrammen. Nätverken skiljde sig åt utifrån storleken på indata, och i ett fall användes en färdig modell som var förtränad på externa data. Det visade sig vara svårt för modellerna att skilja mellan spridning och icke-spridning utifrån mammogrammen. De identifierade egenskaperna hos bilderna från ett av nätverken överfördes till ett nätverk av typen multilayer perceptron. Detta nätverk tränades på både patient- och tumördata och informationen från mammografibilderna. Denna modell gav ett sämre resultat än modellen som efterliknade Dihge et al.'s modell, vilket inte var förvånande givet att datan från mammografibilderna inte verkade innehålla information om metastasering. Resultatet från projektet har synliggjort beteenden hos modellerna, vilket kan bidra till det fortsatta arbetet för preoperativ prediktion av bröstcancermetastasering till lymfkörtlarna i armhålan.

---

<sup>1</sup>Statistiken över bröstcancer är hämtad från [1] som redovisar binära könsidentiteter.

## Table of contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Deep learning . . . . .	5
2.2	Multilayer perceptron . . . . .	5
2.3	Activation functions . . . . .	6
2.4	Loss function . . . . .	7
2.5	Optimization function . . . . .	7
2.6	Convolutional neural networks . . . . .	8
2.7	Generalization . . . . .	9
2.7.1	AUC-score . . . . .	9
2.7.2	K-fold cross validation . . . . .	10
2.8	Regularization . . . . .	10
2.8.1	L2-regularization . . . . .	11
2.8.2	Dropout . . . . .	11
2.9	Transfer learning . . . . .	11
<b>3</b>	<b>Data</b>	<b>13</b>
3.1	Clinicopathological data . . . . .	13
3.2	Mammography images . . . . .	15
<b>4</b>	<b>Method</b>	<b>16</b>
4.1	Pipeline . . . . .	16
4.2	Convolutional neural networks . . . . .	18
4.2.1	Convolutional neural network with images resized to 600 · 400 . .	18
4.2.2	Convolutional neural network pretrained with InceptionV3 . . . .	24
4.2.3	Convolutional neural network with image patches . . . . .	25
4.3	Multilayer perceptrons . . . . .	28
4.3.1	Benchmark multilayer perceptron . . . . .	29
4.3.2	Final multilayer perceptron . . . . .	31
<b>5</b>	<b>Results</b>	<b>32</b>
5.1	Convolutional neural networks . . . . .	32
5.1.1	Convolutional neural network with images resized to 600 · 400 . .	32
5.1.2	Convolutional neural network with image patches . . . . .	32
5.2	Multilayer perceptrons . . . . .	33
5.2.1	Benchmark multilayer perceptron . . . . .	33
5.2.2	Final multilayer perceptron . . . . .	33
<b>6</b>	<b>Discussion</b>	<b>34</b>
<b>7</b>	<b>Conclusion</b>	<b>36</b>
<b>8</b>	<b>Future perspectives</b>	<b>36</b>
	<b>References</b>	<b>37</b>



# 1 Introduction

Breast cancer is the most frequently diagnosed cancer worldwide, and the leading cause of cancer death in women [1]. In Sweden, the disease accounts for approximately 30 % of all cancer in women [3]. However, the overall prognosis of the disease is good, and the cancer is often diagnosed in an early stage [4]. The diagnosis is made using the gold standard triple assessment, comprising mammography images including ultrasound of the axilla, physical examination and biopsy and/or cytology of the tumor [5, 6]. When a cancer metastasizes from the primary tumor, the prognosis worsens [7]. Lymph nodes close to the primary tumor are often the first organs to which cancer spread. In the case of breast cancer, the lymphatic fluid is drained uniformly to a few lymph nodes, called *sentinel lymph nodes*, see figure 1 [8, 9]. There are multiple factors that may increase the risk of axillary sentinel lymph node metastasis [10]. These include tumor size, lymphovascular invasion (LVI), multifocality, estrogen receptor (ER), progesterone receptor (PR), HER2, Ki-67, histological grade, histological type and location of the tumor in the breast [10].

Tumor size is defined as the greatest diameter of the breast tumor; the smaller the tumor, the less likely lymph node involvement is. The risk of metastasizing to the lymph nodes is higher when there is tumor deposit present within an endothelial-lined space in the breast tissue surrounding the cancer. This state is called lymphovascular invasion. Multifocality, defined as having two or more invasive tumors within the same breast quadrant, could be a predictor of nodal metastasis. Studies conducted on the effect of presence of the estrogen receptor and progesterone receptor status on axillary metastasis have not been consistent, but most of them present an association with presence of hormonal receptors and risk of nodal metastasis. The oncogene HER2 is overexpressed in 15 – 30 % of all breast cancer cases. These patients need therapies directed to the HER2 to improve their prognosis. Some reports suggest that HER2-positive tumors are connected to increased metastasizing, while others could not display a difference in lymph node status with HER2-negative and HER2-positive tumors. The protein Ki-67 is expressed in some of the phases of a proliferating cell, and has been shown to relate to nodal metastasis. The histological grade is determined by combining three morphological features of the tumor. An earlier report suggested that the histological grade is an independent predictor of nodal metastasis, but a more recent publication did not confirm this [10]. The growth pattern of the breast tumor is called the histological type. There are many different histopathological types, for example ductal and lobular. Some studies have suggested that nodal metastatic load is higher in lobular cancer than in ductal, while others show the opposite, or no difference in axillary metastasis [10]. Breast cancer is most likely to occur in the upper, outer quadrant of the breast. The reason for this is so far unknown, but this group of patients display a better prognosis compared to patients with tumors located in other quadrants of the breast [10].



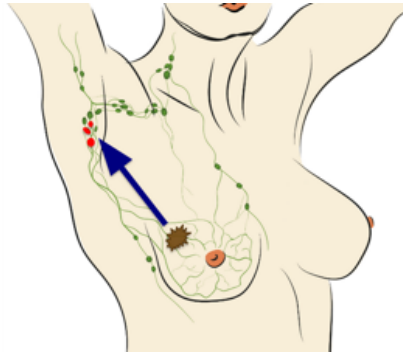


Figure 1: The axillary sentinel lymph nodes are the first lymph nodes to which breast cancer spreads. The tumor is represented by a brown star shape, and the sentinel lymph nodes are marked in red. Image courtesy of Dihge, L.

To confirm or exclude lymph node metastasis, axillary lymph nodes are examined routinely after surgical excision of them. If there are clinical findings of enlarged lymph nodes, the nodes are excised by axillary lymph node dissection (ALND). During an ALND, ten or more nodes are removed [9]. If there are no clinical signs of metastasis, the sentinel lymph nodes are examined by sentinel lymph node biopsy (SLNB), where 1-4 sentinel lymph nodes are removed [4]. The accuracy of surgical management in SLNB is measured by the false negative rate (FNR) and ranges between 5 – 10 %, where the lower rates are achieved by more experienced surgeons [10]. The rate describes the percentage of the node-positive patients that are not diagnosed as such.

The surgical intervention is associated with considerable morbidity [11]. Soon after the operation, the patient can suffer from seroma (affecting 20 – 30 % of the patients), bleeding and infections (affecting less than 5 % of the patients). Late morbidity includes decreased shoulder mobility, hypersensitive skin, decreased sensibility and swelling of the arm. The symptoms can vary from being minor to patients suffering from severe lymphedema [11]. Today, the overall node-positive rate of primary breast cancer is about 15 – 30 %, resulting in up to 85 % of primary breast cancer patients not having nodal metastasis. For these node-negative patients, the surgical intervention has no therapeutic benefit [2].

To avoid unnecessary surgical intervention, the axillary status would need to be predicted preoperatively using non-invasive methods. There have been different approaches to develop predictive models that can replace SLNB with retained or improved sensitivity. Examples are models with data gathered from using magnetic resonance imaging (MRI) [12] and ultrasonography [13]. Developing a predictive model using mammography images may be difficult, since even hospital personnel cannot determine nodal status from this data with certainty. Hu et al. (2021) [14] recently published a study where predictions using multivariate logistic regression on ultrasonography data reached an FNR of 7 %. Dihge et al. (2019) [2] compared the use of multivariable logistic regression and artificial neural networks (ANN) to predict axillary status. Using postoperative clinicopathological data, the ANN models showed better performance in discriminating nodal status end-points than corresponding multivariable logistic regression models. Using an FNR of 5 – 10 %, Dihge's model could reduce the number of SLNB with 8-27 % and thus decrease the number of unnecessary SLNB. The future goal is to extend the project to preoperative variables obtained from

mammography imaging and tissue biopsy, and thus be able to preoperatively predict sentinel nodal status. In Dihge et al.'s study, the internally validated area under the curve score (AUC-score) to distinguish disease-free axilla (N0) versus nodal metastasis (N+), was 0.74 (95% CI 0.72 – 0.76) [2]. Possible improvements of the model or additional data could potentially reduce the rate of unnecessary SLNB further.

This project builds upon the result of Dihge et al.. The aim is to investigate whether ANN predictions to determine axillary lymph node status can be improved by combining clinicopathological data with features from mammograms. This will be done by investigating the potential of different architectures, hyperparameter values and pre-processing alternatives. The project uses the same cohort as Dihge et al., and is part of an overall aim to decrease the number of unnecessary surgeries for patients with benign axillary sentinel lymph nodes. Multiple models will be developed to extract information from mammography images and to predict the sentinel nodal status. Figure 2 shows the input data that the final neural network will train on to perform the predictions.

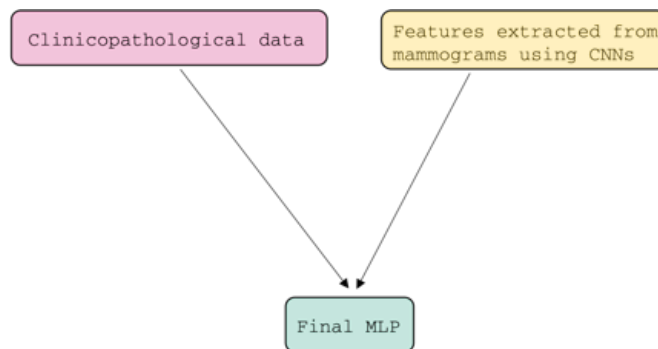


Figure 2: Dihge et al. [2] have explored the possibility of using clinicopathological data to train an MLP for preoperative prediction of sentinel nodal status. The aim of this project is to investigate whether the performance of the MLP can be improved when adding features from corresponding mammography images as input data. The features will be extracted using CNNs.

The result of Dihge et al. [2] shows that discriminating limited axillary nodal metastasis (N1) from disease-free axilla (N0) is harder than discriminating other nodal end-points with a higher number of affected lymph nodes. Therefore, this study was limited to investigating the end-points disease-free axilla versus nodal metastasis (N0 vs N+). Mammography images are usually taken from three views: craniocaudal (CC), mediolateral (ML) and mediolateral oblique (MLO). To limit the complexity of the input data, and thereby increase the chances of identifying patterns in the images, only one view was used. After discussions with Magnus Dustler, PhD, Research associate at Diagnostic Radiology in Malmö, Lund University, Sweden, the craniocaudal view was chosen, as this view is most likely to include useful information that the model could interpret. The study was conducted in accordance with the Declaration of Helsinki. Study approval date: 2012 – 08 – 15, by the Regional Ethical Review Board in Lund, Sweden. Registration number: 2012 – 340. KVB permission number allowing extraction and use of mammograms: 006 – 21.

First, background information on deep learning, two different types of neural net-

works and mathematical concepts, is presented, as well as a short description of generalization and regularization of networks. Transfer learning is briefly introduced. Then follows a section about data and methodology, respectively. The result is presented and discussed, and the report ends with a short conclusion and future perspectives.

## 2 Background

To investigate the problem of distinguishing disease-free axilla (N0) from nodal metastasis (N+), a short overview of the theoretical framework of deep learning for supervised binary classification is presented. Two types of models are introduced; the multilayer perceptron and the convolutional neural network. For a binary classification problem, the target is assumed to be Bernoulli distributed [15], and an introduction to common activation, optimization and loss functions for Bernoulli distributed data is given. Regularization techniques and performance measures are presented. Lastly, transfer learning, a technique used to take advantage of already trained models, is briefly described.

### 2.1 Deep learning

In machine learning, computer algorithms are used to find patterns in data. These patterns are automatically identified by letting the algorithm learn features on a set of training data. Thereafter, corresponding features may be detected in similar datasets, even if the algorithm has not seen that exact data before [16]. The technique can be used for a wide range of purposes. One common task is classification, where the model learns to classify unseen data into different categories. If the model has access to the true categories of the data, the method is called supervised learning [17, 18]. In supervised learning, the true target and the model prediction can be compared to evaluate the performance of the model.

Deep learning is a type of machine learning method which allows computers to learn a hierarchy of features, so that more complex patterns can be identified in the data. This has shown to be beneficial in for example speech recognition, computer vision and natural language processing [19]. In the hierarchical model, each feature can be understood in relation to a simpler one. So called hidden layers are placed between the input and the output layer of the model, where more complex concepts can be learned when reaching deeper layers. This hierarchical learning architecture is inspired by deep layered learning process of the human brain, where the primary sensorial areas of the neocortex automatically extracts features and abstractions from the underlying data [19]. When a computer gathers knowledge in this way, it eliminates the necessity of humans to specify features and patterns for the computer, enabling the computer to learn on its own [18].

### 2.2 Multilayer perceptron

A multilayer perceptron (MLP) is one of the most quintessential machine learning methods [18]. The model is considered a deep neural network after reaching a certain depth, although there are ambiguities in how many hidden layers the model needs to be considered deep. The MLP consists of an input layer, a number of hidden layers and an output layer [15], where each layer contains a number of nodes. Each node-to-node connection between the layers is weighted, see figure 3. The output of the MLP can be expressed as a sum of the incoming hidden nodes and weights, fed into an activation function, see equation 1,

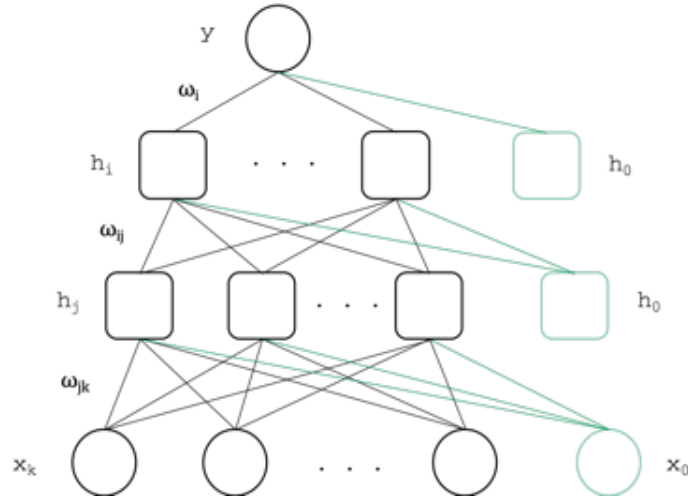


Figure 3: Example of a multilayer perceptron with two hidden layers.  $x$  represents the input layer,  $h$  the two hidden layers,  $\omega$  the weights and  $y$  the output layer.  $h_0$  and  $x_0$  represent bias nodes.

$$y(\mathbf{x}) = \varphi_o \left( \sum_i h_i \omega_i + h_0 \right) \quad (1)$$

where  $y$  is the output,  $\varphi_o$  the output activation function,  $h_i$  the incoming hidden nodes and  $w_{ni}$  the weights of the incoming nodes. The hidden nodes can be expressed as a sum of their incoming nodes and weights fed into their respective activation functions in the same manner. The bias nodes,  $h_0$  and  $x_0$ , are added as offsets to the input layer and to each hidden layer to make sure that the weights can affect the output of the layer, independently of the input value.

The weights of the network are updated during training until a loss function reaches its minimum [17]. The loss function describes the relationship between the true target and the prediction of the model, and needs to be differentiable. The loss function can then be differentiated with regard to the weights, finding the optimal weights by obtaining the minimum of the function. The procedure in which the derivatives of the loss function are evaluated is called back-propagation [17]. The parameters of the network that are fixed during training are called hyperparameters. The hyperparameters need to be tuned before training the model and their properties are often important for the model's ability to perform well.

### 2.3 Activation functions

The activation function adds non-linearity to the network. Without non-linear activation functions, a deep network can always be represented by a network without hidden layers, since successive linear transformations is a linear transformation itself [17]. The non-linearity also enables the network to learn more complicated features [15]. Deep neural networks often use the non-linear function rectified linear unit (ReLU) as activation function for the hidden layers, see equation 2,

$$\varphi(a) = \max(0, a) \quad (2)$$

where  $\varphi(a)$  equals 0 for  $a < 0$  and  $a$  otherwise. Different activation functions are used in the output layer depending on the task of the network. When the data is assumed to be Bernoulli distributed, the sigmoid activation function is used to categorize the input data into one out of two categories, see equation 3,

$$\varphi(a) \equiv \frac{1}{1 + e^{-a}} \quad (3)$$

where  $\varphi(a)$  is the activation function evaluated in  $a$ . A plot of the ReLu function and the sigmoid function, respectively, can be seen in figure 4.

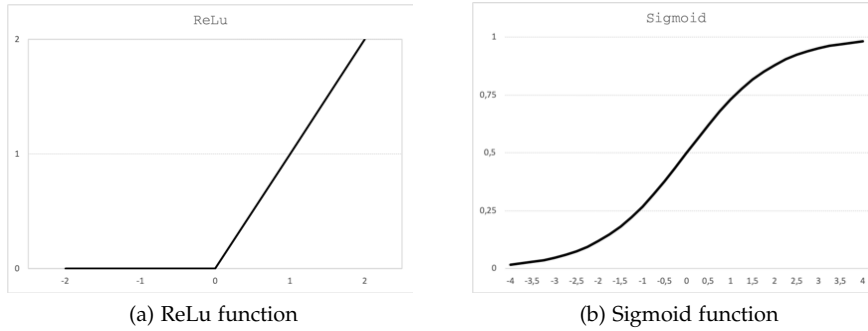


Figure 4: Activation functions commonly used in neural networks. The ReLu function is often found in the hidden layers, while the sigmoid function classifies binary data in the output layer.

## 2.4 Loss function

The loss function describes the relationship between the prediction and the true target [17]. Minimizing the loss function for a binary classification function can be expressed as minimizing the negative log likelihood of observing either target value [17], leading to the binary cross-entropy loss function (equation 4),

$$E = - \sum_n [t_n \ln(y_n) + (1 - t_n) \ln(1 - y_n)] \quad (4)$$

where  $E$  is the loss function,  $t_n$  is the true target and  $y_n$  is the prediction for pattern  $n$ . The cross-entropy function reaches its minimum when all  $y_n$  are equal to their respective  $t_n$  [17, 15].

## 2.5 Optimization function

An optimization function controls how the loss function is minimized. Some optimization algorithms use momentum, which suppresses fluctuations in the gradients so that the loss function can converge faster. The adaptive moment estimation (Adam) algorithm uses momentum and an adaptive learning rate for different parameters, and has shown to efficiently solve deep learning problems, both in MLPs and CNNs [20].

## 2.6 Convolutional neural networks

Convolutional neural networks (CNNs) are deep feed-forward networks that maintain spatial structure and information between inputs [15]. The networks are frequently used in tasks with image data and have been very successful in practical applications [18]. The network uses the mathematical operation convolution, by convolving the input with a kernel, see equation 5,

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \quad (5)$$

where  $I$  is an image convolved with a kernel,  $K$  and  $S(i, j)$  is the feature map [18]. The kernel moves across an image matrix using the same parameters at multiple locations. This is called parameter sharing. Parameter sharing reduces the memory required by the model. By convolving an input image with a kernel smaller than the input size (sparse interactions), fewer parameters are needed, which also reduces the model's memory requirement [18], see figure 5. This is beneficial for images, since they usually contain a lot of data. A convolutional layer contains a convolutional stage and an activation stage, which together are called a filter [18]. Different filters can extract different types of features in an image. Each convolutional layer may **consist of** many filters, resulting in an image with a large channel depth [15]. To decrease the time of training, one can add a batch normalization step between the convolution operation and the activation step. Batch normalization normalizes the layer's input. A convolutional network may also include a pooling stage. Pooling makes the input invariant to small translations and may reduce the data size significantly. In max pooling, the maximum value of each patch of each feature map is transferred to the next step of the network [18], thus highlighting the most present feature of every feature map, see figure 6.



Figure 5: The convolutional stage of a CNN architecture. An input image convolved with a 3 · 3 kernel and filter size 1. The kernel moves across the image matrix, reducing the number of parameters compared to fully connected networks, yielding a 5 · 4 · 1 output matrix.

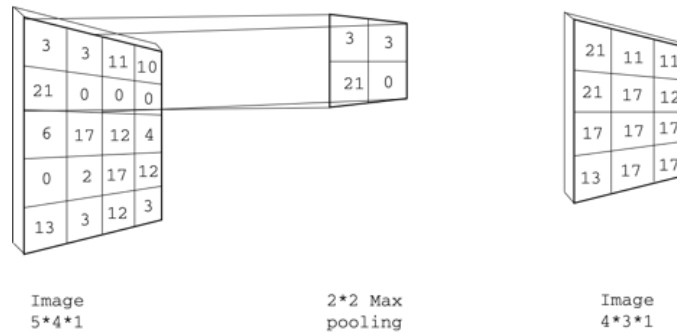


Figure 6: The pooling stage of a CNN architecture. An input image pooled with  $2 \cdot 2$  max pooling. The input image corresponds to the output in figure 5, after a ReLU activation function layer. The middle image shows the first max pooling step of the  $5 \cdot 4 \cdot 1$  image, yielding the number 21 at the top left of the right output image.

## 2.7 Generalization

For a model to be useful it needs to generalize well, i.e. have good prediction accuracy on unseen data. Therefore, datasets are divided into a test set and a training set, where the model's performance is evaluated on the test set using a performance measure, for example the AUC-score. It is assumed that the data in the training and test sets are independent of each other and that the two sets are identically distributed [18]. The training set is divided into a training and a validation set, where the validation set is used to evaluate the model's performance during the development of the model. When this performance evaluation is performed through repeated division of the training set into training and validation sets, the method is called cross validation.

### 2.7.1 AUC-score

The performance measure area under the ROC curve (AUC) tests a model's discriminative ability [21], see figure 7. The receiver operating characteristic (ROC) curve describes the relationship between the sensitivity (true positive rate) and  $1$ -specificity (false positive rate), and the area under the curve provides a score of how well the model can separate between the two categories. In other words, the AUC-score tests the overall usefulness of a model. Maximum sensitivity is achieved when all truly positive results are being targeted as positive, i.e. there are no false negative results [22]. In healthcare, it is especially important to identify all diseased patients, which is why a low false negative rate is strived for in for example SLNB. The false negative rate is the percentage of the total amount of positive cases that are not identified as positive [22]. Maximum specificity is achieved when there are no false positive results, corresponding to no healthy patients being targeted as having the disease. A well generalized model yields a high AUC-score for the validation data.



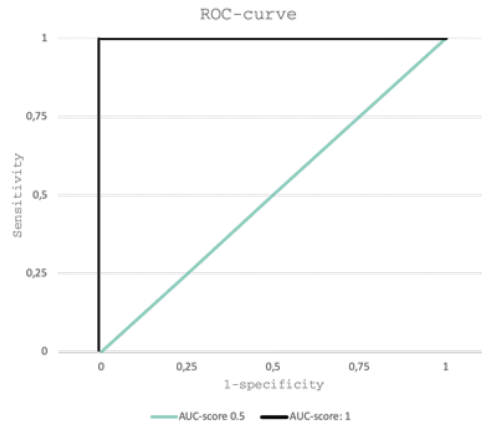


Figure 7: ROC-curve with perfect and random binary classification. The black line illustrates perfect classification (AUC-score 1), and the green line illustrates random classification (AUC-score 0.5)

### 2.7.2 K-fold cross validation

The generalization performance of a model indicates how well it will perform on unseen data [15]. In k-fold cross validation, the training data is split randomly into  $k$  subsets, see figure 8. Training is then performed on  $k - 1$  subsets and validated on 1 subset. This is done for each fold, where the subset being the validation subset changes for each fold. The performance is measured as an average of the performances in each of the  $k$  validation sets [15]. If one class is rare, one may perform stratified partitioning to ensure inclusion of all classes in all subsets [15].

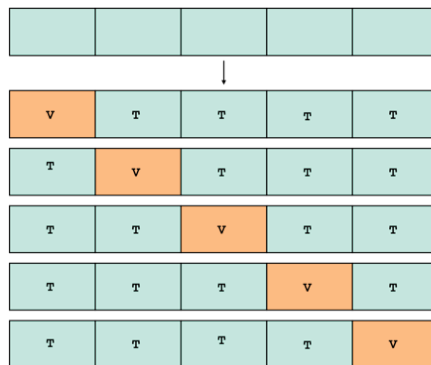


Figure 8: 5-fold cross validation. V stands for validation data and T for training data. The original training data is divided randomly into  $k = 5$  subsets so that training and validation is performed 5 times, with the subsets changing each fold.

## 2.8 Regularization

Regularization is used to neutralize overfitting. Overfitting means that the model is trained to work very well on the training data, but may perform worse on data that

differs from the training data [15]. An overfitted model may result in a low generalization performance [15] and can be identified through a high AUC-score of the training data and a low validation AUC-score. Two techniques to regularize models are L2-regularization and dropout.

### 2.8.1 L2-regularization

One regularization strategy is to add a regularization term to the loss function [15, 18], see equation 6,

$$\tilde{E}(\omega) = E(\omega) + \alpha\Omega(\omega) \quad (6)$$

where  $\tilde{E}(\omega)$  is the modified loss,  $\alpha$  the regularization strength and  $\Omega(\omega)$  the regularization term. L2-regularization is one of the most common penalties and favors small weights [15, 18], see equation 7,

$$\Omega = \frac{1}{2} \sum_i \omega_i^2 \quad (7)$$

where  $\omega_i$  are the weights in the network. The weights on features with a small covariance with the target, compared to the added regularization strength, are decreased [18].

### 2.8.2 Dropout

Dropout is another regularization strategy used to avoid overfitting [15]. The technique temporarily removes network nodes and thus forces the network to concentrate on more general features by suppressing large curvatures [15]. Given a probability to keep nodes,  $p$ , each node may or may not be removed with probability  $1 - p$ . When a node is temporarily removed, so are all its connected weights, resulting in a thinned network [15]. During training, a new thinned network is constructed for each pattern. The weights are updated on the new thinned network, and then averaged depending on the number of times they were present in all the thinned networks constructed so far [15].

## 2.9 Transfer learning

Models can be hard to train from scratch in medical applications since medical data often is limited [23]. One way to get around this problem is to use transfer learning. Transfer learning has similarities with the generalization theory of transfer [24], which suggests that learning is transferred by the generalization of experience. If there is a connection between two learning activities, such as learning to play two different music instruments, the learning of the second activity can take shorter time than the first one. Although, this is only the case if there are common grounds in the two activities; learning to play the piano will probably not result in improved skills on the football field [24]. Transfer learning from non-medical tasks to medical datasets has shown good results [25].

In practice using transfer learning, a network is trained on a large dataset where it learns features that are general for that specific data. This pretrained model can then be used for machine learning tasks on similar datasets. To adapt a pretrained model to

a new dataset, the top layers of the architecture can be modified or exchanged. Another possibility is to unfreeze the weights in a number of the topmost layers and fine-tune the network to the new data [26].

There are public online databases, such as the ImageNet [27] database, containing tens of millions of labeled images, that can be used to create models for transfer learning. Top performing models trained on ImageNet can be downloaded using the deep learning API Keras [28, 29]. One such model is InceptionV3, that has previously been used in medical applications, see for example [30, 31]. It is pretrained to classify images into 1000 different classes and has a top-5 accuracy of 0.937 on the ImageNet validation dataset [28].

## 3 Data

The data consists of a clinicopathological dataset and mammograms from the same cohort of patients. The data was collected in a consecutive and prospective way and gathered in population based registries. All patients were diagnosed with primary invasive breast cancer between 2009 and 2012 in mid Scania, Sweden (Mellersta Skåne). Approximately 35% of the patients in the final cohort were node positive.

### 3.1 Clinicopathological data

The clinicopathological dataset was collected by M.D. PhD Looket Dihge. The cohort consisted of 800 cases, where exclusion criteria included *men, previous ipsilateral breast- or axillary surgery, omission of surgical axillary staging, neoadjuvant chemotherapy and clinical axillary lymphadenopathy*. Data including age at diagnosis, menopausal status, weight and height, tumor localization in the breast and clinical axillary status was retrieved from medical records. The mode of detection (screening or symptomatic presentation) was retrieved from The Swedish National Quality Registry for Breast Cancer. The histopathological variables were extracted by a breast pathologist from the surgical specimen of the breast and axilla, and included tumor size, multifocality, histopathological subtype, histological grade, status of ER, PR, HER2 and Ki-67, occurrence of LVI and pathological nodal status [10]. The retrieved clinicopathological data can be seen in table 1.

<b>Data</b>	<b>Description</b>
ID	Identification number
Overall nodal status (N0/N+)	N0: No positive lymph nodes, N+: $\geq 1$ positive lymph nodes
Mode of tumor detection	Screening vs symptomatic presentation
Age	In exact numbers, ex. 89.425 years
BMI	Calculated from weight and height data
Menopausal status	Pre-, post- or perimenopause
Multifocality	Foci of tumors within the same breast quadrant
Tumor size	The greatest diameter of the breast tumor
Histological grade	Valuation of differentiation of cancer cells based on their development and organization
Histological type	Signifies the growth pattern of the breast tumors
Lymphovascular invasion	Presence of tumor cells close to lymphatics or blood vessels
ER status	Estrogen receptors $\geq 1\%$ vs $< 1\%$
PR status	Progesteron receptors $\geq 1\%$ vs $< 1\%$
HER2 status	Presence of human epidermal growth factor receptor 2
Ki-67	Percentage of nuclear protein associated with cellular proliferation
Unilateral/Bilateral	Tumor in one vs both breasts
Quadrant of breast localization	Localization of tumor in breast
Tumor localization in breast side	Tumor in left vs right breast

Table 1: Original clinicopathological data. Information gathered from [10].

### 3.2 Mammography images

The mammograms were obtained from the Picture Archive and Communication Systems (PACS)/ Radiology Information Systems (RIS) in the summer of 2020. Images were taken of both breasts unless the patient previously had gone through a mastectomy. The images were collected from both mammography screening and from mammograms taken due to suspected cancer. Therefore, patients could have more than one mammogram registered per breast. Some patients also had magnified images of the cancerous area. Many different x-ray machines were used. Different nurses handled the machines, but the final diagnostics of axillary metastasis was done by a single pathologist after the images were collected. For this project, the image dataset was received from Magnus Dustler. The mammography dataset consisted of three views; Craniocaudal (CC), Mediolateral (ML) and Mediolateral Oblique (MLO), as well as a number of magnified images. Their original format was 'tif' format. An example of a CC image can be seen in figure 9.

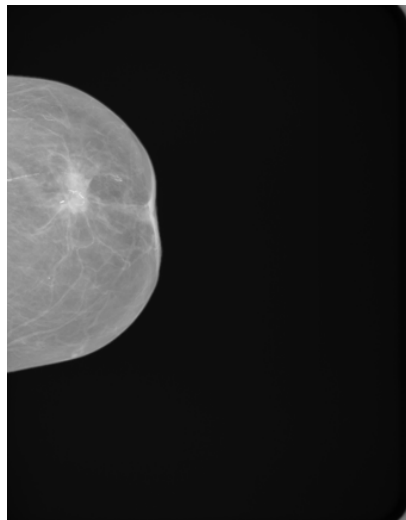


Figure 9: A mammography image taken from the craniocaudal (CC) view. The tumor can be seen as an almost star shape, in the middle of the breast. Image courtesy of Dustler, M.

## 4 Method

An important part of developing a model is to identify an architecture and hyperparameters values that are suitable for the input data. This section provides a description of how the architecture and hyperparameters were decided for each model. Due to the black box nature of deep neural networks [32], the methodology contains trial and error to estimate which hyperparameter values the networks can benefit from. Limitations in computer power brought challenges given the rather large size of the input data, and resulted in inconsistencies in how the models were developed. Overall, the architectures and hyperparameter values were obtained using either random, grid or manual search over a larger range of values. The hyperparameter values were then fine-tuned using a smaller range of values. To motivate choices of methodology, results are presented throughout the section. The resulting AUC-scores for each model are then presented in section 5. This section begins with introducing the pipeline of the methodology. Thereafter, the preprocessing, architecture and training of the CNNs and the MLPs are described.

### 4.1 Pipeline

First, all data was preprocessed. Then the clinicopathological data was combined with its corresponding mammography image. The first CNN was trained on images resized to  $600 \cdot 400$  pixels. The second CNN used Keras' InceptionV3 model pretrained on the ImageNet dataset, with the same input as in the first CNN. The third CNN was trained on images that, instead of being resized, had been divided into small patches. Each CNN's possibility to identify features that related to the nodal status was evaluated individually by its validation AUC-score. The two MLPs constitute a benchmark model, developed to approximate the result of Dihge et al. [2], and a final MLP. The final MLP was evaluated using an input of both clinicopathological data and feature vectors from one of the CNNs. The convolutional neural networks are presented first, then the multilayer perceptrons.

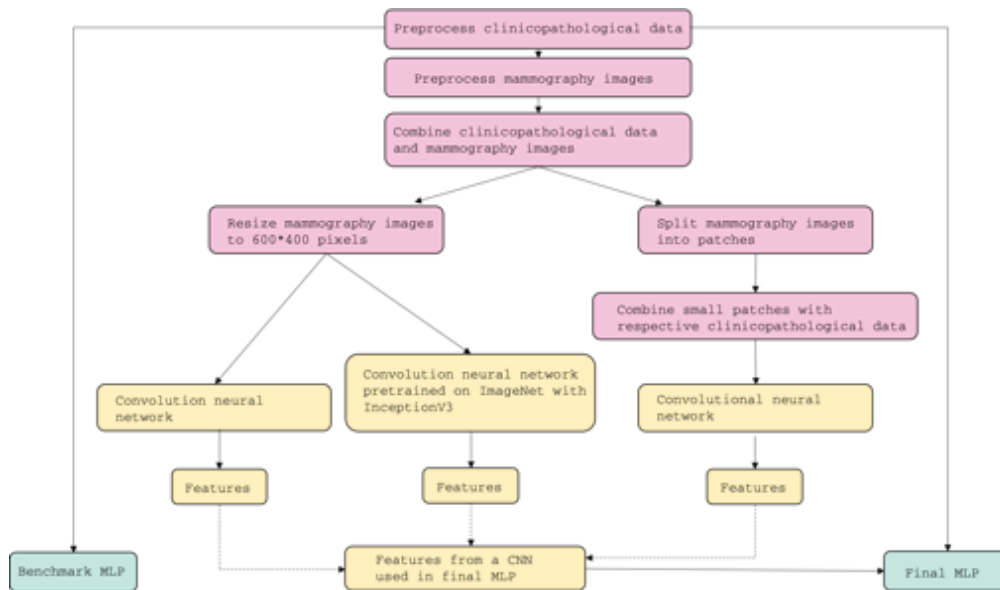


Figure 10: The pipeline of the project. After preprocessing of the data (seen in the pink boxes), three CNNs were constructed (yellow boxes). Two with mammograms resized to  $600 \cdot 400$  pixels and one with image patches as input. One MLP was developed to be used as a benchmark for the final MLP. The final MLP received features extracted from a CNN and clinicopathological data as input.



## 4.2 Convolutional neural networks

To increase the possibility of finding relevant features in the mammography images, the performance of three different CNNs were compared. An overview of the three networks and their input data can be seen in figure 11.

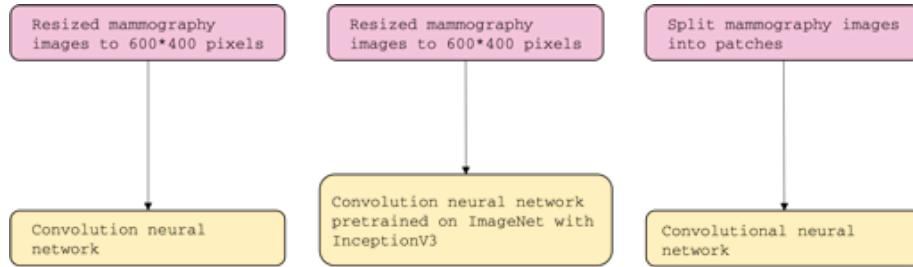


Figure 11: Three different convolutional neural networks were trained to extract features from the mammograms.

### 4.2.1 Convolutional neural network with images resized to 600 · 400

The craniocaudal mammography images were identified, reviewed and preprocessed using two softwares. The first part took part in MATLAB [33] and the second in Python [34] with Spyder [35] as IDE. The CC mammograms were extracted and manually inspected; a selection of images were removed. The selected mammograms were cropped and their background set to one colour. Then the images were resized. A flowchart of the image selection and preprocessing can be seen in figure 12. After image selection and preprocessing, the final image dataset consisted of 705 images, which were combined with their corresponding clinicopathological data. The dataset was divided into a training and a test set. The training set was further divided into a training set and a validation set used to develop models. The test set was only used to evaluate the performance of the final models. The first network attempts were trained on a 4 core Intel i7-4790K, 4.0 GHz, 1x NVIDIA Geforce GTX 980, 4 GB computer, and the later networks were trained on a 4 core Intel i7 – 47703.4 GHz, 1x NVIDIA GTX 1080, 11 GB computer.

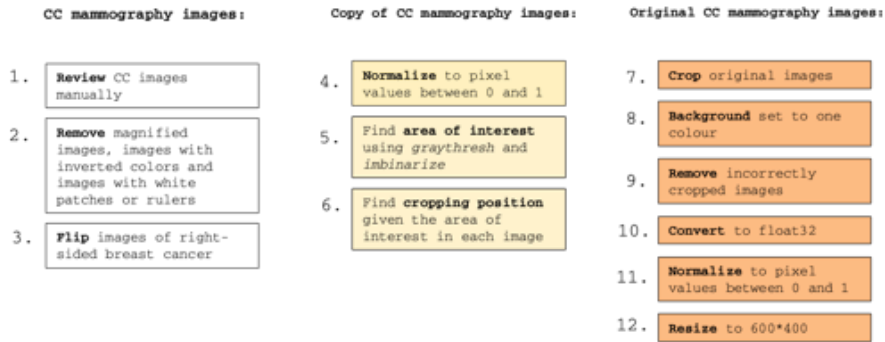


Figure 12: Pipeline of the image selection and preprocessing of the images. The white boxes illustrates preprocessing done on all CC mammograms. The yellow boxes illustrates preprocessing done on the copied images, and the orange boxes illustrates preprocessing done on the original images. Steps 1 – 9 were done in MATLAB, and steps 10 – 12 in Python.

#### *Mammography image selection and preprocessing*

The CC images were extracted using the images' file name, then they were manually inspected. A number of images were removed, for example magnified images and images with inverted colors. A few files were named by the full name of the view (i.e. *craniocaudal*) instead of the abbreviation (i.e. CC); These were included in the final dataset, despite them having inverted colours. They were first eliminated from the dataset due to their naming but after reconsidering, to be able to train the models on as much data as possible, they were also included. However, the first images with inverted colors were not included; Since they had already been manually excluded, a new manual search through the images would have been too time consuming. The width of the images ranged from approximately 1900 to 3300 pixels. The height ranged from approximately 2300 to 5900 pixels.

After the dataset was reviewed and modified, the images were cropped and their background set to one colour. Cropping was done to decrease the amount of background, whilst not loose any relevant information. To preprocess all the images with the same algorithm, the images of right-sided breast cancer were flipped. An algorithm consisting of steps three to nine in figure 12 was constructed. A copy of each image was made and used to find cropping positions in the original images. The area of interest was identified in the copied images, using MATLAB's *graythresh* function. This algorithm uses the Otsu's threshold method to find the optimal threshold between background and foreground [36, 37]. Thereafter, each image copy was binarized with MATLAB's *imbinarize* function. The binarized image copies were then used to identify cropping positions for their corresponding original image. The horizontal cropping positions were determined by moving through the image rows, from the middle, until a predetermined number of foreground pixels in a row were found. The cropping position was set a few rows out, to account for a margin of error. The predetermined number of foreground pixels was determined by visually examining the results of the cropping algorithm for a selection of images. The vertical cropping position was found by moving through the columns of the image, starting at column 100, until the limit between breast and background was found.

Using the cropping positions from the binarized images, the original images were

cropped, see figure 13. This resulted in images of different sizes. To remove non-relevant information, the backgrounds were set to one colour. The width of the cropped images ranged from approximately 400 to 2900 pixels. The heights ranged from approximately 1300 to 4800 pixels. Images with widths or heights smaller than 200 pixels were removed.

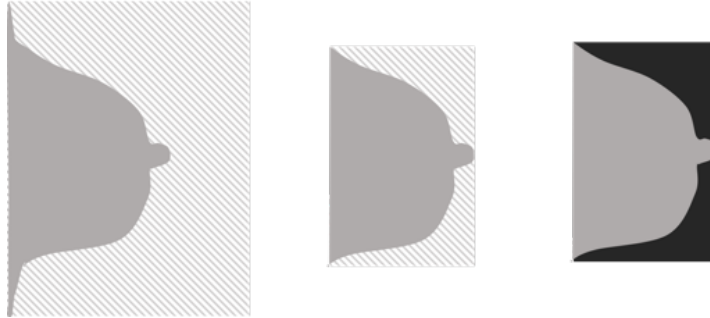


Figure 13: Schematic of a mammography image; illustrating cutting and setting its background to one colour. The figure corresponds to steps 7 – 8 in figure 12.

The preprocessed mammograms were imported into Spyder where they were converted to float32 matrices and normalized to values between 0 and 1. Given that the breast areas differed for different images, all cropped images were reshaped to a common size of  $600 \cdot 400$  pixels, using the Open Source Computer Vision [38] function *resize*, with *INTER\_AREA* as interpolation alternative. This size was chosen to keep as much information as possible, whilst reducing the image size significantly. The uniform size simplified the use of the images as input to the CNNs.

The corresponding clinicopathological data and preprocessed mammograms were combined in Spyder, using their common study ID. The most recent image of the cancerous breast was kept for each ID. Patients with bilateral cancer had two images and two unique IDs included in the study, one of each breast. Hence, both the unilateral and the bilateral cases could be handled in the same way. The final cohort included 705 images. The clinicopathological data and the image datasets were randomly divided into training/validation and test sets using the scikit learn [39] function *train\_test\_split*, see table 2. The training set consisted of 80 % of the datasets, resulting in a total of 564 images and corresponding clinicopathological data to train and validate the models.

Dataset	Node positive	Node negative	Total
Training	201	363	564
Test	43	98	141
All data	244	461	705

Table 2: Division of node positive and node negative images and corresponding clinicopathological data in training and test sets.

#### Architecture

The initial aim of the model development was to perform a random search to identify a suiting architecture. This, however, resulted in repeated out of memory errors.

After many attempts with different architectures and different ranges of hyperparameters, 13 architectures that had previously been tried and proven not to cause memory errors were chosen instead. These architectures, originally tested on a dataset that was normalized slightly differently than the dataset used, were still used due to time constraints. The network architectures can be seen in table 18 in the Appendix. The networks could all overfit when trained on the training data.

All CNNs in this project were trained during 75 epochs, using a stratified 5-fold cross validation where the data was shuffled using seed 42. For all models, CNNs and MLPs, the activation function for the input and hidden layers, and the output layers were ReLu and the sigmoid function, respectively. The loss function was binary cross-entropy for all networks and Adam was used as optimizer. The 13 networks in table 18 in the Appendix were first trained with learning rate 0.001 and no regularization. The two best models yielded very similar validation AUC-scores and their architecture and hyperparameters were identical, except for their batch size of 70 and 100, respectively. They had four convolutional and two max-pooling layers, see figure 14. The networks used batch normalization and had a flattening layer added after the convolutional and pooling layers, followed by a 20-noded dense layer and a one-noded dense layer. The AUC-scores were measured using scikit-learn’s [39] function, *auc*, that uses the trapezoidal rule to calculate the area under the ROC curve [40].

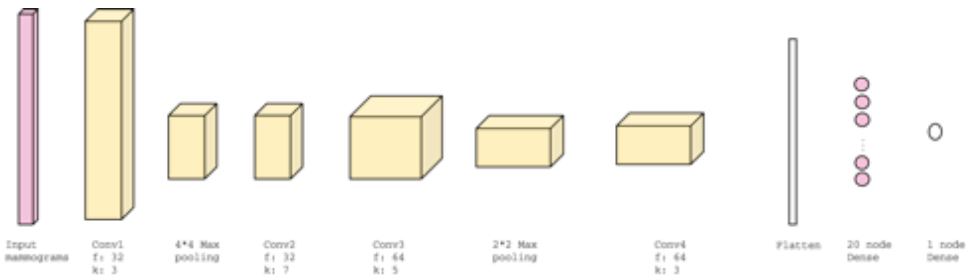


Figure 14: Architecture of the best CNN model after the first round of training. The number of filters were 32, 32, 64, 64, kernel sizes were 3, 7, 5, 3 and the aggressiveness of the max pooling was 4, 0, 2, 0. The 20-nodes dense layer represents the layer from which a feature vector may be extracted to use in the final MLP. Activation layers and batch normalization layers are not included in the figure.

Two grid searches were performed to regularize the network, using the hyperparameters in table 3. Dropout layers were added after the flattening layer and after the 20-node dense layer, respectively, see figure 15. During the first grid search, the L2-penalty was added to all convolutional and dense layers, and during the second one, the L2-penalty was only added to the dense layers. The networks were trained using the same settings, number of iterations and epochs as in the previous search, except for a change in batch size to 64 to limit the risk of out of memory errors.

Hyperparameter	Range
Regularization	L2(0.01), L2(0.001)
Dropout	0.1, 0.2

Table 3: Hyperparameter ranges used to fine-tune the CNN with grid search. To avoid memory error when tuning the hyperparameters, the batch size was slightly decreased compared to the original networks.

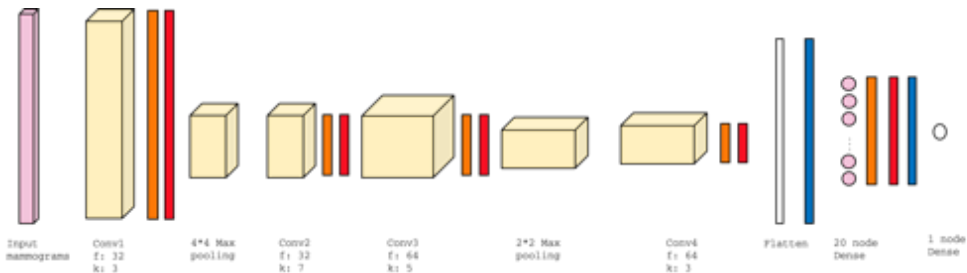


Figure 15: The same architecture as in figure 14, but with visible batch normalization layers and activation layers as well as being regularized with dropout layers. The orange layers are batch normalization layers, the red are activation layers and the blue are dropout layers.

The results with L2-penalty on both convolutional and dense layers indicated that the models were either under-regularized and overfitted, or stronger regularized but without finding general features in the mammography images. Either the training AUC-scores reached close to one and validation AUC-scores close to 0.5, or, with a higher L2-penalty, the training AUC-scores decreased but the validation AUC-scores remained low. The AUC-scores can be seen in the top part of figure 20, Appendix. With L2-penalty added on just the dense layers, the training AUC-scores for the lower regularization strengths indicated an overfitted model while the training AUC-scores for the higher penalties indicated that the model could be more regularized without losing its ability to find features. The AUC-scores can be seen in the bottom part of figure 20, Appendix. Therefore, the network with only the dense layers regularized was chosen for another fine-tuning grid search. This time, a higher L2-penalty strength was used and also a slightly lower learning rate to try to smoothen the accuracy and loss plots. The hyperparameter ranges can be seen in table 4.

Hyperparameter	Range
Learning rate	<b>0.0005</b> , 0.001
Regularization	L2(0.01), <b>L2(0.05)</b>
Dropout	<b>0.1</b> , 0.2, 0.3

Table 4: Hyperparameter ranges when fine-tuning the network that was regularized only on the dense layers, using grid search. The values in bold correspond to the hyperparameters that yielded the highest validation AUC-score, while the model still could learn according to the training AUC-score.

After the grid search, the model with highest validation AUC-score was still overfitted, and an even higher L2-regularization was tested (L2(0.1)). This resulted in a lower validation AUC-score, whilst the model still seemed overfitted. However, when calculating the validation AUC-score when extracting features (see section 5.2.2), the more regularized model gained a higher value and was thus kept. The accuracy plots for five folds can be seen in figure 21, Appendix. The network was trained during a two iteration stratified 5-fold cross validation. The mean and standard deviation of the ten AUC-scores for training and validation, respectively, were calculated using the numpy [41] functions *mean* and *std*. The same functions were used for these calculations for all models in the project. The standard deviation was used as dispersion metric due to the dependency of the AUC-scores.

The final hyperparameter values can be seen in table 5. To allow the network to learn as general features as possible, it was trained on all training data, leaving no part for validation. The model was then used for prediction on the test data, to get a final measurement of its performance.

<b>Hyperparameter</b>	<b>Value</b>
Learning rate	0.0005
Regularization	L2(0.1)
Dropout	0.1

Table 5: The final hyperparameters for the CNN with input images of size 600 · 400.

#### 4.2.2 Convolutional neural network pretrained with InceptionV3

No distinguishable features nor higher training AUC-score than approximately 0.5 was achieved with InceptionV3. The attempt to use the pretrained model for feature extraction was thus not pursued. The network was trained on a computer with specification: 4 core Intel i7-4770 3.4 GHz, 1x NVIDIA GTX 1080, 11 GB.

##### *Preprocessing mammography images*

The images used were preprocessed in the same way as the images in section 4.2.1.

##### *Architecture*

InceptionV3 requires three dimensional inputs, enabling prediction upon color images. Since the mammograms were in grayscale, each 2D-image was added to a depth channel three times, creating a 3D-image of depth three. InceptionV3 was imported with weights pretrained on the ImageNet dataset. As InceptionV3 was created for multi-class classification, the top layer was not included. To investigate the possibilities of using InceptionV3 as a pretrained model, histograms over 24 feature vectors of the model were plotted. The histograms showed that the model could not distinguish clear features in the data.

To adapt the model to the task, layers more suitable for the mammography dataset were added. First, a global average pooling (GAP) layer was added to reduce the output to a one dimensional vector. Then, a fully connected layer with 20 nodes and a fully connected layer with one node, to perform the final prediction, was added. It was examined whether InceptionV3 could classify correctly if a number of layers were unfrozen during training, see table 6. The learning rate was 0.001 and there was no regularization used on the added top layers of the pretrained model. Stratified 5-fold cross validation was used. The mean training and validation AUC-score, respectively, over five folds were calculated and the resulting scores were plotted over the number of frozen layers of InceptionV3, see figure 22, Appendix. Neither this showed potential for learning. Due to the lack of distinguishable features and the unsatisfying AUC-scores, the method was discarded.

Hyperparameter	Range
Number of unfrozen layers of InceptionV3	10, 20, 30, 40, 50

Table 6: Hyperparameter ranges used for the modified pretrained model InceptionV3.

### 4.2.3 Convolutional neural network with image patches

To not risk losing and/or changing information in the mammography images when resizing them, and to facilitate for a larger range of architecture alternatives, the mammograms were cropped into smaller patches. The preprocessing was therefore done slightly differently compared to the previous models. Due to the increased amount of images, the initial networks were trained on a selection of the training data. The training dataset contained 107 016 image patches and the test dataset contained 26 741 image patches.

#### *Preprocessing mammography images*

The images were cropped and preprocessed in the same manner as in section 4.2.1, except for the resizing step. After combining the images with their corresponding clinicopathological data, the images were once again imported to MATLAB where they were cut into patches of  $200 \cdot 200$  pixels from four directions, see figure 16. By this step, a decrease in image resolution due to resizing was avoided. Every cropped image resulted in a different number of small patches, depending on its size. All small patches originating from the same image were assigned the same label as the cropped image. To limit the amount of noise, patches that only included background pixels were removed. The networks were trained using a computer with specification: AMD Ryzen Threadripper 2990WX 32-Core Processor, 3000 Mhz, 4x NVIDIA GeForce RTX 2080 Ti 11 GB RAM, 128 GB RAM.

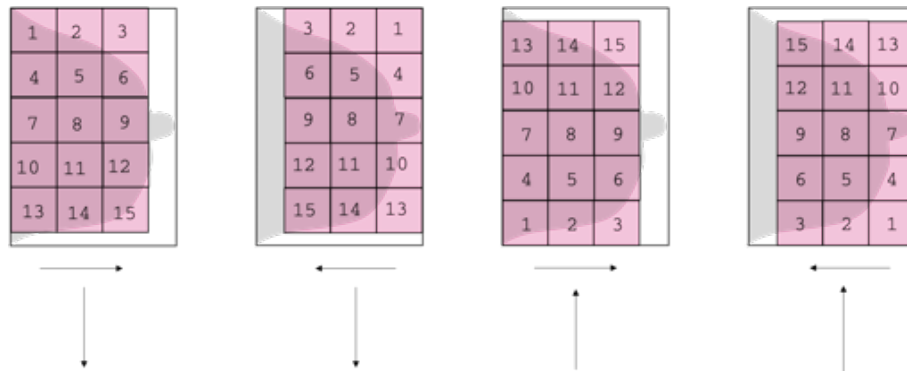


Figure 16: The original images were cut into patches by cutting images of size  $200 \cdot 200$  out of the original image from four directions. Depending on the image size, a different number of patches was extracted from each image.

#### *Architecture*

To reduce the time for training, a batch of images corresponding to 31 patients (which again correspond to slightly more than 5000 image patches) were chosen for preliminary training. The set had approximately the same distribution of node-positive and node-negative images as the original dataset. This set was divided into a training and a validation set through stratified 5-fold cross validation. Random search was performed to determine the model size and hyperparameters, see table 7. 20 combinations of hyperparameters were tested, and each was run through one iteration of the cross validation. The model with the best average validation AUC-score was kept (AUC-score 0.7583). The fine-tuning of this network was made with 5-fold stratified cross valida-



tion, using the hyperparameter ranges seen in table 8. The architecture can be seen in figure 17.

Hyperparameter	Range
Number of hidden layers	1, 2, 3, <b>4</b> , 5
Learning rate	0.01, <b>0.001</b> , 0.0001
Batch size	32, 64, <b>96</b>
Regularization	<b>None</b> , L2(0.1), L2(0.01), L2(0.001)
Dropout	0, 0.1, 0.2, 0.3, <b>0.4</b> , 0.5
Kernel sizes	3, 5, 7
Filters	16, 32, 64
Max pooling	Layer 1 : 0, 2, 4, Layer 2 – 5 : 0, 2
Batch normalization	Yes, <b>No</b>

Table 7: Hyperparameter ranges used for random search in CNN with image patches. Values marked in bold yielded the highest validation AUC-score. The number of filters were 64, 16, 16, 16, kernel sizes were 3, 5, 7, 7 and there was a  $2 \cdot 2$  max pooling after the two last convolutional layers, respectively.

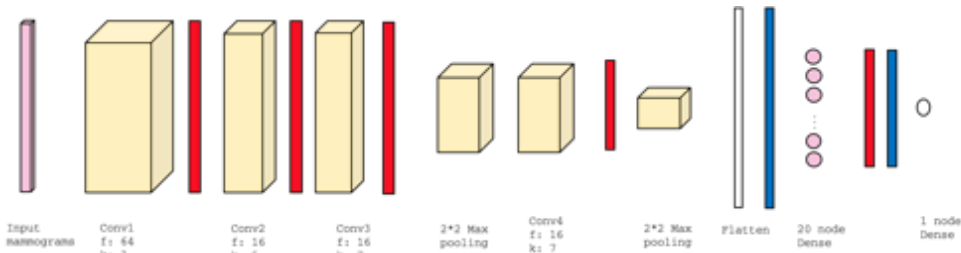


Figure 17: CNN architecture of the network. The red layers are activation layers and the blue layers are dropout layers. The architecture was decided upon after a random search where the networks were trained on image patches corresponding to 31 patients.

Thereafter, the same network was trained and validated on the whole training dataset. This yielded a network that did not seem to learn properly. To troubleshoot why an architecture and hyperparameter values that worked well for a small part of the dataset did not work for the full sized dataset, all regularization was removed. The non-regularized network's ability to train and overfit using the whole training dataset was tested and confirmed. A number of regularization strengths were tried, where the network did not seem to learn on the data. To try to reach the fine line between the network overfitting to the data and not learning at all, the network was fine-tuned using regularization with fairly low strength, see table 9. After identifying the hyperparameter values resulting in the highest validation AUC-score when the network was still able to learn, see table 10, all the training data was used to train the network and model was evaluated on the test data.

Hyperparameter	Range
Learning rate	0.005, 0.001, <b>0.0005</b>
Regularization	None, <b>L2(0.001)</b> , L2(0.005)
Dropout	0.35, 0.4, <b>0.45</b>

Table 8: Hyperparameter ranges used for fine-tuning of the CNN with image patches using grid search. The values in bold represents the hyperparameter values that resulted in the highest validation AUC-score.

Hyperparameter	Range
Learning rate	0.0005
Regularization	L2(0.0001), <b>L2(0.001)</b>
Dropout	0, <b>0.2</b>

Table 9: Hyperparameter ranges used when fine-tuning the network training on the full sized dataset using grid search. The values in bold represents the hyperparameter values that resulted in the highest validation AUC-score.

Hyperparameter	Value
Learning rate	0.0005
Regularization	0.001
Dropout	0.2

Table 10: The final hyperparameters for the CNN with image patches as input.

### 4.3 Multilayer perceptrons

Two multilayer perceptrons with different input data were constructed; a benchmark MLP with clinicopathological input data and a final MLP with both clinicopathological data and features from mammography images as input, see figure 18. The purpose of the first MLP was to achieve a benchmark validation AUC-score which could be compared both to the result of Dihge et al. [2] and the final MLP.

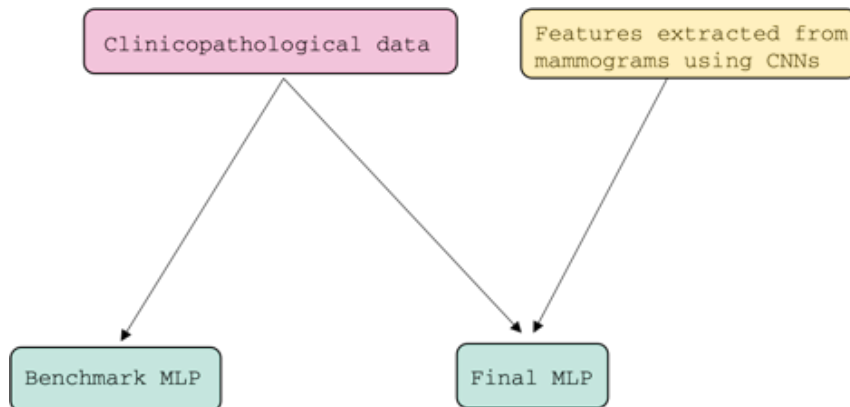


Figure 18: Two MLPs were constructed; One as a benchmark, with clinicopathological data as input, and one final MLP with both clinicopathological data as input and features from a CNN as input.

### 4.3.1 Benchmark multilayer perceptron

The benchmark MLP was developed to approximate the result of the model of Dihge et al. [2]. The cohort used for training was slightly smaller than that of Dihge et al., since some of the clinicopathological data did not have a corresponding image in the final mammography image dataset. First the preprocessing of the data is presented, followed by a discription of how the model was developed.

#### *Preprocessing clinopathological data*

The clinopathological data was preprocessed by normalization and random imputations. Some of the data was also re-encoded to make the input comprehensible for deep learning networks. The same 15<sup>2</sup> variables as used in [2] were chosen, see table 1. Menopausal status, histological type and the location of the main tumor were re-encoded using one-hot-encoding. The main part of the variables were binary or ordinal and needed no preprocessing. The remaining variables: age, BMI, Ki-67 percentage and tumor size were normalized. Missing values were accounted for by random imputations. The chance for a missing value to be replaced by 0 or 1 was equal in a binary category, i.e., given the small amount of data the distribution of values for each category was not accounted for. The re-encoding was done in Microsoft Excel [42] and the normalization and imputations were done in Python with Spyder as IDE.

#### *Architecture*

The MLP architecture consisted of one hidden layer and an output layer. The optimal MLP architecture for binary classification was determined using grid search over the following hyperparameters; number of hidden nodes, dropout on hidden layer, learning rate and strength of L2-regularization. The parameter ranges iterated over can be seen in table 11. Three iterations of each hyperparameter combination was run. For all architectures of the benchmark MLP, the networks were trained using 5-fold cross validation without shuffling. However, the two end-points in the dataset were fairly evenly distributed and thus the unshuffled division into folds should not differ considerably from a stratified and shuffled division. The mean training and validation AUC-score over the five folds was saved for each iteration. The batch size was 70 and the number of epochs was 300.

Hyperparameter	Range
Nodes	20, 25, 30, 35, <b>40</b> , 45
Learning rate	0.01, <b>0.001</b> , 0.0001
Regularization	L2(0.1), L2(0.01), <b>L2(0.001)</b> , <b>L2(0.0001)</b>
Dropout	<b>0</b> , 0.1, <b>0.2</b> , 0.3, 0.4, 0.5

Table 11: Hyperparameter ranges used for MLP grid search. The hyperparameter values that yielded the highest validation AUC-score for two of the iterations were very similar, therefore they were both kept to the next fine-tuning. The bold alternatives represent the combinations that yielded the highest AUC-score.

<sup>2</sup>ID was used for combining the clinicopathological data with the corresponding mammography images. Overall nodal status (N0/N+) was used as target. Quadrant of breast localization and Tumor localization in breast side were combined into a common variable describing the localization of the tumor. The remaining 15 variables were used as input to the network.

The models from each of the three iterations with the highest mean AUC-score from the 5-fold cross validation was examined. The two model's with the highest AUC-score had similar hyperparameter values and values in close range of these hyperparameter values were chosen for fine-tuning. The fine-tuning was done using grid search with the hyperparameters found in table 12. Each hyperparameter combination was run over three iterations, each with 5-fold cross validation and 300 epochs. The average AUC-score for each 5-fold cross validation was calculated, and the architecture yielding the highest average validation AUC-score was kept.

Hyperparameter	Range
Nodes	39, <b>40</b> , 41
Learning rate	<b>0.0005</b> , 0.001, 0.005
Regularization	L2(0.005), L2(0.001), L2(0.0005), <b>L2(0.0001)</b> , L2(0.00005)
Dropout	0, 0.05, 0.1, 0.15, 0.2, <b>0.25</b>

Table 12: Hyperparameter ranges used for MLP fine-tuning of the best two model from the preliminary grid search. The hyperparameter values that yielded the highest validation AUC-score are shown in bold.

The best architecture was run through a two iterations stratified 5-fold cross validation, shuffling with seed 42, yielding ten training and ten validation AUC-scores. The mean and standard deviation was calculated. Then, the network was trained on all training data, leaving no part for validation, and the architecture and its weights were saved through keras' *save()* function. The saved model was used to predict on the training and the test dataset, yielding an AUC-score for each dataset.

Hyperparameter	Value
Nodes	40
Learning rate	0.0005
Regularization	L2(0.0001)
Dropout	0.25

Table 13: The final hyperparameters for the benchmark MLP.

### 4.3.2 Final multilayer perceptron

The final multilayer perceptron had the same architecture and hyperparameter values as the benchmark MLP in table 13, section 4.3.1. Features were extracted from the CNN in table 5, section 4.2.1 and added as input to the final multilayer perceptron.

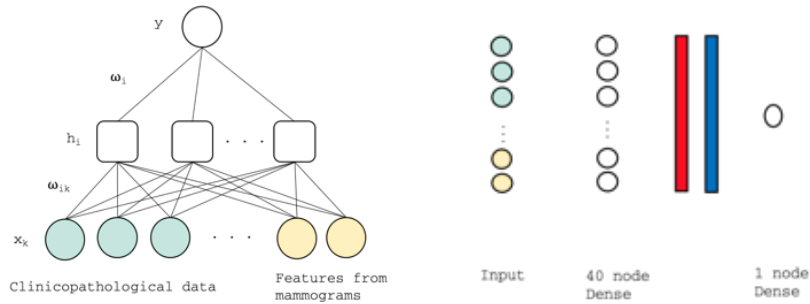
#### *Preprocessing clinicopathological data*

The clinicopathological data was preprocessed in the same way as the clinicopathological data in section 4.3.1.

#### *Input data and training*

The CNNs from section 4.2.1 and section 4.2.3, respectively, yielded similar highest validation AUC-score. Both models achieved high training AUC-scores, but the CNN in section 4.2.1 was slightly less overfitted to the data. Therefore, this CNN was chosen for feature extraction. The network was trained using a stratified 5-fold cross validation, where the architecture and its weights were saved through keras' *save()* function. The saved model was used to extract 20 features for each image of the validation dataset. To extract the feature vectors, the four topmost layers of the saved model were excluded, leaving the 20 node dense layer as the new top layer. This was done for each fold, resulting in one feature vector for each image in the training and validation dataset.

The feature dataset was combined with the clinicopathological data, and together they were fed into the fine-tuned architecture in section 4.3.1, see figure 19. The network was trained over two iterations of 300 epochs each, with stratified 5-fold cross validation using shuffled data with seed 42. This yielded ten AUC-scores for training and validation, respectively. The mean and standard deviation of the training and validation, respectively, were calculated. The network was then trained using all training and validation data, and the performance of the resulting model was tested by a prediction on the testdata.



(a) The dropout layer and activation layer is not included in the figure.

(b) The activation layer is the red layer, and the dropout layer is the blue layer.

Figure 19: The final MLP architecture, where both clinicopathological data and features from mammograms were used as inputs. The network in (a) does not include a dropout or a activation layer, while the figure in (b) does.

## 5 Results

In the previous section, results yielded from the process of developing the models were presented. Following are the resulting mean AUC-scores for the final models of section 4 presented. The scores were calculated for prediction on training data and validation data, respectively, during two iterations of 5-fold cross validation. The test AUC-score for each model, trained on the full training and validation dataset are also provided.

### 5.1 Convolutional neural networks

Since the CNN pretrained with InceptionV3 was discarded, just the results of the two other CNNs are presented.

#### 5.1.1 Convolutional neural network with images resized to 600 · 400

Two iterations with stratified 5-fold cross validation of the final model in section 4.2.1 yielded the results in table 14. The prediction performed using the test dataset yielded an AUC-score of 0.5266.

AUC-score	Training	Validation
Mean	0.8079	0.4735
Std	0.1290	0.046

Table 14: Two iterations of the model in table 5. The mean and standard deviation for training and validation over ten AUC-scores, respectively.

#### 5.1.2 Convolutional neural network with image patches

Two iterations with stratified 5-fold cross validation on all the training data with final model in section 4.2.3 can be seen in table 15. The test AUC-score was 0.5850. The accuracy plot for the prediction on the training data can be seen in figure 23, Appendix.

AUC-score	Training	Validation
Mean	0.9925	0.5069
Std	0.0152	0.0543

Table 15: Two iterations of the model in table 10. The mean and standard deviation for training and validation, respectively, over ten AUC-scores.

## 5.2 Multilayer perceptrons

First are the AUC-scores of the model which was develop to achieve a result similar to that of the model of Dihge et al presented, and then the final MLP's resulting AUC-scores.

### 5.2.1 Benchmark multilayer perceptron

Two iterations of stratified 5-fold cross validation yielded the mean AUC-score and standard deviation for training and validation, respectively, seen in table 16. The AUC-score for prediction on the test dataset was 0.6675.

AUC-score	Training	Validation
Mean	0.8368	0.7190
Std	0.0110	0.0465

Table 16: Two iterations of the model in table 13. The mean and standard deviation over ten AUC-scores for the training and validation dataset, respectively.

### 5.2.2 Final multilayer perceptron

The mean and standard deviation over ten AUC-scores, obtained by two iterations with stratified 5-fold cross validation with the model from 4.3.1, can be seen in table 17. The clinicopathological data and the features extracted from model 5 were used as input. The resulting model, after training the network using all training and validation data, yielded an AUC-score of 0.6172 when predicting on the test dataset.

AUC-score	Training	Validation
Mean	0.9237	0.6573
Std	0.0133	0.0470

Table 17: Two iterations of the model in table 13 with clinicopathological data and features as input. The mean and standard deviation for training and validation data over ten AUC-scores, respectively.



## 6 Discussion

Breast cancer is the deadliest cancer disease in the world among women. The overall prognosis is good, but worsens if the cancer metastasizes. For the majority of patients, breast cancer does not metastasize, but all patients still undergo either ALND or SLNB. The surgical intervention can cause considerable morbidity, which is why this project was conducted with the aim of contributing to the development of models for predicting sentinel nodal status preoperatively. No features that, sufficiently enough, could distinguish between nodal status N0 and N+ could be identified in the mammography images with the CNN models. Unsurprisingly, the benchmark MLP did not improve when features extracted from the mammograms were added as input to the network. The benchmark MLP, developed to approximately correspond to the model of Dihge et al., yielded a validation AUC-score of 0.7190 (std 0.0465), and the same network with features added as input achieved a validation AUC-score of 0.6573 (std 0.0470). The model developed by Dihge et al. obtained a validation AUC-score of 0.74 (95% CI 0.72 – 0.76). This indicates that the performance of the benchmark model is comparable to that of Dihge et al., even though a slightly smaller dataset was used for training the network. The final MLP obtained an inferior result.

During this project, three convolutional neural networks were developed. Two of the CNNs used resized mammograms as input, one of which used Keras' InceptionV3 model pretrained on the ImageNet dataset, and the third CNN used mammography image patches as input. This project has contributed with results derived from different preprocessing alternatives (i.e. various input sizes) and the use of transfer learning. We have revealed difficulties in finding features in mammography images using the previously mentioned models. Still, the results of this project may form a foundation for future attempts at developing models for preoperative prediction of sentinel nodal status with features in mammograms as input.

The plots in figure 21 show that the accuracy alternates between approximately 0.65 and 0.35. This indicates that the CNN either predicts all mammograms to be node positive or node negative, provided that node positive patients constitute around 35% of the cohort. Figure 23 shows how the network in section 10, during the final training on all training data, got stuck in a local minimum at around 0.65. This indicates that the model classified all patients as node-negative and hence did not learn from the data. However, training a network on image patches corresponding to 31 patients yielded a promising validation AUC-score (0.7583). It would be interesting to further investigate what properties these images contained and why the network's ability to learn decreased when using the full sized dataset.

The challenges in finding relevant features can have multiple explanations. Since it is impossible to, with certainty, predict lymph node metastasizing from a mammography image manually, neural networks might not be able to find relevant features in mammograms either. It is also possible that with, for example, more data and/or different or no resizing of images, relevant features from mammography images could be found and an improved nodal prediction could be achieved. There were quite large variation in the AUC-score between folds, even when the data was stratified, see figure 20, Appendix. This indicates that there were inconsistencies in the mammography images. The preprocessing and max pooling of the mammograms may have affected the information the images contained. By resizing the mammograms, information may have been change or lost, and the mammograms may also have had slightly different

resolution depending on their size before reshaping. The downsampling through max pooling can also have affected the information in the images, since the model in table 5, section 4.2.1, needed quite aggressive and early pooling. It is also possible that a larger model would have been required to distinguish desirable features. This would, however, have required more computational power. More computational power could also have enabled using all views of the mammography images, which potentially could have improved both the result and the generalizability of the models. Another approach for a better performing model could be to augment or gather more data.

The results might have been improved by higher consistency in preprocessing and model development. During preprocessing, it was (and still is) unclear what features corresponding to nodal status could be identified by the networks, and if any properties of the images are more important than others. Therefore, it was difficult to adapt the algorithms specifically to the dataset and the task. However, the manual search for exclusion of mammograms (e.g. magnified images) could have been repeated for a more reliable dataset, and all mammograms with inverted colours could have been included for consistency. The algorithm used for cropping could have been fine-tuned.

Identifying trends in how the regularization type and strength affected the CNNs was challenging. Different computers with different sized GPUs were used throughout the project and thus different strategies were implemented to find suitable hyperparameters. A more extensive random search might have resulted in alternative architectures and hyperparameter values, yielding better model performance, and also potentially a more reliable result provided that all models were developed and fine-tuned using the same procedure.

The results of this project have not clarified whether ANN predictions to determine axillary lymph node status can be improved by combining clinicopathological data with features from mammograms. They may however, by demonstrating behaviours of the models, contribute to the journey towards preoperatively prediction of sentinel nodal status.

## 7 Conclusion

The addition of data from mammography images to the benchmark MLP did not improve the prediction of sentinel nodal status. However, the results obtained in this project could be used to guide future attempts at using mammograms to improve models that predict nodal status, if it is possible to find features corresponding to nodal status in mammograms.

## 8 Future perspectives

A possible expansion of the project is to annotate the mammograms with the tumor's location and use a patch from that area as input to a neural network instead of the whole mammography image. However, this would require that the tumor's appearance indicates metastasizing. It is possible that, for example, the density of the breasts is more influential on metastasizing, in which case annotation of the tumor's location and the use of an image patch from that area might not be an effective strategy. It would also be interesting to examine multiple mammograms over time, to see if changes over time influence metastasizing. This could, possibly be done by using mammograms from patients in the national screening program.

## References

- [1] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J Clin*, 71(3):209–249, 2021. ISSN 1542-4863 (Electronic) 0007-9235 (Linking). doi: 10.3322/caac.21660. URL <https://www.ncbi.nlm.nih.gov/pubmed/33538338>.
- [2] L. Dihge, M. Ohlsson, P. Eden, P. O. Bendahl, and L. Ryden. Artificial neural network models to predict nodal status in clinically node-negative breast cancer. *BMC Cancer*, 19(1):610, 2019. ISSN 1471-2407 (Electronic) 1471-2407 (Linking). doi: 10.1186/s12885-019-5827-6. URL <https://www.ncbi.nlm.nih.gov/pubmed/31226956>.
- [3] Cancerfonden. Bröstcancer, 5 February 2020. URL <https://www.cancerfonden.se/om-cancer/cancersjukdomar/brostcancer>. Accessed: May 12 2021.
- [4] Bröstcancerförbundet. Kirurgi, 20 September 2019. URL <https://brostcancerforbundet.se/om-brostcancer/behandlingar/kirurgi/>. Accessed: 27 May 2021.
- [5] Ingvar Andersson. Bröstdiagnostik, book section 8, pages 109–140. AstraZeneca AB, Södertälje, 2 edition, 2009. ISBN 978-91-977572-5-6.
- [6] Regionala cancercentrum i samverkan. URL <https://kunskapsbanken.cancercentrum.se/diagnoser/brostcancer/vardprogram/diagnostik/>. Accessed: 27 May 2021.
- [7] Regionala cancercentrum i samverkan. Bröstcancer - nationellt vårdprogram. Report, 2020.
- [8] A. Chagpar, 3rd Martin, R. C., C. Chao, S. L. Wong, M. J. Edwards, T. Tuttle, and K. M. McMasters. Validation of subareolar and periareolar injection techniques for breast sentinel lymph node biopsy. *Arch Surg*, 139(6):614–8; discussion 618–20, 2004. ISSN 0004-0010 (Print) 0004-0010 (Linking). doi: 10.1001/archsurg.139.6.614. URL <https://www.ncbi.nlm.nih.gov/pubmed/15197087>.
- [9] Stacia Novia Marta, Nyoman Dwi Aussi Hary Mastika, and Hendry Irawan. A review and current update on sentinel lymph node biopsy of breast cancer. *Open Access Macedonian Journal of Medical Sciences*, 8(F):78–83, 2020. doi: 10.3889/oamjms.2020.4316. URL <https://oamjms.eu/index.php/mjms/article/view/4316>.
- [10] L. Dihge. Predictors of Lymph Node Metastasis in Primary Breast Cancer - Risk Models for Tailored Axillary Management. Thesis, 2018.
- [11] Leif Bergkvist. Bröstcancerkirurgi - lymfkörtlar, book section 11, pages 179–196. AstraZeneca AB, Södertälje, 2 edition, 2009. ISBN 978-91-977572-5-6.
- [12] Yuhao Dong, Qianjin Feng, Wei Yang, Zixiao Lu, Chunyan Deng, Lu Zhang, Zhouyang Lian, Jing Liu, Xiaoning Luo, Shufang Pei, Xiaokai Mo, Wenhui Huang, Changhong Liang, Bin Zhang, and Shuixing Zhang. Preoperative prediction of sentinel lymph node metastasis in breast cancer based on radiomics of

- t2-weighted fat-suppression and diffusion-weighted mri. European Radiology, 28(2):582–591, 2018. ISSN 1432-1084. doi: 10.1007/s00330-017-5005-7. URL <https://link.springer.com/article/10.1007/s00330-017-5005-7>.
- [13] Dechun Cai, Tong Lin, Kailin Jiang, and Zhizhong Sun. Diagnostic value of mri combined with ultrasound for lymph node metastasis in breast cancer: Protocol for a meta-analysis. Medicine, 98(30):e16528–e16528, 2019. ISSN 1536-5964 0025-7974. doi: 10.1097/MD.00000000000016528. URL <https://pubmed.ncbi.nlm.nih.gov/31348268https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6709118/>.
- [14] Xi'E Hu, Jingyi Xue, Shujia Peng, Ping Yang, Zhenyu Yang, Lin Yang, Yanming Dong, Lijuan Yuan, Ting Wang, and Guoqiang Bao. Preoperative nomogram for predicting sentinel lymph node metastasis risk in breast cancer: A potential application on omitting sentinel lymph node biopsy. Frontiers in Oncology, 11(1370), 2021. ISSN 2234-943X. doi: 10.3389/fonc.2021.665240. URL <https://www.frontiersin.org/article/10.3389/fonc.2021.665240>.
- [15] M. Ohlsson and P. Edén. Lecture notes on introduction to artificial neural networks and deep learning (fytn14/extq40/ntfo05f). Computational Biology and Biological Physics, Lund, 2020.
- [16] Christopher M Bishop. Pattern Recognition and Machine Learning, volume 4 of Information science and statistics. Springer, 2006. ISBN 9780387310732. doi: 10.1117/1.2819119. URL <http://www.library.wisc.edu/selectedtocs/bg0137.pdf>.
- [17] Christopher M. Bishop and Press Oxford University. Neural networks for pattern recognition. Oxford University Press, Oxford, 2013. ISBN 9780198538646 0198538642.
- [18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016. <http://www.deeplearningbook.org>.
- [19] Maryam M. Najafabadi, Flavio Villanustre, Taghi M. Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1):1, 2015. ISSN 2196-1115. doi: 10.1186/s40537-014-0007-7. URL <https://doi.org/10.1186/s40537-014-0007-7>.
- [20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2015.
- [21] Charles Ling, Jin Huang, and Harry Zhang. Auc: a statistically consistent and more discriminating measure than accuracy. Proc. 18th Int'l Joint Conf. Artificial Intelligence (IJCAI), 05 2003.
- [22] Michael J. Campbell, Stephen John Walters, and David Machin. Medical statistics : a textbook for the health sciences. 2021. ISBN 9781119423645 1119423643.
- [23] Azam Hamidinekoo, Erika Denton, Andrik Rampun, Kate Honnor, and Reyer Zwiggelaar. Deep learning in mammography and breast histology, an overview and future trends. Medical Image Analysis, 47, 2018. doi: 10.1016/j.media.2018.03.006.

- [24] Karl Weiss, Taghi Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3, 05 2016. doi: 10.1186/s40537-016-0043-6.
- [25] B. Q. Huynh, H. Li, and M. L. Giger. Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *J Med Imaging (Bellingham)*, 3(3):034501, 2016. ISSN 2329-4302 (Print) 2329-4302. doi: 10.1117/1.Jmi.3.3.034501.
- [26] François Chollet. Transfer learning fine-tuning, 12 May 2020 2020. URL [https://keras.io/guides/transfer\\_learning/](https://keras.io/guides/transfer_learning/). Accessed: 27 May 2021.
- [27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y. URL <https://doi.org/10.1007/s11263-015-0816-y>.
- [28] Keras. Keras applications, . URL "<https://keras.io/api/applications/#usage-examples-for-image-classification-models>". Accessed: 27 May 2021.
- [29] Keras. About keras, . URL <https://keras.io/about/>. Accessed: 27 May 2021.
- [30] C. Wang, Delei Chen, L. Hao, X. Liu, Yu Zeng, Jianwei Chen, and Guokai Zhang. Pulmonary image classification based on inception-v3 transfer learning model. *IEEE Access*, 7:146533–146541, 2019.
- [31] Qing Guan, Xiaochun Wan, Hongtao Lu, Bo Ping, Duanshu Li, Li Wang, Yongxue Zhu, Yunjun Wang, and Jun Xiang. Deep convolutional neural network inception-v3 model for differential diagnosing of lymph node in cytological images: a pilot study. *Annals of translational medicine*, 7(14):307–307, 2019. ISSN 2305-5839 2305-5847. doi: 10.21037/atm.2019.06.29. URL <https://pubmed.ncbi.nlm.nih.gov/31475177https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6694266/>.
- [32] Vanessa Buhrmester, David Muench, and Michael Arens. *Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey*. 2019.
- [33] Mathworks Inc. Matlab. R2019b. Natick, Massachusetts.
- [34] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995. Python version 3.6 and 3.8.
- [35] Spyder Doc Contributors. *Spyder*, 2009-2021. Spyder version 4 and 5.
- [36] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, January 1979. doi: 10.1109/TSMC.1979.4310076. URL <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4310076>.
- [37] MATLAB. graythresh, global image threshold using otsu’s method. URL <https://se.mathworks.com/help/images/ref/graythresh.html>. Accessed: 27 May 2021.

- [38] Open Source Computer Vision. Geometric transformations of images. URL "[https://docs.opencv.org/master/da/d6e/tutorial\\_py\\_geometric\\_transformations.html](https://docs.opencv.org/master/da/d6e/tutorial_py_geometric_transformations.html)". Accessed: 27 May 2021.
- [39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [40] scikit learn. `sklearn.metrics.auc`. URL "<https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html>". Accessed: 27 May 2021.
- [41] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL <https://doi.org/10.1038/s41586-020-2649-2>.
- [42] Microsoft Corporation. Microsoft excel. URL <https://office.microsoft.com/excel>.

## 9 Appendix

Architecture	Batch size	Filter 1	Kernel size 1	Max pooling 1	Filter 2	Kernel size 2	Max pooling 2	Filter 3	Kernel size 3	Max pooling 3	Filter 4	Kernel size 4	Max pooling 4	Dense 1	Dense 2
1	100	32	3	4	32	7	0	64	5	2	64	3	0	20	1
2	70	16	3	2	32	7	0	64	5	2	64	3	0	20	1
3	70	16	3	2	32	7	0	64	5	2	64	3	2	20	1
4	70	16	3	4	32	7	0	64	5	2	64	3	0	20	1
5	70	16	3	2	32	7	0	32	5	2	64	3	2	20	1
6	70	16	3	2	32	7	0	32	5	2	32	3	2	20	1
7	70	16	3	2	32	7	0	32	5	2	16	3	2	20	1
8	70	16	3	2	32	7	0	32	5	0	16	3	2	20	1
9	70	32	3	2	32	7	0	32	5	0	16	3	2	20	1
10	90	16	3	2	32	7	0	32	5	0	16	3	2	20	1
11	70	16	3	2	32	7	0	32	5	0	16	3	2	20	1
12	70	16	3	2	32	7	0	32	5	2	16	3	2	20	1
13	70	16	3	2	32	7	0	32	5	2	8	3	2	20	1

Table 18: Different architectures that did not cause an out of memory error when trained on the mammography dataset with images of size  $600 \cdot 400$ .



```

Validation
All AUC scores: [0.4193117273638489, 0.4910958904109589, 0.47226027397260273, 0.4805555555555557, 0.4388888888888889]
Mean: 0.460422467238371
Std: 0.026981926292068675
Training
All AUC scores: [0.8660991379310344, 0.7790533304776184, 0.9892696508888413, 0.9570553456703166, 0.9870226889500758]
Mean: 0.915700307835773
Std: 0.00166656810483922
Hyperparameters: 0.1_0.001_12(0.001)

Validation
All AUC scores: [0.47811560307383905, 0.4804794520547945, 0.5208904109589041, 0.5798611111111112, 0.44548611111111114]
Mean: 0.500966537861992
Std: 0.04613302085554098
Training
All AUC scores: [0.6148922413793103, 0.9190618976226173, 0.7863782394517027, 0.9006424622740176, 0.9341102644554012]
Mean: 0.8310179210366099
Std: 0.11994195296263646
Hyperparameters: 0.2_0.001_12(0.001)

Validation
All AUC scores: [0.47744737721349817, 0.46712328767123285, 0.5760273972602741, 0.5670138888888888, 0.4076388888888889]
Mean: 0.4990501679845565
Std: 0.06385387911109837
Training
All AUC scores: [0.7097413793103449, 0.5813664596273291, 0.5401799100449776, 0.8782096433373887, 0.7172312223858615]
Mean: 0.6853457229411803
Std: 0.11890595052243650
Hyperparameters: 0.1_0.001_12(0.01)

Validation
All AUC scores: [0.4340126962913465, 0.44931506849315067, 0.4239726027397261, 0.5253472222222223, 0.4736111111111111]
Mean: 0.4612517401715113
Std: 0.0361529213183557
Training
All AUC scores: [0.7357327586206898, 0.5078817733990147, 0.6494538445063182, 0.6090371603594372, 0.7069859768201319]
Mean: 0.6418183027411184
Std: 0.08017312449210003
Hyperparameters: 0.2_0.001_12(0.01)

malin_hj@rowling:~/Documents/Breast_cancer> ^C
malin_hj@rowling:~/Documents/Breast_cancer> python3 read_txt.py

Validation
All AUC scores: [0.4834614099565653, 0.4496575342465754, 0.49417808219178083, 0.5586055555555555, 0.4472222222222222]
Mean: 0.48663996083453986
Std: 0.04044312485622538
Training
All AUC scores: [0.85125, 0.8405225958449346, 0.8064253587491969, 0.9339395103626389, 0.9516125589635227]
Mean: 0.8767500047840586
Std: 0.05618411307292088
Hyperparameters: 0.1_0.001_12(0.001)

Validation
All AUC scores: [0.5168727029736051, 0.4253424657534247, 0.473972602739726, 0.5461805555555556, 0.5256944444444445]
Mean: 0.4976125542933512
Std: 0.04312820338311253
Training
All AUC scores: [0.8307974137931035, 0.8904904690511887, 0.9086528164489183, 0.9836929841412136, 0.834688693944633]
Mean: 0.8896644754758114
Std: 0.05601833147507477
Hyperparameters: 0.2_0.001_12(0.001)

Validation
All AUC scores: [0.5055128633478116, 0.5037671232876713, 0.4595890410958904, 0.4326388888888889, 0.43229166666666663]
Mean: 0.4667599166573858
Std: 0.03248091892015167
Training
All AUC scores: [0.8192456896551725, 0.6018419361747698, 0.5402227457699722, 0.9716761648630765, 0.9345158054257112]
Mean: 0.773580468377484
Std: 0.17388347722252745
Hyperparameters: 0.1_0.001_12(0.01)

Validation
All AUC scores: [0.5863681924490478, 0.44760273972602743, 0.5006849315068493, 0.5416666666666667, 0.44895833333333335]
Mean: 0.5050561727363849
Std: 0.05370692199548286
Training
All AUC scores: [0.6839008620689655, 0.8487470550439066, 0.7222745769972156, 0.8225438090969243, 0.9466820345350153]
Mean: 0.8048296675484053
Std: 0.09359886828966917
Hyperparameters: 0.2_0.001_12(0.01)

```

Figure 20: Top 4: L2-penalty on both convolutional and dense layers. Bottom 4: L2-penalty added only to the dense layers.

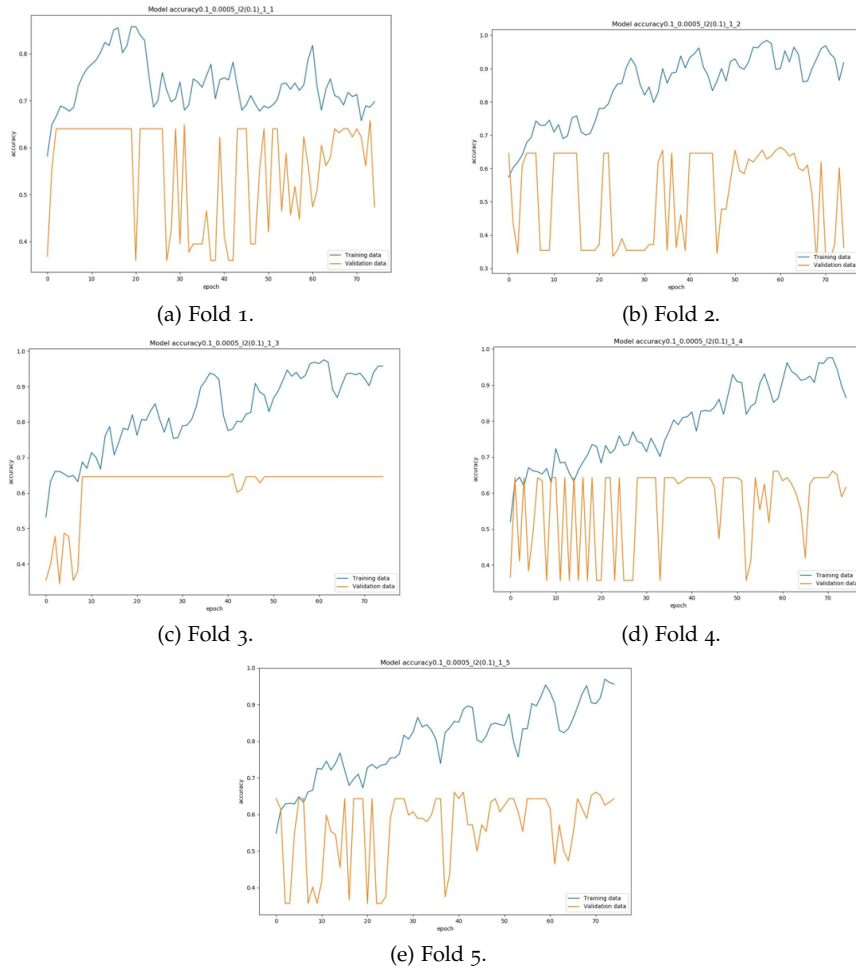
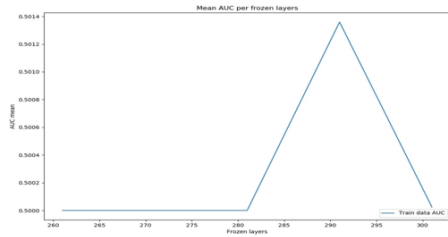
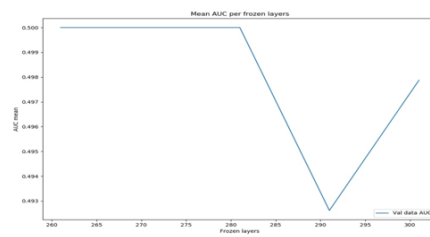


Figure 21: The accuracy plot over five folds for the model in table 5. The validation accuracy is alternating between approximately 0.35 and 0.65.

The average AUC-score obtained when unfreezing different amounts of layers in InceptionV3 can be seen in figure 22.



(a) Training



(b) Validation

Figure 22: The mean AUC-score over 5-fold cross validation for the training data and the validation data, respectively. An AUC-score of 0.5 is comparable with the model guessing.

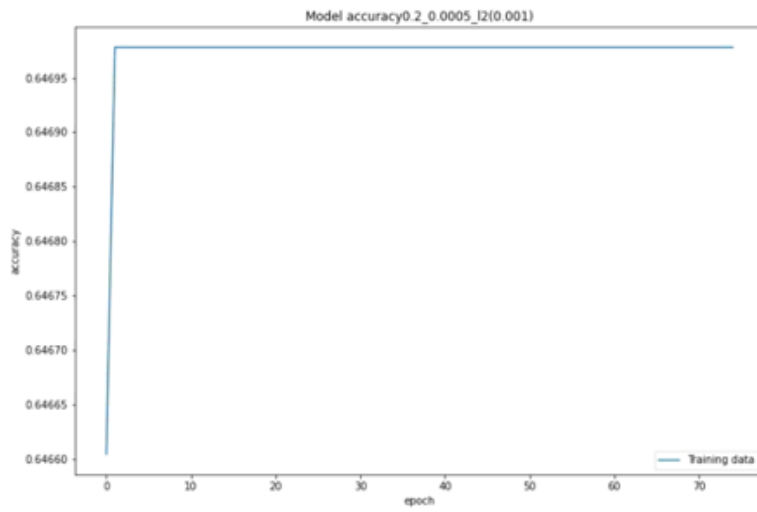


Figure 23: The accuracy during training, for the final CNN with image patches as input. The network seemed to end up in a local minimum.

Master's Theses in Mathematical Sciences 2021:E37  
ISSN 1404-6342  
LUTFMS-3418-2021  
Mathematical Statistics  
Centre for Mathematical Sciences  
Lund University  
Box 118, SE-221 00 Lund, Sweden  
<http://www.maths.lth.se/>